33 SUPER PROGRAMS INSIDE!

COMPUTE!'s GAZETTE

1988 Special
Display until May 12, 1988
$3.95
$5.50 Canada
02220

FOR **COMMODORE** PERSONAL COMPUTER USERS

Special 1988 Issue

33 of the all-time best from COMPUTE! and Gazette

Laser Chess

A dazzling, two-player strategy game that moves chess into the 21st century.

PLUS:

Snapshot

Store up to 56 different screens and instantly retrieve any one for display.

64+

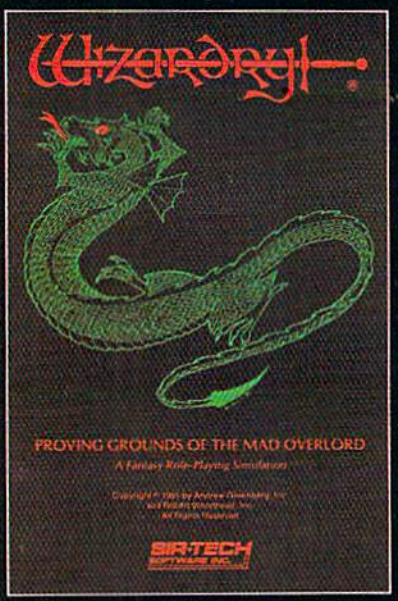
Add lots of useful features to your 64's operating system.

Miami Ice

Frozen roads in South Florida? A frantic driving experience is guaranteed in this challenging game with versions for the 128 and 64.

85
0 71486 02657 0

Wizardry®



Now the world of Wizardry comes to Commodore 64/128

PROVING GROUNDS of the MAD OVERLORD

Also available for:
Apple II Series,
IBM PC & compatibles,
Tandy 1000 SX & EX,
Macintosh

With your Commodore 64 or 128 you can now experience an exciting initiation into the internationally-acclaimed world of the Wizardry Gaming System.

Begin with the first scenario*: Proving Grounds of the Mad Overlord. It challenges you to overcome devious tricks and sudden traps, strange and mysterious encounters, maze upon maze of puzzles, stimulating and intricate sub-plots and sophisticated themes. Ingenious multi-level, mind-bending tests abound with vile creatures, unspeakable villains, courageous adventurers and strong allies to hinder and help you achieve ultimate victory.

SIR-TECH

Sir-tech Software, Inc.,
P.O. Box 245, Ogdensburg, NY 13669
(315) 393-6633

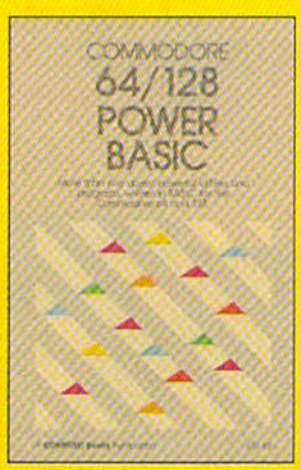
PLAY IT TO THE HILT!

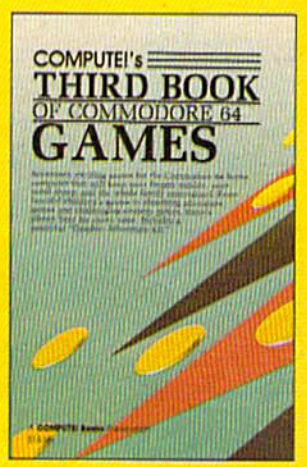
* Many other Wizardry scenarios coming for Commodore players.

Exciting & New!

Commodore 64/128 Books

from COMPUTE!





Get more power, education, and relaxation from your Commodore 64 or 128 with these new books from COMPUTE!. And, as always, the programs are thoroughly tested and easy to enter using our special program entry system. (Disks are available, too.)

COMPUTE!'s Commodore 64/128 Power BASIC
Edited
Collected in this one volume are more than two dozen short BASIC programs from *COMPUTE!'s Gazette*'s popular monthly column, "Power BASIC." Included are utilities that add an 8K RAM disk; read and write to disk sectors; make programs read joysticks more quickly; and automatically generate line numbers. There are also programs such as "Stop and Go," which creates a pause button to temporarily halt a program, and "Time Clock," which puts a digital clock on your screen. *There is a companion disk available for $12.95 which includes all the programs from the book.* (998BDSK)
$16.95 ISBN 0-87455-099-8

COMPUTE!'s More Machine Language Games for the Commodore 64
Edited
Seven of the best machine language games for the 64 have been gathered into one volume in this follow-up to the popular *COMPUTE!'s Machine Language Games for the Commodore 64*. Selected from recent issues of *COMPUTE!* and *COMPUTE!'s Gazette*, the games range from the frantic "Prisonball" to the delightful "Biker Dave." This is more than just a collection of exciting fast-action games, though, because complete and commented source code for each program is included in the book. Machine language programmers can see exactly how each game is written and what design techniques are used. *A disk is available for $12.95 which includes all the programs in the book, including source code.* (947BDSK).
$16.95 ISBN 0-87455-094-7

COMPUTE!'s Third Book of Commodore 64 Games
Edited
COMPUTE! Publications, the leading publisher of programs for the Commodore 64, has brought together another exciting collection of nerve-tingling games that will delight the whole family—from preschoolers to teenage arcade fans to those who enjoy games of logic. Clearly written, with non-technical instructions, this book contains hours of challenging entertainment for beginning computer fans as well as experienced programmers. *There is a companion disk available for $12.95 that includes all the programs in the book.* (955BDSK).
$15.95 ISBN 0-87455-095-5

These books are now available at your local book or computer store.
You can also order directly from COMPUTE! by calling toll free 800-346-6767 (in NY call 212-887-8525) or mailing your order to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.
Customer Service hours are 10:00 a.m.–12:30 p.m. and 1:30 p.m.–3:00 p.m. EST Mon.–Fri.

Please include $2.00 postage and handling per book or disk. NC residents add 5 percent sales tax and NY residents add 8.25 percent sales tax. Please allow 4–6 weeks for delivery.

COMPUTE! Publications, Inc.
A Capital Cities/ABC, Inc. Company

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

FEEL THE HEAT

No matter what kind of game you're in the mood for, you'll find that if it's in a box marked FIREBIRD, it's really hot!

Firebird offers the excitement of a whole range of challenges — Adventures and Strategies for long rainy weekends, Arcade games for exhilarating evenings, Simulations when you need to get away from it all, Productivity software when you're feeling — well, productive.

Look for the Firebird — it stands for top-quality software in virtually *every* category.

How About This Ball of Fire From Firebird!

This is SENTRY, a unique Strategy game with over 10,000 full-color 3D vector landscapes and four-way scrolling. Do you have truly exceptional concentration and near perfect hand/eye coordination? Then maybe — just maybe — you possess the skill to outwit the Sentry at each level of the extraordinary challenge.

Firebird Fireworks Unlimited

When your mood shifts from the deliberation of strategic planning to the excitement of fast-paced Arcade action, look for Firebird's ELITE, a unique game which encompasses the best of both worlds. Or try STARGLIDER, an amazing 3D combat flight simulation.

If it's Adventure you're after, here's the breathtakingly illustrated GUILD OF THIEVES, with its revolutionary text-handling system.

Firebird even offers economical, professional-quality Productivity software! ADVANCED ART STUDIO is a comprehensive art package with powerful graphics routines.

This is Firebird — the best in interactive software of every description. We'll prove to you that you don't have to keep switching brands to satisfy your obsession for challenge!

CREDIT CARD HOTLINE
(201) 444-5700

MasterCard VISA

FIREBIRD

"The First Full Line In Software"

Firebird Licensees, Inc.
P.O. Box 49, Ramsey, NJ 07446
(201) 444-5700

Firebird software for Commodore 64 systems, including the popular Golden Path, Pawn, Colossus Chess IV, Cholo, Talking Teacher, Concise Music System and Advanced Music System, is available from $24.95 to $59.95.

GAZETTE contents

games

education

graphics

disk utilities

applications

128 only

utilities

typing aids

COMPUTE!'s GAZETTE is published monthly by **COMPUTE!** Publications, Inc., 825 7th Avenue, New York, NY 10019 USA. Phone: (212) 265-8360. Editorial offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. POSTMASTER: Send address changes to *COMPUTE!'s GAZETTE*, P.O. Box 10957, Des Moines, IA 50340. Second class postage paid at Greensboro, NC 27403 and additional mailing offices.

ISSN 0737-3716.

COMPUTE! Publications, Inc. is part of ABC Consumer Magazines, Inc., one of the ABC Publishing Companies: ABC Publishing, President, Robert G. Burton; 1330 Avenue of the Americas; New York, New York 10019.

Publisher/Editorial Director	William Tynan
Managing Editor	Kathleen Martinek
Associate Publisher	Selby Bateman
Editor, COMPUTE! and COMPUTE!'s Gazette	Lance Elko
Assistant Editors, COMPUTE! and COMPUTE!'s Gazette	Rhett Anderson, Randy Thompson, Clifton Karnes
Production Director	Tony Roberts
Editor, COMPUTE!'s Atari ST Disk & Magazine and COMPUTE!'s PC Magazine	Tom R. Halfhill
Editor, COMPUTE!s Apple Applications Magazine	Gregg Keizer
Features Editor	Keith Ferrell
Assistant Technical Editors	Dale McBane, Jim Fuchs
Assistant Editor, COMPUTE!'s Atari ST Disk & Magazine	Todd Heimarck
Assistant Editor	John Shadle
Assistant Features Editor	Tom Netsel
Programming Supervisor	Patrick Parrish
Editorial Programmers	Tim Midkiff, William Chin, George Miller
Buyer's Guide Coordinator	Caroline Hanlon
Copy Editors	Karen Uhlendorf, Karen Siepak, Jill Champion, Lori Sonoski
Editorial Assistant	Mickey McLean
Submissions Reviewer	David Hensley
Programming Assistants	Troy Tucker, Joyce Sides
Executive Assistant	Debi Nash
Administrative Assistants	Julia Fleming, Iris Brooks, Sybil Agee, Claudia Earhart
Receptionist	Anita Armfield
Associate Editors	Jim Butterfield Toronto, Canada
	Fred D'Ignazio E. Lansing, MI
	David Thornburg Los Altos, CA
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division Editor	Stephen Levy
Assistant Editors	Tammie Taylor, Robert Bixby, Lynne Weatherman
Programming Assistant	David Florance
Production Manager	Irma Swain
Art Director	Janice R. Fary
Assistant Art Director	Lee Noel, Jr.
Assistant Production Manager	De Potter
Designer	Tony Jacobson
Artists	Robin Case, Kim Potts, Scotty Billings
Typesetting	Terry Cash, Carole Dunton
Illustrator	Harry Blair
Group Advertising Director/ Consumer Electronics	Peter Johnsmeyer
Advertising Director	Bernard J. Theobald, Jr.
Production Coordinator	Kathleen Ingram
Customer Service Manager	Diane Longo
Dealer Sales Supervisor	Jose Cruz
Individual Order Supervisor	Cassandra Green

William Tynan, Vice President
Richard J. Marino, Vice President, Advertising Sales
Ilene Berson Weiner, Vice President, Production

Editorial offices:	324 West Wendover Avenue, Suite 200 Greensboro, NC 27408 USA
Corporate offices:	825 7th Avenue New York, NY 10019, 212-265-8360
Individual Orders:	800-346-6767 (In NY 212-887-8525) 10:00 AM-12:30 PM; 1:30 PM-3:00 PM Monday-Friday
Dealer Sales:	800-638-3822 (In NY 212-887-8566) 9 AM-5 PM Monday-Friday

Advertising Sales Representatives

New England & Mid-Atlantic
Bernard Theobald, Thomas Link: 212-315-1665

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

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

Southeast & International
Harry Blair: 919-275-9809

Send all advertising materials to:
Kathleen Ingram
324 West Wendover Ave., Suite 200
Greensboro, NC 27408

Member MPA ABC Audit Bureau of Circulations

editor's notes

When we first decided to launch a compilation of our best-ever Commodore programs, we looked through the archives of COMPUTE! and GAZETTE. We had our favorites, and there were those that drew impressive responses from our readers. We compiled a list of these and discovered that there was enough material (over 100 programs) to fill at least two magazines.

Our initial offering, *Best of COMPUTE! & GAZETTE For Commodore 64 & 128*, contains many of the programs from our "best of" list. It hit the newsstand last October and was received very well. This issue contains 33 more of our best-ever ready-to-type-in 64 and 128 programs, spanning several categories: games, education, graphics, disk and programming utilities, applications, and 128-only programs. For those of you unfamiliar with our program listing conventions, see "How To Type In COMPUTE!'s Gazette Programs," "Automatic Proofreader," and "MLX" before beginning your typing session. For those who would rather spend a little money and save a lot of typing time, we have a special disk offer (see page 26). For $6.95 plus $2 postage and handling, we're offering a companion disk that contains all the programs in this issue. The disk contains an easy-to-use menu that calls the programs with the press of a key.

We carefully selected the programs for this issue, so that regular GAZETTE subscribers will have many programs that never appeared in GAZETTE. And newer subscribers (the past 12 months) will see some GAZETTE programs that are no longer available by back order. We're confident that this collection offers some of the finest Commodore programs ever published, and represents an excellent value for the Commodore 64 and 128 owner.

COMPUTE! Publications, Inc. abc
One of the ABC Consumer Magazines, Inc.
A Capital Cities/ABC, Inc. Company
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

Preview The Future!

PROJECT: STEALTH FIGHTER

Stealth aircraft . . . the hottest topic of discussion in aviation circles today. Space age technology and engineering make these aircraft virtually undetectable by enemy radar. Stealth fighters are believed to now be operating from top secret military bases, flown only on the most sensitive missions and only by the most experienced pilots.

Precise information about this next generation of Air Force jets is highly classified. But extensive research has enabled the aviation experts at MicroProse to incorporate the potential design and performance characteristics in a powerful new simulation.

Slip into the cockpit of **PROJECT: STEALTH FIGHTER** and familiarize yourself with the advanced "heads-up display"(HUD) and sophisticated 3-D out-the-cockpit view. At your disposal is a dazzling array of state-of-the-art electronics, weapons and countermeasures.

But remember, flying a stealth aircraft takes more than just fighter pilot reflexes. The real challenge is mastering the jet's electromagnetic profile to avoid enemy radar, while executing a deep-strike mission into North Africa or conducting delicate photo reconnaissance over Eastern Block seaports. Also featured are carrier and land-based takeoffs and landings, as well as land, sea, and air targets and threats.

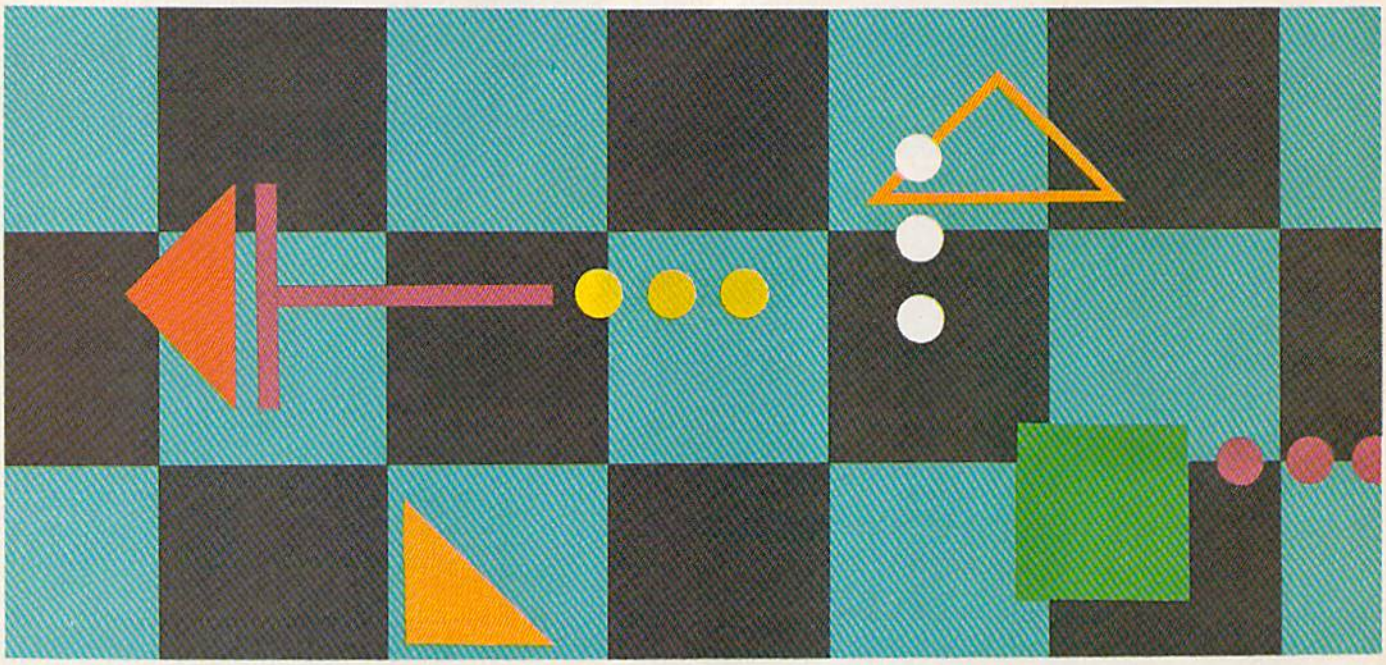
PROJECT: STEALTH FIGHTER combines the combat action thrills of the best-selling F-15 STRIKE EAGLE with the sophisticated flight environment pioneered in GUNSHIP. Add the danger and suspense of clandestine missions in an innovative "invisible aircraft" and you have **PROJECT: STEALTH FIGHTER,** the latest flying challenge from MicroProse!

PROJECT: STEALTH FIGHTER is available from a "Valued MicroProse Retailer" (VMR) near you. Call us for locations!

For Commodore 64/128, IBM-PC/compatibles and Apple II/e/c. Call or write for specific machine availability, and for MC/VISA orders if product not found locally.

MICRO PROSE™
SIMULATION • SOFTWARE

180 Lakefront Drive • Hunt Valley, MD 21030 • (301) 771-1151



Laser Chess™

Mike Duppong

Laser Chess™ *won First Prize in our $10,000 programming contest for COMPUTE!'s Atari ST Disk & Magazine. Awarded $5,000 for its originality and skillful programming,* Laser Chess *is a two-player strategy game patterned after traditional chess—with some fascinating new twists. The original version was written in the Modula-2 language for the Atari ST. Here we provide a machine language translation for the Commodore 64 (and Commodore 128 in 64 mode). At least one joystick is required.*

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

Since the program is written in machine language, it must be typed in with the "MLX" machine language entry program printed elsewhere in this issue. Here are the starting and ending addresses for MLX:

Starting address: 0801
Ending address: 1BB8

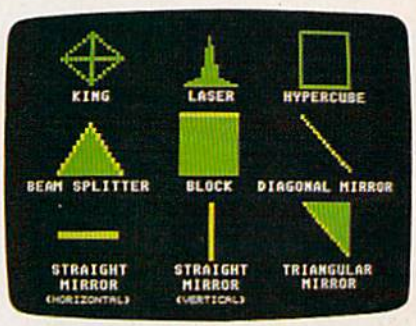
There are eight basic types of pieces in *Laser Chess*, and each has unique capabilities. Over time, you'll learn each piece's advantages and limitations. Obviously, the more you play *Laser Chess*, the more you'll understand the pieces in your arsenal, which in turn will make you a better player. So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

A Geometric Army

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

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

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



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

The *king* is the most important piece in *Laser Chess*. When the king is eliminated, the other player wins the game. Since it has no reflective surfaces, it can be destroyed by a laser from any angle. It can also be

captured by an opposing piece. The king is not totally defenseless, however. It can capture any opposing piece by moving onto its square. But you can use it for a capture only once per turn.

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

Tricky Pieces

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

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

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

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

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

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

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

Making Moves

As in the conventional game of chess, a move in *Laser Chess* consists of moving or otherwise manipulating a game piece. A cursor indicates your position on the board. Use the joystick to move the cursor over the piece you wish to move. (If you have only one joystick, plug it into port 2. You can simulate the second joystick by pressing the left-arrow, 1, 2, and CTRL keys to move the joystick left, right, up, and down, respectively, and pressing the space bar as a substitute for the button.)

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

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

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

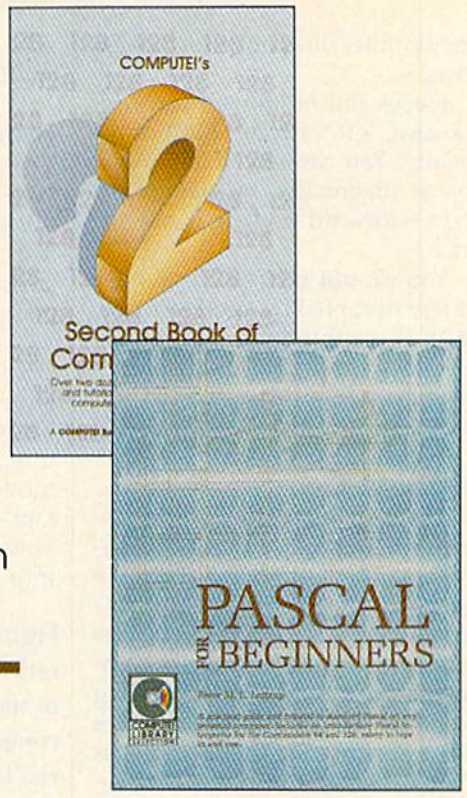
Figure 3: This full-screen view of Laser Chess shows its 9 × 9 board grid and game controls.

Before you move or rotate a piece, you must select it. When a piece is selected, the appearance of the piece changes.

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

After you've selected a piece, your next decision is whether to move or rotate it. Moving a distance of one square takes one move; moving two squares takes two moves (although you can move a piece two squares in one step). Since you have only two moves per turn, the maximum distance a piece can be moved in one turn is two squares.

New books from COMPUTE!



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

Pascal for Beginners

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

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

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

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

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

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

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

$18.95 ISBN 0-87455-080-7

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

COMPUTE!'s Second Book of the Commodore 128

$16.95 ISBN 0-87455-077-7

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

Mapping the Commodore 64, Revised

$16.95 ISBN 0-87455-082-3

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

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

Please include shipping and handling: $2.00 per book in U.S. and surface mail; $5.00 airmail. NC residents add 5 percent sales tax and NY residents add 8.25 percent sales tax. Please allow 4–6 weeks for delivery. Customer Service hours are 10:00 a.m.–12:30 p.m. and 1:30 p.m.–3:00 p.m. EST Mon.–Fri.

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company

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

The computer does not allow illegal moves.

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

You cannot move a piece onto a square occupied by another piece. The only exceptions are captures with blocks and kings, and moves of the hypercube as described above.

Rotating A Piece

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

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

Special Features

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

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

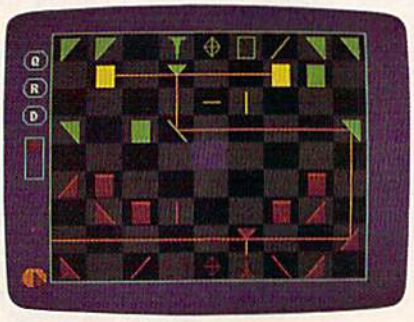
The restart button (R) lets you start a new game without finishing the current game. (For instance, a player may be so hopelessly behind that he or she wants to resign.) Again, the program asks that you confirm this choice.

Firing The Laser

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

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

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



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

Laser Chess Strategy

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

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

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

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

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

See program listing on page 83.

WORK WITH US ON A PROFESSIONAL LEVEL.

Some people really like working on their machines. And then there are some who prefer to do their tinkering under the hood. For those of you who can't wait to get your hands greasy, we proudly introduce geoProgrammer. The most sophisticated machine language programming tool on the market.

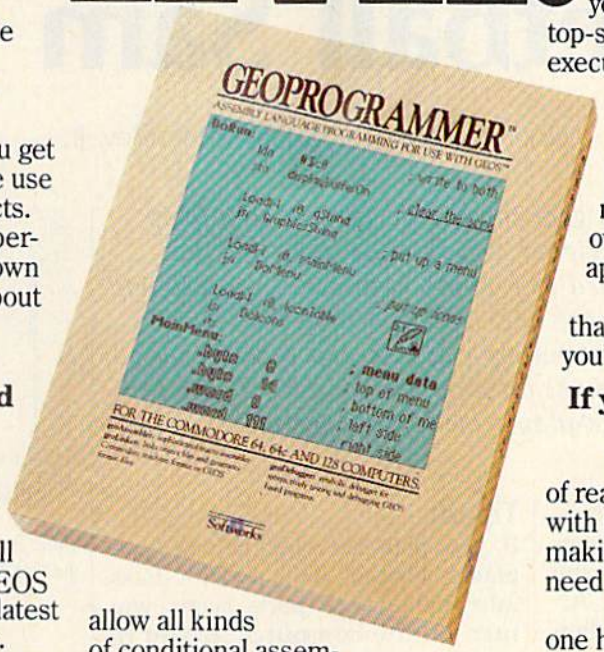
With geoProgrammer, you get the same kind of technology we use for developing our other products. Which means you can write supercharged software. Pump your own programs. And assemble just about any kind of application you can imagine.

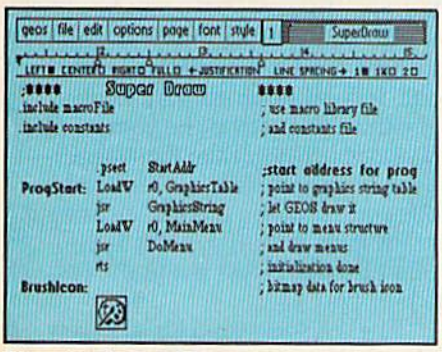
It reads and writes and stomps on bugs.

The first three tools that any serious programmer needs are an assembler, a linker and a debugger. So we've installed all three, complete with sample GEOS applications that teach you the latest programming tricks in seconds.

geoAssembler not only reads directly from geoWrite files, but contains enough Pseudo Ops to allow all kinds of conditional assemblies. It has all the state-of-the-art features you'd expect, including some you probably never thought possible. For example, integrating graphics is as simple as cutting and pasting the image from geoPaint directly into your program.

geoLinker ties your program modules together, supporting GEOS SEQ and VLIR applications and desk accessories. You can even use geoAssembler and geoLinker to create non-GEOS applications.





*Also available for 80 col. C128's.

geoDebugger allows your program to be tested in memory with full symbolic disassembly, along with line assembly for patching code in memory. It also allows your code to be single-stepped or top-stepped, with sub-routines fully executed. It can stop a running program with one key, or use up to eight conditional breakpoints. When your program hits the breakpoint, it prominently displays the error in an overlay window, leaving the applications screen intact.

Pro enough for you yet? Well, that's only a *partial* list of what you're in for.

If you understood all that, read on.

Chances are that we lost a lot of readers by now. But if you're still with us, hang in there. You have the makings of a real GEOS pro. All you need now are the right tools.

And all of them come in this one handy box.

So if you're serious about programming, consider geoProgrammer. After all, you've got nothing to lose —except your amateur status.

To order call 1-800-443-0100 ext. 234

geoProgrammer $69.95

(California residents add 7% sales tax.)

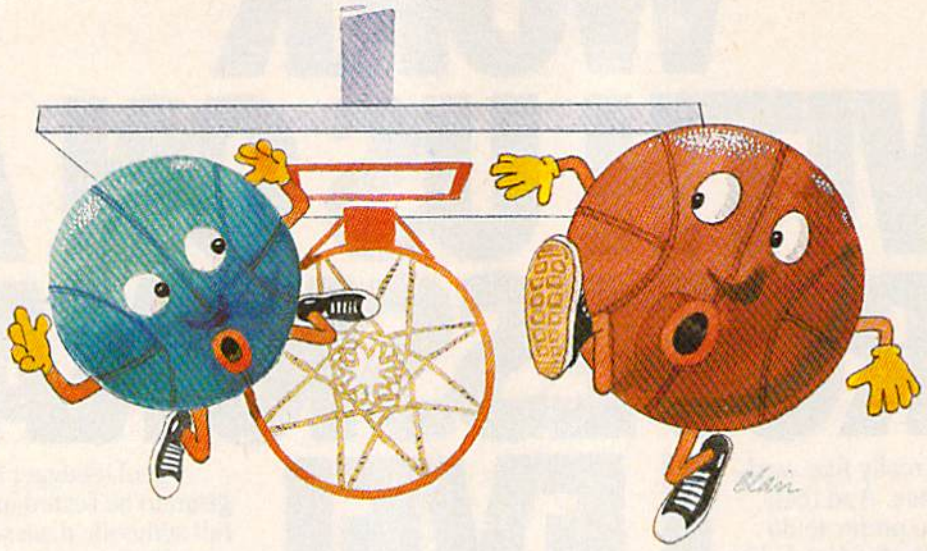
$2.50 US/$5.50 Foreign for shipping and handling. Allow six weeks for delivery.

GEOPROGRAMMER

GEOS COMPATIBLE

Berkeley Softworks

The brightest minds are working at Berkeley.



Basketball Sam & Ed

Rhett Anderson and David Hensley, Jr.

What better way to score a goal than to go through the hoop yourself? This delightfully clever arcade-style game for the Commodore 64 is guaranteed to provide hours of entertainment for young or old. Included are one- and two-player modes. Excellent playability and outstanding graphics and sound make "Basketball Sam & Ed" a must for any game lover. One joystick is required for the one-player mode; two are required for the two-player mode.

"Basketball Sam & Ed" is a whimsical version of one-on-one basketball that can be enjoyed by the whole family, young and old. Although the game is most fun when played by two people, a computer opponent is always ready for the challenge.

Basketball Sam & Ed is written entirely in machine language, so you'll need to use the "MLX" machine language entry program found elsewhere in this issue to enter it. When MLX asks for a starting and ending address, respond with these values:

Starting address: 0801
Ending address: 18C0

Be sure to save a copy to disk or tape after you finish typing. Name the program *SAM & ED* if you wish to be able to use the accompanying Basketball Sam & Ed Customizer program. (See below.)

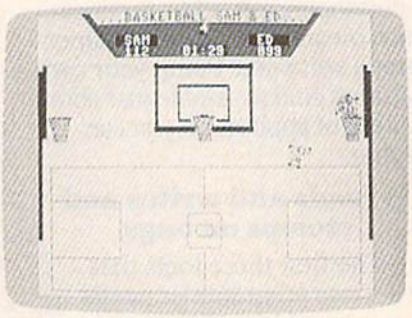
Tip-off

If you plan on playing the two-player version, plug two joysticks into the joystick ports *before* you turn on the computer. If you're playing against the computer, use a joystick plugged into port 2.

Although Basketball Sam & Ed is a machine language program, it can be loaded, saved, and run just like a BASIC program. When you're ready to play, load the program and type RUN. The program prompt asks *ONE PLAYER OR TWO?* Respond by pressing 1 or 2.

At the top of the screen is a scoreboard that shows Sam's score, Ed's score, the period (1–4), and the amount of time left in the period. A scrolling message board provides you with announcements and important information about the game.

Sam and Ed, true to their dedication to the sport, are basketballs—with arms, legs, and faces. Sam is the orange ball and Ed is the blue one. In one-player mode you control Sam, while the computer handles Ed. This mode offers a good way to learn to play Basketball Sam & Ed, since the computer plays a little better than most beginning players. In two-player mode, Ed is controlled by the player using the joystick in port 1. Sam is controlled by the player using the joystick in port 2.



Sam bounces off of Ed to make one last basket in this high-scoring game.

The joystick controls are easy to learn. You can run either left or right by moving the joystick in the appropriate direction. To jump, press the fire button. You can jump

to the left, the right, or straight up into the air, depending on the direction of the joystick when you press the fire button.

As in real basketball, the only way to score is by shooting the ball through the hoop. In Basketball Sam & Ed you can score in any of the three goals. Since you can't jump high enough to score by going through the hoop, you'll have to bounce off of your opponent. It's not uncommon to bounce off your opponent, the floor, and a backboard before scoring—any way that you can make it into the basket counts. As in the real game, each goal is worth two points.

Once you're airborne, you have no control over your path, so you'll have to time your jump just right to score. If you miss with your jump, your opponent may be able to take advantage of your helpless bouncing to score.

The game begins with Sam and Ed bouncing wildly around the screen. When both players land on their feet, the first of four periods begins. Each period lasts for 10 game minutes, which is equivalent to about 2 minutes and 40 seconds in realtime. At the end of each period—when the timer reaches 00:00 and you hear "Sweet Georgia Brown"—the players lose control of Sam and Ed, but if either scores before he stops bouncing, the goal counts.

If the score is tied at the end of the fourth quarter, an overtime period, which is the same length as a regular game period, must be played. You'll be given as many overtime periods as you need to determine a winner.

When the game ends, the screen freezes so that you can view the score. To start a new game, press any key. If no key is pressed after approximately 15 seconds, another game begins automatically.

Press the Commodore key to pause the game. Press it again to resume play. RUN/STOP/RESTORE stops the game. Type RUN to restart.

Strategy

Basketball Sam & Ed looks and plays like an arcade action game, but after playing a few games you'll develop strategies. Most players eventually come up with their own "plays"—ways of timing their jumps so that they score or keep their opponent from scoring. For example, it's possible to block a goal-bound opponent by hitting him with the side of your head as he bounces on the ground. It's also possible to score two baskets in a row after your opponent scores one. Both moves (especially the latter one) demand quick thinking and some practice.

Keep your eye on the clock and don't forget to make that last scoring attempt as time runs out.

Customizing The Game

If you use a black-and-white TV or monitor to play Basketball Sam & Ed, you may not be able to tell Sam and Ed apart. To solve this problem, we've included a customizing program written in BASIC. You can use this program to make your own modifications. (A disk drive is required to use this program.) The screen colors and the body and feature colors of both Sam and Ed can be changed. Three of the scrolling messages can also be changed. You can also make the periods shorter so the game takes less time to play.

Type in Programs 2 and 3 (using the Automatic Proofreader program found elsewhere in this issue), and save them on the same disk as SAM & ED (Program 1). You should save Program 3 with the name S & E CUSTOMIZER, since this is the name Program 2 attempts to load (line 40). Then load and run Program 2, the Basketball Sam & Ed Customizer Loader. This program will first reconfigure memory, then automatically load Program 3. (Do not run Program 3 directly—certain BASIC pointers need to be set by the customizer loader, Program 2.) Follow the instructions and prompts given by the Customizer. When you've finished making your changes, you are prompted to put a disk into the drive. After inserting the disk, press RETURN. A new file called CUSTOM SAM & ED is written to this disk. If the disk already contains a previous version of CUSTOM SAM & ED, the old version will be scratched before the new one is written. The customizer program does not alter the original SAM & ED program file.

See program listings on page 86.

Chain Reaction

Mark Tuttle

In this explosive strategy game, the contest is never finished until your last bomb has been thrown. A joystick is required.

"Chain Reaction" is a clever strategy game for one or two players. Whether you play against the computer or another human, the objective is the same: to eliminate all of your opponent's bomb-shaped pieces from the field of play. The game is played on a 5 × 6 grid of squares, and the players alternate turns, placing one bomb in a square on each turn.

The results of a move depend on how many bombs are already in the chosen square and adjacent squares. Whenever any square reaches "critical mass," it explodes and sends its bombs into neighboring squares. If those squares are already loaded to capacity, they explode too, creating a chain reaction that can engulf a large area of the board.

Type in and save a copy of the program.

Bomb Begets Bomb

When you run Chain Reaction, it begins by asking whether you wish to play with one or two players. If you've never played before, you may want to play a game or two against the computer to learn what sort of strategies it favors. When you choose to play against the computer, the program also asks whether you'd like the computer to take the first turn.

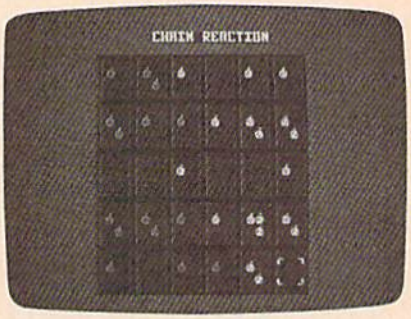
The first part of most games involves placement of initial pieces, without many explosions. As the board fills up, however, explosions occur with increasing frequency. Play continues until one player's pieces are completely eliminated from the board.

The position of a square in the grid determines how many bombs it requires to create an explosion. A corner square can hold a maximum of one bomb. When you place a second bomb in a corner square that already holds one, both bombs explode, sending a bomb of your color into two neighboring squares. After an explosion, the original square is emptied.

Other squares require more bombs to create an explosion. A border square that isn't on a corner can hold a maximum of two bombs. When you place a third bomb in a border square, its explosion sends three bombs into the squares that adjoin it. Squares in the center of the game board hold the most bombs and also create the most devastating explosions. When you place a fourth bomb in a central square, it sends four bombs into squares which adjoin that position.

When an explosion sends bombs into adjacent squares, any bombs in that square change color to match the color of the exploding bombs. Should one of the adjoining squares surpass its limit, that square, too, will explode, creating the potential for even more explosions. This process continues until no more explosions are possible.

Thus, the situation in Chain Reaction is often volatile. The lead frequently seesaws back and forth between players, as each creates increasingly more widespread chain reactions. Even if defeat seems almost certain, you can often regain the lead with clever play. When a game ends, the program announces the winner and permits you to play a new game or quit.

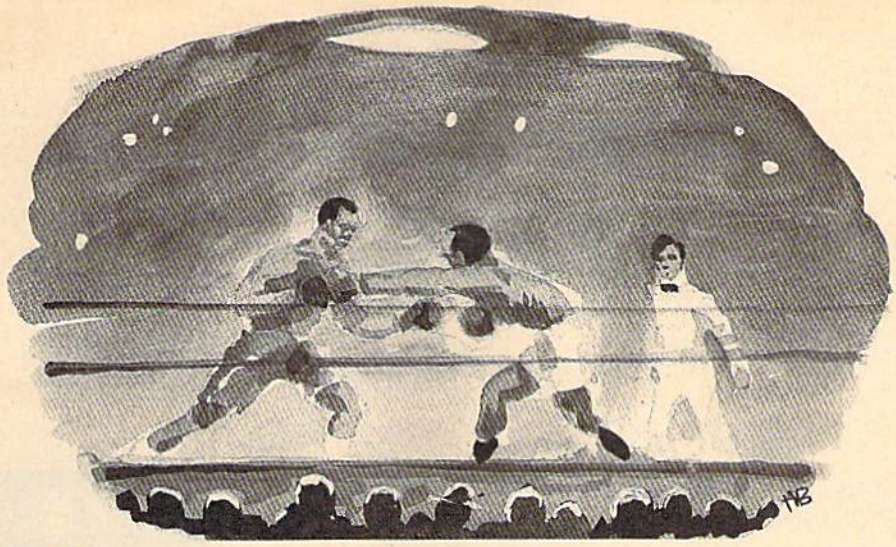


"Chain Reaction" for the Commodore 64, an explosive strategy game.

Like other games of strategy and placement, Chain Reaction rewards the player who can think ahead. At first, you may be tempted to start making explosions as quickly as possible. But that's not always the best long-term tactic. By spreading bombs of your color throughout the board, you may be able to survive chain reactions that would otherwise wipe you out.

The game requires at least one joystick (plug the joystick into port 2 if you are using one joystick). If you choose the two-player option, the program also asks whether you wish to use one or two joysticks. A movable cursor of your color indicates your current position on the board. To place a bomb, move the cursor to the desired square and press the joystick button.

See program listing on page 95.



Ringside Boxing

Anthony Bertram

It's the Friday night title bout in this exciting two-player arcade-style game for the Commodore 64. Two joysticks are required.

The bell rings just in time—you're on your last legs. But after a short breather, you're ready to try again for the heavyweight title in "Ringside Boxing."

Ringside Boxing is a two-player game that requires skill, strategy, and endurance. The match lasts three rounds—unless there's a knockout, of course.

Jabs, Hooks, And Blocks

The game is simple to play—there are only a few moves to learn—but with two evenly matched players it's not easy for either to win. Move the stick diagonally up and toward the other player for a jab. Move it diagonally down and toward your opponent for a hook. To block, push the joystick straight up or down. A push on the fire button executes your command. To move left or right, just push the joystick in the corresponding direction.

Each round lasts for three minutes. The clock at the top of the screen keeps the time. When the bell sounds at the end of the round, the two fighters automatically return to their corners. During the break, the boxers' energy goes up slightly.

Displayed above each player is an energy rating. It's important to keep an eye on this number. If your energy sinks to 25 percent or less, you'll be knocked down with the next punch. If it sinks to 15 percent or less, the next punch landed will be a knockout. Energy is lost two ways: by punching and by getting punched. Throwing a jab costs one point of energy; getting hit costs two. Throwing a hook costs two points; getting hit with one costs four. Energy is constantly being replenished. If you find that yours is getting too low, it's a good idea to block and back away until the end of the round.

If the fight goes the full three rounds, the score will decide the champion. Five points are given for a scoring jab and ten for a hook. In case of a tie, the match goes to the defending champion—the boxer on the left who wears the purple trunks.

In the event of a knockdown, the count clock begins a ten-second count. If the player can't get up in time, the match is over.

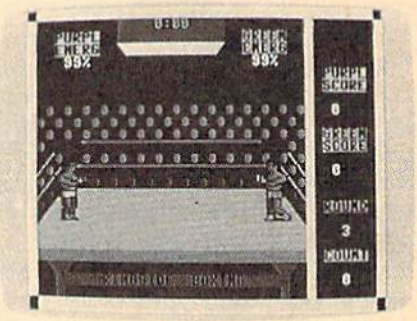
Typing It In

Ringside Boxing is written entirely in machine language. Type it in using "MLX," the machine language entry program found elsewhere in this issue. When you run MLX, you'll be asked for a starting address and an ending address. The correct values for Ringside Boxing are:

Starting address: 0801
Ending address: 1AB0

When you're finished typing in all the data, be sure to save a copy to disk or tape before you exit from MLX. Although Ringside Boxing is written entirely in machine language, it can be loaded and run just like a BASIC program. To load the program, type LOAD"*filename*",8 (tape users should substitute a 1 for the 8 in this statement) where *filename* is the name you used to save the file. Then type RUN to start the game.

See program listing on page 89.



You can employ jabs, hooks, and blocks in this two-player action game for the 64.

Euchre

David Shimoda

Here's a finely detailed implementation of the popular card game of Euchre. The author originally wrote the program in Pascal on an IBM PC. He then translated the Pascal program to BASIC for the Commodore 64. A joystick is optional.

"Euchre" is a four-handed translation of the popular card game of the same name. In this version, you play with a computer partner against two computer opponents. The computer will deal the cards, keep score, and play your partner's as well as your opponents' hands. Even better, it never gets bored or commits blunders such as trumping your ace. Nearly all the subtleties of the original card game are reproduced faithfully, including lone hands, short suits, and more. You can even choose different personalities for your partner and opponents. Type in the program for your computer and read the special-application notes before you run it.

Computer Personalities

The game begins by asking you to choose personalities for your partner and your opponents. Move the reverse-video cursor to your choices, and make selections by pressing the joystick button or the Return key.

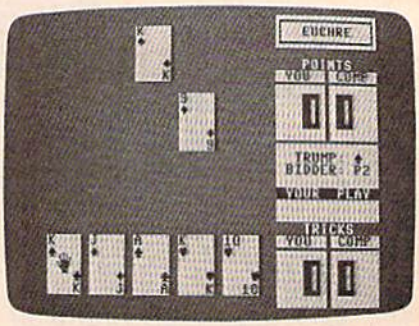
The normal personality plays a more cautious game, while the aggressive personality tends to take more risks. Both opponents must have the same personality, but the partner's personality is chosen separately. This makes the game much more varied than if the computer players always stick to the same, predictable strategy. One of the more difficult combinations is to choose a normal partner and aggressive opponents. Of course, your own style of play will have an impact on which combination you prefer.

Dealing And Trump

This Euchre variation uses only 24 cards from the standard 52-card deck. Each suit includes only the 9, 10, jack, queen, king, and ace. (The ace is high.) Before actual play begins, the first dealer must be selected. This is done by dealing out cards until a black jack is thrown. The first person who receives a black jack becomes the first dealer. After each hand, the position of dealer passes to the next player in clockwise order.

The dealer deals out 5 cards to each player and then places 1 card, face up, on the center of the table. The program automatically deals the cards, as it handles many other details in this game. As a consequence of this scheme, only 21 of the 24 cards are in play for any given hand. (Three cards are always left unplayed.)

The next step is to choose *trump*; the trump suit is the most powerful of the four suits for the current hand. Trump is determined by moving around the table in clockwise order, giving each player an opportunity to choose whether the dealer should pick up the center card. Each player can either pass or *order up*—order the dealer to pick up the center card. When the dealer is forced to take the center card, that card's suit becomes trump, and the dealer discards one card. The computer players, of course, decide for themselves whether to pass or order up in this phase of the game.



"Euchre" for the Commodore 64 (and 128 in 64 mode) reproduces the subtleties of the familiar card game.

If no player chooses to order up in the first circuit of the table, each player then has a chance to pick any other suit as trump. If no player chooses trump on the second circuit, the hand is thrown out completely, and another is dealt.

Lone Hands

On certain occasions, a player may choose to exclude his partner from play, a tactic which is known as playing *lonehand*. The player who chooses trump must choose at the same time whether or not to play lonehand. If a player orders up a card into his partner's hand, the player who ordered up must play lonehand. (If your partner is the dealer and you order up, you must play lonehand).

For instance, you might want to play lonehand in a case where you hold most of the high cards in a suit, and your partner is the dealer, and the center card is a high card of your strong suit. By excluding your partner and playing lonehand, you are in a very strong position to take most or all of the tricks.

Tricks And Hands

A hand consists of five *tricks*. A trick consists of all players laying down one card. The player to the left of the dealer throws down the first card in the first trick. Subsequent tricks are begun by the winner of the previous trick. Suit must be followed within a trick. That is, you must throw a card of the suit which was led, as long as you have any card of that suit.

If no trump cards are thrown in a trick, the trick is won by the player who laid the highest card of the leading suit. If trump is thrown, then the highest trump card takes the trick.

For all suits except the trump suit, the rank of the cards follows the usual order. (The 9 is low, and the ace is high.) For the trump suit, however, the jack is the highest-ranking card. The jack of the same color, but different suit, is considered part of the trump suit—and it is the second highest ranking card. For example, if the trump suit is chosen as clubs, it follows this ranking:

jack of clubs
jack of spades
ace of clubs
king of clubs
queen of clubs
10 of clubs
9 of clubs

A hand is won by the side which wins a majority of tricks (three or more). If you or your partner orders up a card, your side must take the majority of tricks in that hand or else be *euchred*, meaning that the opposite side gets two extra points.

Scoring

A game of Euchre ends when one side accumulates ten or more points. You score one point for winning a hand, two points for winning all the tricks in a hand, and four points for winning all the tricks lonehand.

The game can be played with either a joystick or the keyboard. To play with the keyboard, use the cursor keys to move the colored cursor, and press RETURN to make a selection.

See program listing on page 108.

Miami Ice

Jeff Kulczycki

Here's an action game that challenges both your driving skills and powers of concentration. Originally written for the Commodore 128 with a disk drive, "Miami Ice" has been translated to work on the Commodore 64 as well. A joystick is required.

Ah, Miami—sun city of the South. A sparkling metropolis blessed with a tropical climate, palm trees, beaches, revived art deco architecture, stylish pastels, and classy elegance. Almost paradise.

You wake up on another bright, sunny Miami morning, sip a glass of freshly squeezed orange juice, don your white linen suit and sunglasses, and stroll outside—then get the shock of your life.

What's going on here? Overnight, a freak shift in the jet streams has piped a blistering cold front down from Ohio. The weatherman had predicted a brief shower last evening, but that's not what happened. Instead, the Florida peninsula was blasted by the worst ice storm in 400 years. The Everglades are frozen solid. The pink flamingos are blue. And the streets of Miami are coated with a shimmering layer of slippery ice.

As you start your car—the pampered engine coughs and sputters in the bitter cold—you wonder what it's going to be like driving to work. A Miami native, you've never driven on ice before. In fact, you've never even *seen* this much ice since your boss's retirement party last year, when the caterers made that life-size ice sculpture of Ponce de Leon. You've heard the horror stories told by tourists about winter driving conditions up North, but never thought it could happen to you—not here, in Miami.

The minute you pull out onto the street, your worst fears come true. When you step on the gas pedal, the wheels spin and the car accelerates sluggishly. When you turn the steering wheel, the car slides all over the road. And when you step on the brakes—well, forget it.

You realize, desperately, that you've got to make it to the parking garage across town without smashing your car to smithereens. It won't be easy. But at least there's one thing in your favor—you've got the whole road to yourself. Everyone else, it seems, had the good sense to stay home.

Out Of Control

Despite minor variations, the two versions of "Miami Ice" work basically the same. Using a joystick, you have to drive your car over ice-covered streets to reach the safety of a garage. The joystick button is the gas pedal, and pushing the stick right or left steers the car in the corresponding direction.

But here's the twist—the car doesn't respond instantly to your commands. It tends to slide in the same direction even after you've steered it toward another direction. Then, when you try to recover, you often overcorrect and start sliding in yet another new direction. It's an

inertial nightmare—much like real winter driving.

When you hit a guardrail or some other obstruction, your car cracks up. You get three cars per game. If you reach the safety of the garage, the game isn't over. Instead, you advance to another screen whose streets are even harder to navigate.

The number of points you score depends on how soon you reach the garage. As an incentive to recklessness, a timer starts counting down when you begin each new screen. If you reach the garage, you score the number of points left on the timer. If the timer runs out, you can still reach the garage, but you won't get any points. However, you will advance to the next screen.

Be sure to read the special instructions for each version before typing in the program and playing the game.

Commodore 128

The 128 version of Miami Ice (Program 1) is written completely in BASIC using BASIC 7.0's excellent sprite commands. It runs as fast and as smoothly as the other version, which employs machine language.

Plug a joystick into port 2 and leave a disk in the drive. After each game, if your score ranks you among the top players, the program lets you enter your initials and then saves the high score data to disk.

To complete each level, you merely have to steer your car into the parking garage from any angle. There are a total of four screens, and each screen displays the timer value in the upper-left corner and your current score immediately to the right.

The 128 version of Miami Ice attempts to keep a high score file on the disk. If you want the program to keep track of high scores, be sure to copy the program to a disk that is not write protected.

Commodore 64

The 64 version of Miami Ice is written completely in machine language and must be entered with the Commodore 64 version of the "MLX" machine language entry program found elsewhere in this issue. Be sure you read and understand the instructions for using MLX before you begin entering the data from Program 2. When you first run MLX, you'll be asked to supply a starting address and an ending address. Here are the addresses you'll need for Miami Ice:

Starting address: 0801
Ending address: 1320

Be sure to save at least one copy before you exit MLX. Although the 64 version of Miami Ice is written in machine language, you start the program as if it were written in BASIC: load the program, then type RUN and press RETURN.

Plug a joystick into port 1. To steer your car safely into the parking garage and advance to the next screen, you have to enter the front of the garage without bumping into the black lines which mark its three walls. Indicators on the screen show the timer value and your current score.

There are seven screens in all. The game normally starts at screen 1, but you can begin a new game at any screen you want by moving the joystick up or down to change the screen number. This lets you skip the easier screens as you become a better player, or peek at the hardest screens while you're still a beginner.

See program listings on page 103.

COMPUTE!'s Gazette is looking for utilities, games, applications, educational programs, and tutorial articles. If you've created a program that you think other readers might enjoy or find useful, send it, on tape or disk, to:

Submissions Reviewer
COMPUTE! Publications
P.O. Box 5406
Greensboro, NC 27403

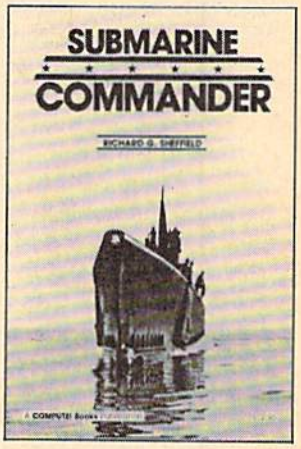
Please enclose an SASE if you wish to have the materials returned.
Articles are reviewed within four weeks of submission.

New from COMPUTE!

War game simulations come alive with two new books from COMPUTE!. Whether you're dodging torpedoes, bombing targets, or planning to defend against an enemy invasion, these books contain the hints and strategies you need for a successful defense—and offense. There's historical information on actual battles, technical information about your war machines, plus many tips on using hidden features of the software.

The Electronic Battlefield
Bob Guerra
ISBN 0-87455-117-X
$12.95

This comprehensive reference book on computer war games—how to play them and what strategies to use—will find a place on thousands of computer owners' desks. Whether acting as fierce opponent or impartial referee, the computer is perfect for war game play. *The Electronic Battlefield* offers general and specific hints, tips, and play techniques for all the most popular computer games. Filled with detailed information on such games as *Silent Service, Roadwar 2000, Field of Fire, Arctic Fox,* and *Balance of Power,* this book will appeal to players of all ages and levels of experience. Also included in this book are undocumented features of games like *F-15 Strike Eagle,* command reference charts for many of the games, and even tips from the game designers. The perfect resource for any electronic war gamer.



Sub Commander: Tactics and Strategy for WWII Submarine Simulations
Richard G. Sheffield
ISBN 0-87455-127-7
$12.95

Run silent, run deep. This hands-on approach to popular submarine simulators like *Silent Service, Gato, Sub Battle,* and *Up Periscope* puts you at the periscope of an American boat in World War II. This collection of undersea warfare tactics and strategies will appeal to every simulator captain. Information about actual World War II submarines, crews, and captains provides an excellent introduction. Detailed accounts of real submarine battles illustrate ways you can pilot your own electronic sub. The heart of the book outlines offensive and defensive tactics and maneuvers that bring you into torpedo range and get you out again safely. Learn how to decimate heavily escorted convoys, how to shoot "down the throat" at a charging destroyer, and how to weather determined-depth charge attacks. You'll be able to immediately apply these techniques and tricks to your own submarine software.

Note: The books do not include any of the software products mentioned.

To order COMPUTE! books, call toll free 1-800-346-6767 (in NY 212-887-8525) or write COMPUTE! Books, Customer Service, P.O. Box 5038, F.D.R. Station, New York, NY 10150. Customer Service hours are 10:00 a.m. to 12:30 p.m. and 1:30 p.m. to 3:30 p.m. EST Monday through Friday.

Please add $2.00 per book shipping and handling. NC residents add 5 percent sales tax and NY residents add 8.25 percent sales tax. Please allow 4–6 weeks for delivery.

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

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company

Omicron

Sean Wagle

Be ready and alert. A host of deadly enemies can make quick work of you in this frantically paced, multilevel arcade-action game for the 64. A joystick is required.

"Omicron" is a game which creates a universe full of hostile aliens. To play and win, you must survive 16 waves of 15 enemies each. This is anything but an easy task—you'll be facing some of the deadliest and fastest creatures ever to visit the screen of your computer.

Omicron is written partly in BASIC and partly in machine language (ML). First, type in and save Program 1, the BASIC section of the program. Since the BASIC section attempts to load the ML portion of the program, don't run the program until you've typed in and saved Program 2.

Typing in Program 2, the ML section of Omicron, requires use of the "MLX" machine language entry program found elsewhere in this issue. When you run MLX, you'll be asked for the starting and ending addresses of the data you'll be entering. For Omicron, respond with the following values:

Starting Address: 33CE
Ending Address: 43BD

When you've finished typing in the data, save a copy to disk before leaving MLX. You must save the data from Program 2 under the name OMICRON ML, since that is the name Program 1 looks for. If you're using tape instead of disk, change the ,8,1 at the end of line 10 in Program 1 to ,1,1. Also, be sure to save the data for Program 2 immediately following Program 1 on the tape.

When you're ready to play Omicron, make sure that a joystick is first plugged into port 2; then load and run the BASIC program. *The game does not work properly unless the joystick is plugged in before you run the program.* When the title screen appears, choose one of the four skill levels by pressing a function key. It's recommended that you start with the Rookie level (f7) so you can get used to game play. The skill levels are as follows:

f1 Expert
f3 Pro
f5 Advanced
f7 Rookie

How To Play

When the game begins, your first ship appears in the upper left corner of the screen. You and your enemies can move freely, but you'll bounce off the edges of the screen and off the status board in the center of the screen, which holds the score, current level, and number of ships remaining.

The first thing to master is the joystick control of your ship. You can fire missiles by pressing the fire button. To move, aim the joystick in the desired direction. Since your ship can build momentum, your direction does not change immediately. The direction in which the ship points does not change unless you press the fire button while moving the joystick, so with careful planning you can move in one direction and shoot in another.

To complete a level, you must eliminate all 15 enemies. If you lose a ship, you'll start over on the same level. A bonus of 1000 points is awarded if you make it through a level on your first try. Bonus ships are awarded after completing levels 4, 8, and 12.

Enemies	Points	Color	Appears as
argon mine	10	cyan	◇
ricochet net	20	green	≡
splinter bomb	50	purple	✶
fusion bomb	100	yellow	F
indestructo	none	yellow	▟
mystery mine	10–100	cyan	?
drifter	200	blue	ʕ
bomber	200	cyan	ʕ
destroyer	400	yellow	ʕ
battle satellite	600	orange	ʕ

Your Enemies

At the beginning of each level, your enemies are dark blue *drifters*. They are harmless at this stage in their life cycle. As time passes, some of the drifters become cyan *bombers*. Bombers leave an assortment of deadly objects in their wake. See the table for a description of all the enemies in Omicron.

If they are not destroyed, bombers become yellow *destroyers*. Destroyers move twice as fast as bombers, and they shoot yellow missiles which are similiar to the ones you fire. Watch destroyers carefully—they can change their direction at the first corner they encounter.

Near the end of the level, the remaining enemies can become orange *battle satellites*, which fire like destroyers, leave bombs like bombers, and bounce around the screen in a furious attack. Needless to say, you should try to kill these off as quickly as possible.

If you wish to pause the game at any time, press SHIFT or SHIFT LOCK. If you use RUN/STOP–RESTORE to break out of the game, you cannot restart play simply by running Program 1 again. Instead, you must turn the computer off and back on, then reload Program 1.

See program listings on page 91.

Solarpix

Simon Edgeworth

Set the solar system in motion with this series of astronomical simulations for the 64. You can also trace the orbits of Halley's and other comets, and look through a file of fascinating facts about our solar system.

Whether you're new to stargazing or a veteran astronomer, this easy-to-use demonstration of the planets and comets is fun to watch. The program is divided into four parts: Sun and Planets, which simulates the movement of the planets around the sun; Sun, Earth and Moon, which shows how the phases of the moon are created by its orbit; Comets, which shows the path of Halley's and other comets; and Facts, which lists the key properties of each planet and the sun and moon.

Type in "Solarpix" using the "Automatic Proofreader," published regularly in the GAZETTE, then type RUN. In the first part, Sun and Planets, you can watch the orbits of up to five planets at once. The sun is stationary at the center of the screen. Select the innermost and outermost planets you'd like to see, and a year from 1 to 2000. Press G to set them in motion, + to increase the speed, − to reduce the speed, and S to stop. Elapsed time is displayed in years on the right side of the screen.

Lunar Lore

The second menu selection, "Sun, Earth, and Moon," shows how the moon's orbit around the earth causes its different phases. The earth is shown orbiting the sun, while the moon orbits the earth. The distance between the earth and the moon has been magnified by 30 to make it visible. In the top right corner of the screen, the current appearance of the moon is illustrated. The elapsed time in days is displayed to the right. You can control the display using the G, S, +, and − keys as in the first part.

The moon takes 27.3 days to orbit the earth. However, it's 29.6 days from one full moon to the next. If you watch the display carefully, you can see why this is—the moon actually has to complete slightly more than one orbit around the earth to reach the same position relative to the sun.

Because the moon's orbit is slightly tilted, the moon, earth, and sun are normally never exactly aligned. On the rare occasions when they are, an eclipse occurs. If this happens at full moon, we see a lunar eclipse. If it happens at new moon, we see a solar eclipse.

Comets

The third menu selection shows what the orbits of comets look like. Planets and comets move in elliptical orbits, which look like flattened circles. The extent to which the circle is flattened is called its eccentricity, ranging from zero (circular) up to almost one. The eccentricities of the major planets range from Venus (almost zero) to Pluto (.25). A comet with an eccentricity of one or more would leave the solar system.

Enter a value between zero and .96 and press RETURN. Here are the names and eccentricities of some comets: Faye (.56), Finlay (.70), Enke (.85). To see a simulation of the orbit of Halley's Comet, enter .96R. The R stands for reverse, because Halley's Comet orbits in the opposite direction of the planets.

Notice that the speed of a planet or comet varies as it moves around its orbit. It moves slowest when it's furthest from the sun, and fastest when it's closest to the sun.

Facts And Figures

The last part of Solarpix provides some interesting facts about the solar system. From the menu, you can select a planet, the sun, or the moon, and view all the facts about it; or you can select a characteristic, such as mass or temperature, and compare the values for all the nine planets plus the sun and the moon. Because the distances from the planets to the sun are so hard to visualize, I've added "Drive Time," which shows the amount of time it would take to drive from a given planet to the sun while traveling at 55 miles per hour. For the moon, the drive time is given for travel from the moon to the earth.

Masses are given in units equivalent to the weight of the earth because the values in tons are too large to comprehend (the earth weighs about 6,588,000,000,000,000,000,000 tons). Densities are compared with water (notice that Saturn is less dense than water). Temperatures are average surface temperatures. Moons are listed in order of distance from their planet, starting with the closest. Moons listed as "X" have not yet been named.

For the amateur astronomer, here are a few technical details. In Solarpix, all orbits are assumed to be on the same plane. For Pluto, however, true elliptical motion is simulated. The planet screen is oriented with Pluto's aphelion on the right, and the positions of the planets are calculated from their actual positions on October 27, 1984. All of the graphics screens show views from ecliptic north.

See program listing on page 78.

Math Dungeon

Richard Lowe

Can you escape the dungeon by finding the key which opens the door? First, you must get by the math monsters who block your way, asking questions about addition, subtraction, multiplication, and division.

At the beginning of "Math Dungeon," you are asked for a difficulty level. Levels one and two deal only with addition and subtraction of positive numbers. Level three adds multiplication and division (still only positive numbers). Four brings in negative numbers, covering multiplication and division. And five asks all four types of questions about positive or negative numbers.

You begin at a gate outside an old mansion. First, find a way through the gate and get to the front door, which is closed. If you can get the door open, you can enter the mansion and you'll be whisked to the underground dungeon. Your adventure begins.

As you wander through the dungeon, you may find different colored keys and doors. The keys are color coded according to the door they open. One of the doors leads out of the dungeon (to end the game).

In your search for the exit, you may also find a map. It can be very helpful. Once you get it, type MAP to see where you are. The map uses R to represent rooms, H for hallways, M for monsters, and * for your location.

The Command Vocabulary

This game recognizes a limited number of words. The verbs are:

GET	Pick up an object
INV	Inventory
LOOK	Look around
OPEN	Open a door
CLOSE	Close a door
ON	Turn on a light
OFF	Turn off a light
MAP	Look at the map
N	Go North
S	Go South
E	Go East
W	Go West

The possible nouns include:

GATE	LIGHT
DOOR	MILKBOX
NOTE	GOLD
KEY	MAP

And if you find yourself in a jam, typing ALGEBRA may get you out. Certain commands need only a single word or letter, like LOOK or N. Others combine a noun with a verb, like GET GOLD. The verb usually goes first.

Information about the dungeon is kept in an array FL(10,10), which is filled with rooms, hallways, and monsters in lines 1620–1850. Lines 1620–1640 randomly place monsters, according to the level chosen at the beginning. Lines 1650–1680 randomly place hallways and rooms throughout the dungeon. Lines 1690–1710 make sure a path is available to each monster. Lines 1720–1770 randomly place colored keys and doors around the array.

See program listing on page 81.

Vocab Builder

William J. Treanor

This program not only helps increase vocabulary, but also assists students who must memorize words for spelling, biology, history, or other classes. Up to 100 words and definitions can be entered and saved to tape or disk.

My son's sixth grade teacher regularly distributes lists of words and their definitions to the students. The children then take a spelling test at the end of the week.

"Vocab Builder" was originally written to help my son prepare for these tests. But it was soon obvious that it could be used for other subjects—a history drill, say, with facts and dates in place of vocabulary words. Vocab Builder has even been used by college students learning medical terminology. The program flow is fairly straightforward; modifications (using a printer, for example) should be easy to make.

Four Menu Options

Enter the program, save it, and type RUN. First, you are asked to input your name, which is printed at the top of the screen.

Next, you'll see the main menu, which gives you four options: Enter new words, Study, Test, or End.

If you choose to enter new words, you are asked how many words will be in this particular file. You must then type in the words and their definitions. If you make a mistake, don't worry, you'll have a chance to correct it later. When all words and definitions are entered, they are printed on the screen. You can fix mistakes at this point.

The program then saves the word list as a sequential file on tape or disk. You are prompted for the date, which becomes part of the filename. (Note: Since the date is part of the filename, and the filename must be unique, be sure to use a different filename if you create a second quiz on the same day. For example, if you create a quiz named 11-21-84, use 11-21-84.1.) The program then returns to the main menu.

The second option from the menu allows you to study words which were previously entered. First, you enter the date of the test and the file is loaded from tape or disk (depending on your response, T or D, to the prompt).

The list is put into random order. A definition appears on the screen and the student is given three chances to enter the corresponding word. Spelling is important. If the word is correct, the student is congratulated. If the first two letters are correct but the word is misspelled, the message YOU'RE CLOSE appears.

When all words have been covered, the score (number of answers right and wrong) is displayed. Any word which was answered wrong on the first attempt is counted as incorrect. The student is then given a list of which words were incorrect and need further study.

The test option is similar to the studying option, but the student has only one chance to supply the correct word.

The final option, End, allows you to exit the program.

A Tireless Teacher

For young children, Vocab Builder may help develop a sense of responsibility. They can choose when to study and how long. They don't have to wait until a parent or sibling has some free time to help them with schoolwork.

The child is addressed by name, so it becomes his or her computer program. A computer doesn't tire of helping the child, or lose its patience. And it does not chastise the child for answering incorrectly. Words are randomized automatically (so the student actually has to learn the words rather than memorize them in order).

Memorizing can be a chore. Vocab Builder can help relieve some of the tedium.

See program listing on page 121.

Hi-Res PRINT
For Commodore 64

Scott M. Petty

This short machine language utility allows you to quickly print characters on the Commodore 64's high-resolution graphics screen.

How many times have you thought of an idea for a game, replete with high-resolution screens, colorful animated sprites, and onscreen scoring and timing? Many different utilities are available for drawing shapes on a hi-res screen, but most of them omit one important item: printing text. Of course, you can always copy character patterns onto the hi-res screen in BASIC, but the process is painfully slow. "Hi-Res PRINT" is a short machine language routine which allows you to print letters and numbers anywhere on a hi-res screen, using different colors and reverse mode if desired. Because it's done in machine language, the process is as fast as using normal PRINT statements in BASIC. And you can use the routine from BASIC, without having any machine language knowledge.

Typing The Programs

You'll need to type in four short programs. Program 1 is the machine language (ML) program itself. Program 2 creates an abbreviated character set for use by the ML program, Program 3 demonstrates hi-res character printing, and Program 4 is used to relocate the ML to a different memory area.

Begin by typing in Program 1 with the "MLX" machine language entry program found elsewhere in this issue. Read the MLX instructions carefully and be sure to save a copy of the program when you are done typing. Here are the starting and ending addresses for Program 1:

Starting address: C000
Ending address: C20F

It is *important* that you save this program with the filename HRPRINT so that the other programs can load it by that name.

The ML program will need a set of character patterns to use for printing on the hi-res screen. Program 2 (CHARSETMAKER) is a BASIC program that makes a disk file containing data for the first 64 characters of the Commodore 64 character set. Type in Program 2 and save a copy.

The character file which you create with Program 2 will load at any address that you specify. For now, supply the address indicated below so that you can use this character set with the demonstration program (Program 3). Run Program 2. When it asks for the address of the character set, enter the number 16384 and press RETURN. The program then asks you to name the file in which this character set will be stored. Type in HRCHARSET and press RETURN. Again, it is important to use this particular filename so that Program 3 can load the file from disk. After you answer the second prompt, the program writes the file to disk and ends.

Type in and save Programs 3 and 4, then load and run Program 3 (DEMO) for a demonstration. The program begins by loading the files HRPRINT and HRCHARSET into the correct memory locations and by clearing the hi-res screen. Then it draws a sine wave to prove that you are indeed looking at a hi-res screen. Finally, the program prints several text messages in different colors. In the left portion of the screen are two example score and timer displays which continue to update as long as the program runs. To end the demonstration and return to the normal screen, press any key.

Loading From A Program

Let's learn how to use the machine language routine by observing how Program 3 handles it. Several preparatory steps are required. First, the BASIC program must load both the ML code and the special character set which it uses. Program 3 does this in lines 110–120:

```
SX 1000 GRAPHIC 1,1:FORJ=150 T
        O 20 STEP -30:CIRCLE 1
        ,158,100,J,75:NEXT
CH 1010 DRAW 1,158,25 TO 158,1
        75:DRAW 1,70,40 TO 246
        ,40:DRAW 1,23,68 TO 29
        2,68
QB 1020 DRAW 1,8,100 TO 308,10
        0:DRAW 1,23,132 TO 292
        ,132:DRAW 1,70,160 TO
        {SPACE}246,160
SJ 1030 CHAR 1,17,2,"EARTH"
```

These lines should appear at the very beginning of the program, before any other BASIC statements (except REMarks, which the computer ignores). If you are not familiar with how LOAD works in program mode, these lines may look somewhat baffling. Here is an explanation of how they work.

When it executes a LOAD statement under program control, BASIC performs the load and then reruns the program from the beginning. However, BASIC remembers the values of variables that were previously used in the program. Thus, the first time you run Program 3, the variable A is set to 1, and the computer loads the ma-

chine language file HRPRINT. After the load is complete, the computer runs the program a second time, beginning again with the first line of the program. But now A is equal to 1, so the IF test in line 110 fails, and the computer proceeds to line 120. The variable A is set to 2, the computer loads the file CHARSET, and the program is run from the beginning for a third time. This time both IF tests fail, and the computer goes on to execute the remainder of the program.

Locating The Hi-Res Screen

The next step is to decide where to put the hi-res screen. High-resolution screens require two separate blocks of memory. The largest block, called the bitmap, is 8000 bytes in size; it contains information about which pixels (screen dots) are turned on and which are off. The second block is 1000 bytes in size; it contains color information for each of the 8 × 8–pixel blocks in the bitmap. The computer combines pixel information from the first block and color information from the second block to produce the final picture which appears on the hi-res screen.

The 64's video chip can refer to addresses only within a 16K (16,384-byte) memory zone. As a result, you must always locate the 8000-byte bitmap and its 1000-byte color-memory block within the same 16K area. The Commodore 64's memory can be divided into four such blocks, which are known as *video banks*:

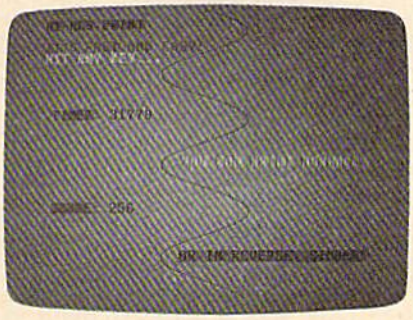
Bank 0:	**0–16383**	**($0000–$3FFF)**
Bank 1:	**16384–32767**	**($4000–$7FFF)**
Bank 2:	**32768–49151**	**($8000–$BFFF)**
Bank 3:	**49152–65535**	**($C000–$FFFF)**

Program 3 locates the bitmap and color memory in video bank 0. The bitmap will start at location 8192 and color memory will go at 1024, the same area used as screen memory in text mode. Line 200 tells the computer the bitmap's location, and line 220 puts the machine in hi-res bitmap mode.

The following shows where Program 3 puts the bitmap, color memory, and character set:

Location	Usage
1024–2023	**color memory**
8192–16383	**bitmap**
16384–16895	**character set**

One disadvantage of using video bank 0 is that it locates the bitmap in the middle of the RAM (Random Access Memory) area normally used to store BASIC programs. BASIC memory ordinarily resides in locations 2049–40959, giving you 38,911 bytes to hold a BASIC program and its variables. In this case, however, it is restricted to locations 2049–8191, leaving only 6141 bytes of BASIC program space. What's more, the program takes no steps to protect the bitmap; if you expand the program, it will eventually encroach on the bitmap



If you think it's difficult to print characters on the Commodore 64's hi-res screen, look again. "Hi-Res PRINT" is a convenient machine language utility which prints letters and numbers at any location on a hi-res screen. No machine language knowledge is necessary to use this program.

area, corrupting its contents. For longer BASIC programs, you may need to use a different video bank—a subject that is beyond the scope of this article. In *Programming the Commodore 64*, available from COMPUTE! Books, you can find detailed explanations of video bank usage and methods for creating a protected memory zone.

Setting Up

Once you have loaded the machine language program and its character set, located the hi-res screen, and enabled hi-res mode, you're nearly ready to print characters on the screen. Before you can use the ML program, however, you must tell it what sort of printing to perform, as well as the location of the bitmap, color memory, and character set. Line 150 sets two important variables—IN and PR—which are used in later SYS statements. The variable IN stands for initialize. This variable is used in a SYS call which passes the initializing information to the ML program. The first initializing statement occurs in line 240:

```
240 SYS IN, 2, 8192, 16384, 1024
```

The first value in every SYS statement is the address of the ML code which you wish to execute. Since we set IN to 49152 (line 150), this SYS statement transfers control to the ML instructions beginning at location 49152. If the ML code loads at location 49152, you should always set IN to 49152. If you relocate the code to a different memory area (see below), IN should be set to the new starting address of the ML code.

This particular ML program is designed to retrieve additional information which appears after the SYS address. The last three values in line 150 should look familiar: Here the numbers 8192, 16384, and 1024 indicate the location of the bitmap, character set, and color memory, respectively. When you execute this SYS statement, the ML program stores this information for future reference.

The second value in the SYS statement (2, in this case) is a special flag for the ML program. This value must be either 1 or 2, depending on what sort of printing you desire and what sort of hi-res screen is in use. For normal hi-res (not multicolor) screens and normal printing, this value should be 1. If you are using a multicolor hi-res screen, or if you wish to have reverse printing on a normal screen, set this value to 2.

Of course, any of the values in the SYS statement can be replaced by numeric variables. For instance, BASE is set to 8192, so you could replace the 8192 in line 240 with BASE.

Once you've initialized the program for use, the next step is usually to clear the hi-res screen. If you don't do this, it will contain random shapes. Clearing the bitmap in BASIC is a time-consuming chore, since you must POKE zeros into 8000 consecutive memory locations. Line 270 does the job in less than a second by using the ML program to print reverse spaces on the entire hi-res screen.

Clearing the screen is a very simple matter. Line 260 creates a string, F$, which consists of 40 spaces. In line 270, the program prints F$ 24 times, once for each character row on the screen. Here is the SYS statement that prints the

string:

```
SYS PR, 15, 0, J, F$
```

Once again, the first parameter after SYS is an address within the ML program. The variable PR stands for PRINT; it is set to 49207 in line 150. The second value sets the printing color (15, in this case, for light gray). The third and fourth parameters set the character column and character row, respectively, where the first character of the string is printed. In this example, we always start printing at column 0, the leftmost column of the screen. The fourth value is set by the variable J, which the FOR-NEXT loop in line 270 causes to cycle from 0 to 24.

The last parameter tells the ML program what to print. On this occasion it is a string variable (F$). You can also use any string or numeric expression that requires no more than 80 characters to print. Here are just a few examples:

```
SYS PR, 15, 0, 0, "HELLO"
SYS PR, 15, 0, 0, LEFT$(A$(12),1)
SYS PR, 15, 0, 0, SIN(TI)*TI↑2
```

The rest of the SYS statements in Program 2 print messages on the screen, reinitializing the ML program as needed for various kinds of printing. By examining and experimenting with these lines, you will see how to use the ML routine in several different ways.

Relocating The ML Program

Program 4 is not immediately necessary, but you may want to type it in for future use. Like many other machine language programs on the Commodore 64, Hi-Res PRINT normally occupies the protected memory area beginning at location 49152. It may happen that you wish to use this program with some other ML utility which also loads at that address. If so, you can run Program 4 to create a copy of the ML program which loads and runs at a different memory address.

Before using Program 4, you must decide on a new location for the ML program. This can be any free RAM area at least 524 bytes in size which is not otherwise in use at the time. When you run Program 4, it loads the HRPRINT file at its normal memory location, and then asks you to enter the new memory address and the filename to use for the new file. Of course, you should use some filename other than HRPRINT for the new file. (When loading the new file under program control, you would then substitute the new filename for HRPRINT.) The program adjusts all of the ML program's internal addresses for the new location and writes the new file to disk.

As a convenience, Program 4 also prints the new addresses to use for IN and PR in the SYS statements for this program (see above). Make a note of these addresses and be sure to set IN and PR accordingly when using the relocated version.

In addition to the memory occupied by the ML code itself, this program stores information in the following memory locations:

679–767	($02A7–$02FF)
820–827	($0334–$033B)
1020–1023	($03FC–$03FF)

You should take care not to POKE into these locations or otherwise disturb their contents when using this program.

See program listings on page 125.

Text Framer

Paul Sawyer

Frame text or character graphics with this utility that lets you easily create any size and color of frame and put it anywhere on the screen.

"Text Framer" is a small machine language (ML) program that can greatly improve the appearance of your BASIC programs. It lets you draw frames on the screen at ML speed—frames of any size, any color, and made up of any characters that you choose. You can make your own pop-up windows, divide the screen into sections, or frame the entire screen.

Versions are included for the Commodore 128 and 64. (The 128 version can draw frames only on the 40-column screen, not on the 80-column display.) The program is a BASIC loader. Since it contains a machine language program in the form of DATA statements, use "The Automatic Proofreader" program located elsewhere in this issue to insure accurate entry. Type in the version for your computer and be sure to save a copy to tape or disk.

When you load the program and type RUN, the program will be POKEd into memory, but nothing else will happen. To use the program to make a frame, you must use a SYS command to call the ML routine. Before calling the routine, you'll probably want to set parameters appropriately. The accompanying table shows the key addresses for each machine.

The default values in the table are the numbers that are set when the program is first run. If you do a SYS 49152 on the 64 or a SYS 3072 on the 128, you'll get a frame with an upper-left corner at 0,0 and a lower-left corner at 11,11. (The length parameters do not include the edges of the frame, so a frame that surrounds a 10 × 10 square is really 12 × 12.) The color of the frame will be white (1), and the frame will be made of the default parameters (straight lines with rounded corners).

The program uses line numbers 10–350, so begin your own program at line 360 when adding frames to your own displays.

Let's change the appearance of the frame by making the horizontal character an equal sign. We'll make the change for the Commodore 64. If you're working with the 128, use the table to find the correct location to POKE. Now, add this line to the program:

```
360 POKE 49160,61: SYS 49152: REM 61
    IS SCREEN CODE FOR =
```

Now run the program.

You can change any parameter simply by POKEing the desired value into the proper location. The character numbers (49160–49165) are screen codes, not ASCII codes, so check your programmer's guide or COMPUTE!'s *Mapping the 64* for the appropriate values.

The program does error checking, so you don't have to worry about crashing your computer when you use it. If the border won't fit on the screen, the computer will give an illegal quantity error.

See program listings on page 97.

128	64	Default	Purpose
3072	49152		SYS address
3075	49155	0	top left X coordinate
3076	49156	0	top left Y coordinate
3077	49157	10	horizontal length
3078	49158	10	vertical length
3079	49159	1	color (0 on Plus/4, 16)
3080	49160	64	screen code for horizontal frame character
3081	49161	66	screen code for vertical frame character
3082	49162	85	screen code for top left corner character
3083	49163	73	screen code for top right corner character
3084	49164	74	screen code for bottom left corner character
3085	49165	75	screen code for bottom right corner character

Snapshot

Clayton R. Newman

This clever utility can store up to 56 different screens in memory and instantly retrieve any one for display. For the Commodore 64 and 128 (in 64 mode).

Here's a program that can be a real boon to programmers. "Snapshot" is a machine language utility that stores whatever you have on the text screen, including colors, into the unused RAM which underlies BASIC. It can store and retrieve up to 56 screens. And you can save the screens to tape or disk. It crunches the information for the screen, storing up to 255 empty spaces or repeated characters in only three bytes.

Since Snapshot is written entirely in machine language, it works fast. It can print a screen more than twice as fast as the equivalent BASIC PRINT statements, and far faster than POKEing data to screen memory.

Two Modes

"MLX," the machine language entry program (found elsewhere in this issue), is required to type in Snapshot. After loading and running MLX, answer the prompts for the starting and ending addresses with the following values:

Starting address: C124
Ending address: C983

After typing it in, save a copy.

Snapshot can be used in either program or immediate mode. The commands are slightly different in each mode. (See the accompanying table for a list of commands.) Immediate mode uses the CONTROL key to input commands. In program mode, several SYS commands are required.

To use it in immediate mode, load Snapshot by typing LOAD "*filename*",8,1 (disk) or LOAD "*filename*",1,1 (tape), then NEW and SYS 49444. Then, create or display the screen you want copied. When the screen is ready, press CONTROL-C. Snapshot copies the screen to memory and asks you for a screen name. All Snapshot screen names must be two characters long. Any two characters, except asterisks, may be used. If you choose a name that's already been used, the old screen will be replaced by the new one. Use the INST/DEL key to make corrections when entering the name. If you decide not to save the screen at all, just press RETURN with no name and it will cancel the command. This escape works with all commands.

To recall a screen, press CONTROL-P and enter the two-character screen name. CONTROL-I displays an index or directory of your screen names. To delete a screen, press CONTROL-D and enter the name. To delete all of your screens, enter an asterisk (*) as your screen name. (Individual screen names should not contain an asterisk.)

CONTROL-S and CONTROL-L are used to save and load the entire group of screens on tape or disk. You're asked for a filename—for the entire group of screens—and whether you're using tape or disk. (Tape users: you must enter a filename, since pressing RETURN with no input aborts the command.)

If you wish, you can copy only part of the screen. To do this, position the cursor at the start of the section you want to save and press CONTROL-F. The cursor changes color. Now move the cursor to the position after the last character you want saved and press RETURN. When these fractional copies are printed, they're returned to the screen in the same location from which they were saved.

If you're working with graphics, you may often want to change the screen location or redefine the character set. Snapshot will work in either case, but remember that screens are printed to the same screen location from which they were originally copied. You could set up several screens ahead of time and scroll through them by changing screen locations. But remember that all screen locations use the same color memory, so if the screen being printed has different colored characters than the screen currently displayed, the colors on the current screen will change too.

Snapshot Commands

Command	Immediate mode	Program mode
Copy	CONTROL-C	SYS 51000,C,screen name
Print	CONTROL-P	SYS 51000,P,screen name
Delete	CONTROL-D	SYS 51000,D,screen name
Copy fraction	CONTROL-F	not available
Index	CONTROL-I	SYS 51000,I
Load	CONTROL-L	SYS 51000,L,filename,D (or T)
Save	CONTROL-S	SYS 51000,S,filename,D (or T)

Disable changing special color registers: POKE 50620,1 Reenable: POKE 50620,0

The program mode commands are very similar to immediate mode. The format is SYS 51000, *command, name*. For example, to print a screen named AB, use SYS 51000,P,AB (P is for Print). You can use a string variable for the screen name, as in SYS 51000,C,A$. Array variables cannot be used, so if you want an array of screen names you must assign the array variable to a regular string variable: A$=B$(20). For loading or saving, add T (tape) or D (disk): SYS 51000,S,*filename*,D. Do not use quotes for the filename. Also, you don't need to SYS 49444 (the command that activates Snapshot) in program mode.

Making It Useful

Snapshot has numerous applications. One is displaying backgrounds for arcade games or text adventures. By using a redefined character set, very complex screens can be drawn, rivaling high-resolution screens. Many commercial games use screens drawn this way. One advantage of Snapshot is that it uses a maximum of 1509 bytes to store a screen while a hi-res screen uses 9000 bytes. Another use for Snapshot is for menus, instructions, and title screens.

Snapshot is transparent to most BASIC programs. This means you can use it in immediate mode to copy screens of programs while they're running.

Snapshot saves the screens in RAM underneath the BASIC ROM at 40960–49151, providing 8192 bytes of storage, enough to store six screens using the maximum size of 1509 bytes per screen. The minimum size for a whole screen is 27 bytes. There is enough room for 327 screens this size, but Snapshot can only handle up to 56 screens at a time.

The reason that different screens require different amounts of memory is that Snapshot compresses the screen as it is saved. Any screen character repeated three times or more is represented as the character code along with a count of how many repeats. Color memory works the same way; however, since 16 colors can be represented in half a byte, two color locations are crunched into each byte to save space. This means that a screen which is mostly empty space, repeated characters, or the same color, uses much less space than a screen with a complex image.

If you run out of memory, Snapshot will display an error message. You can extend your Snapshot memory by lowering the top-of-BASIC pointer in locations 55 and 56. Type POKE 56,PEEK(56)−4*X:CLR where X is the number of kilobytes of added memory.

When Snapshot prints a screen, it also changes the background and border colors to those that were saved. If you use extended background color mode, it will print the extra background colors. It also saves the extra colors for multicolor characters. Sometimes, however, you might not want these changes. To disable this feature, POKE 50620,1. To reenable it POKE 50620,0.

Program 2 demonstrates a little of what Snapshot can do. The program sets up the screen, and then Snapshot quickly displays several screens, simulating animation.

See program listings on page 76.

DOS Window

Bob Kodadek

If you own a Commodore 64 and 1541 disk drive, you won't want to miss this timesaving utility. It displays a window with 11 disk commands—any of which is just a keypress away—without affecting your current screen or BASIC program.

Have you ever wished for a better and easier way to perform disk operations? I'm not talking about the "DOS Wedge" that you probably already have. I mean something even better than that.

Imagine a menu screen that would provide a single keystroke command to perform a particular disk operation. The program could prompt at the appropriate time for user input such as new filename, old filename, disk ID, and so on. It would no longer be necessary to remember or enter lengthy command strings. The directory could be read without the nuisance of screen scrolling. The error channel would be displayed automatically after every operation. And, by using a windowing technique, we could do all this and more without affecting our present screen. While we're at it, let's imagine that this command menu could be accessed, viewed, and used at any time, even while another program is running. A pretty tall order, you say?

The accompanying program, "DOS Window," is just what we described—a menu-driven, interrupt-based, DOS support window. It is RAM-resident and transparent to your other programs. It can be called at the touch of a key to perform disk operations instantly, even while other programs are running. Included in the display is a digital clock for accurate timekeeping. The following commands are available from the menu:

```
D:  Directory
I:  Initialize
V:  Validate
R:  Rename
N:  Format
←:  Save
C:  Copy
S:  Scratch
L:  Load,8
↑:  Load,8,1
U:  Reset
```

Using DOS Window

DOS Window consists of two programs. Program 1, "DOS Window Object File," is written in machine language. "MLX," found elsewhere in this issue, is required to type it in. After loading MLX, respond to the prompts with the following:

```
Starting address:  CA60
Ending address:    CFFF
```

Be sure to save a copy of Program 1 to disk after you've typed in all the data. *(Note: Program 2 attempts to load this file as "DOS-WINDOW .OBJ". If you save Program 1 under any other name, be sure to change the filename which is loaded in line 10 of Program 2.)* Program 2, "DOS Window Boot," is written in BASIC. Be sure to save a copy before you run the program—it erases itself when it's performed all its duties—and be sure to save it on the same disk on which Program 1 was saved.

To use DOS Window, just load and run DOS Window Boot, Program 2. (It automatically loads the machine language in Program 1.) When the program is finished loading, you may elect to set the time on the digital clock. If the current time is not set, the clock will function as a timer. To open the window, press CTRL-D. Then simply select from the menu. When you're through, press RETURN to close the window. To disable the utility, press RUN/STOP–RESTORE. To reactivate it, type SYS 51808.

How It Works

The IRQ vector is altered to point to a routine which scans the keyboard for the current key being pressed. If the CTRL and D keys are pressed simultaneously, a flag is set to divert subsequent interrupts. Then screen memory, color memory, and zero page are quickly copied to the normally unused RAM under the BASIC ROM. Next, the menu screen is displayed and the program enters an input loop. If the input is valid, it is matched to a corresponding command routine and executed. Following the completion of each operation, the error channel is read and displayed at a status line. When the menu screen is exited, all stored memory is returned to its original location, the flag is cleared, and control is returned to the current program running in memory or BASIC, whichever applies.

To avoid conflicts between programs, some restrictions must be observed. Your program may not use the RAM located under the BASIC ROM or locations 51808–53247. Also, the DOS Window program assumes that you're in the standard character mode with the screen at its default location of 1024.

See program listings on page 96.

DOS Calc

Steve Kelly

This convenient, menu-driven utility for the Commodore 64 includes all the disk functions of the familiar "DOS Wedge" program, plus a powerful scientific calculator. A disk drive is required to use the program's DOS functions.

"DOS Calc" is two programs in one. Not only is it a complete replacement for the "DOS Wedge" program supplied on the 1541 or 1571 *Test/Demo* disk, but it has a built-in scientific calculator. The program is entirely menu-driven and employs easy-to-use, pop-down windows.

DOS Calc is written in machine language, so you must enter it with "MLX," the machine language entry program found elsewhere in this issue. Here are the addresses you need to type and save the program with MLX:

Starting address: 0801
Ending address: 16E8

After you have saved a copy of DOS Calc, it can be loaded and run just like any BASIC program. When you run it, DOS Calc relocates itself to the safe memory area beginning at address 49152, a process that takes only an instant.

Once the READY prompt returns, DOS Calc is ready to use. Like the "DOS Wedge," DOS Calc works only in immediate mode (when you are not running a program). To enter DOS Calc, type @ and press RETURN. DOS Calc displays the main menu window with the following selections:

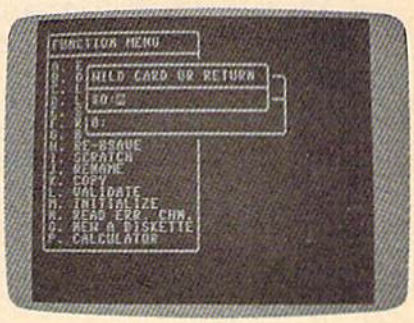
A. Exit
B. Directory
C. Load
D. Load relocated
E. Save
F. Resave
G. Bsave
H. Re-Bsave
I. Scratch
J. Rename
K. Copy
L. Validate
M. Initialize
N. Read error channel
O. New a disk
P. Calculator

To select a function from the main menu, simply press the letter shown next to the function you want. For instance, pressing A exits DOS Calc and returns you to BASIC. When you choose any other function, DOS Calc opens a new screen window in which the interaction for that function takes place.

Directory

The directory function (B) is perhaps the most frequently used DOS function. When you press B, DOS Calc opens a window and prompts you to enter a wildcard specification. As a convenience, it prints the asterisk (*) wildcard in the window. If you wish to view all the files on the disk, simply press RETURN. To view only selected files, change the wildcard accordingly.

For example, suppose that you want to see all the files beginning with GAME. Type GAME* and press RETURN. After you enter the wildcard, DOS Calc opens a third window which contains all the requested directory information. To exit this window and return to the main menu, press A. If the disk contains more files than will fit in the window, you can press B to view the next page of the directory. Press C to enter a new wildcard and begin a new directory search.



"DOS Calc" is entirely menu-driven and employs easy-to-use, pop-down windows.

File Functions

The next eight DOS functions all concern existing files. They are Load, Load Relocated, Resave, Bsave, Re-Bsave, Scratch, Rename, and Copy. After you select any of these functions from the main menu, DOS Calc asks you to enter the name of the file you wish to manipulate. If you enter a null filename (no name), DOS Calc automatically displays the directory for the current disk. This directory differs from the normal directory display, however, in that each filename is prefaced by a letter. To select a file for the current function, simply press the letter key that appears in front of the filename. If you change your mind and decide not to complete the operation, press A to exit.

For instance, let's try the Resave command, which deletes an existing program from disk and replaces it with the BASIC program currently in memory. When you choose Resave from the main menu, DOS Calc opens a window asking for a filename. If you press RETURN

without entering a name, DOS Calc opens a secondary directory window from which you can select the file by pressing a single key. Once this is done, the program completes the resave. You should avoid using any wildcards in the directory function when it appears as part of a Resave or Re-Bsave command.

The Copy and Rename commands require two filenames. The first name you enter is used as the name of the new file for the Copy command or the new name of the existing file for the Rename command. This name must be typed in (it cannot be derived from the directory). The second name indicates the original file; this name can be taken from the directory.

DOS Calc provides two functions that are not included in the "DOS Wedge" program. The B in Bsave and Re-Bsave stands for binary, indicating a binary file containing machine language, graphics data, or something other than a BASIC program. Since these functions save a designated area of memory, you must supply two addresses: the beginning and ending addresses of the area to be saved.

The Validate, Initialize, and Read Error Channel commands require little explanation. Since no information is required for these commands, DOS Calc simply performs them without any introductory prompts. The New command, which formats a disk, requires either one or two items of information. The first item—the disk name—is mandatory. To reformat an already formatted disk, you can supply a new disk name without anything else. To format a disk that has never been used, you must supply a two-character disk ID after the disk name, separating the two items with a comma.

For additional information on DOS operations, refer to the user's manual for your disk drive.

Scientific Calculator

DOS Calc also includes a built-in calculator which is very useful for math, engineering, and various scientific applications. This calculator is different from the familiar pocket calculators used by most people. It uses a notation called *Reverse Polish Notation* (RPN), which allows you to solve complex formulas without parentheses.

When you select the calculator from the main menu, two secondary windows appear on the screen. On the right is the function window, which lists all the calculator functions preceded, as usual, by the letter key which invokes them. To the left is the result window, which is subdivided into three sections. The result area contains either answers from calculations or keyboard input from you. In the middle is the stack display, which holds intermediate results. The bottom area is a message area where the calculator displays error messages such as *Divide by Zero*.

Notation

Before looking at the calculator functions, let's establish some rules for entering numbers. You can enter any number in either standard notation (2534.56 is an example) or in scientific notation (123E34 is an example). Negative numbers are entered somewhat differently than you might expect, since the minus (−) sign is already used to signify subtraction (see below). Use the shifted minus sign (hold down SHIFT and press the minus key) as a substitute. The second trick has to do with the *E* symbol used in scientific notation. This calculator also uses the letter *E* for another function (it causes the stack to wrap down; see below). As a substitute, use the ↑ symbol (the up-arrow key directly to the right of the asterisk). Whenever you enter ↑ as part of a number, DOS Calc puts an *E* in the number you are entering.

Examples

To start with an easy exercise, let's look at how to perform 2 + 5. In RPN there is no need for an equal sign (=). In this case, you enter both numbers followed by the plus sign (+), which is the operator you wish to use. Press the number 2. DOS Calc prints a 2 in the result window. Now press RETURN. The number 2 shifts to the right and also appears as the top entry in the stack area. Press the number 5. The top line of the result window now shows that number. Press the plus key. The 2 on the stack is replaced by a 0 and the result line displays a 7. Simple arithmetic operations use the same general procedure. To subtract 2 from 5, you would enter the numbers 5 and 2, followed by the minus key.

More complex equations demonstrate the power of an RPN calculator. Let's solve the equation 2048/(8*4). Begin by entering 2048 and pressing RETURN. Now type 8, followed by RETURN, and then a 4. At this point the stack contains all the numbers for the equation, in correct order. Now press the asterisk (*) key to perform multiplication. The result line should display 32. Press the slash (/) to perform division. The calculator displays the final result, which is 64.

Each time you pressed RETURN in this example, the calculator pushed the number from the result line onto the stack. All other numbers were pushed down one location, and the fourth entry on the stack was lost.

There are seven other functions which, like RETURN, have some effect on the calculator's stack. The first is Clear Entry: This function clears the result line and leaves all other entries untouched. The Clear Stack function clears the result line and the entire stack. The next four functions cause the entire stack, including the result line, to *wrap* or *roll* in either direction. A wrap allows either the top or bottom entry to move from top to bottom, or vice versa. A roll causes the top or bottom entry to be lost. The seventh stack function, Exchange, swaps the result line with the top stack entry. This operation is useful if numbers are entered in the wrong order and need to be corrected.

All the remaining functions are standard math operations which operate on the result line. Except for Power and Pi, these functions affect only the result line. The Power function raises the number on the top of the stack to the power specified in the result line; the answer is left on the result line and the stack shifts up one location. The Pi function enters the value of *pi* (3.1415926) in the result line just as if you had typed it from the keyboard.

Using an RPN calculator may seem strange at first, but once you become familiar with its powerful features, you may never want to go back to using a simple pocket calculator.

See program listings on page 126.

Gradebook

Stephen Levy and Kevin Mykytyn

A gradebook is an important recordkeeper for a teacher. An electronic gradebook is an efficient, convenient, and even more valuable organizational tool. This all machine language program—which includes some of the features of databases and spreadsheets—handles up to 70 students with 100 grades each. It also alphabetizes names, averages weighted grades, and much more. For the Commodore 64.

Schools continue to buy computers for students. But teachers need the time-saving convenience offered by computers as much as anyone. "Gradebook," designed especially for teachers, is a utility that efficiently handles classroom recordkeeping, replaces the periodic drudgery of averaging grades, and is easy to use.

Gradebook keeps a record of up to 100 grades for as many as 70 students in one file. In addition, the program will average grades, display them to the screen, sort (by name or grade), and print out the results.

A Great Big Work Sheet

When Gradebook is first run, you'll see a work sheet filled with meaningless characters. Press SHIFT-CLR/HOME to clear the work sheet. A gray screen with six columns filled with asterisks then appears. What you're seeing is a window on a much larger work sheet. The top row contains the assignment number, one per column. If you could view the entire work sheet, you'd see that it contains 101 columns, the first one for student names and the next 100 for individual assignments.

Press the cursor-left/right key so that the cursor is anywhere in the column for assignment 1. Now press the cursor-up/down key a few times. Notice how easy it is to move around the work sheet. If you move the cursor to the far right column and press cursor-left/right, all columns scroll to the left to make room for the next column. Pressing SHIFT-cursor-left/right moves the cursor left, and SHIFT-cursor-up/down moves it up.

The left-arrow key (at the upper left on your keyboard) functions like a "toggle," and enables you to move at fast or normal speeds around the work sheet. To move quickly, press the left-arrow key once. Now each time you press one of the cursor-movement keys, you'll move five rows or columns. To return to single-step movement, press the left-arrow key again.

Anytime you wish to move to the top left of the work sheet, press CLR/HOME.

Entering Names And Grades

Move the cursor to the upper-left corner of the work sheet by pressing CLR/HOME. The cursor now appears as a long white bar in the column for names. Each name must start on the left margin, so be sure not to type a space as the first character. There's no need to type names in alphabetical order because the program will do that for you (see below). Be sure to enter last names first if you wish to have students alphabetized by last names. After entering a student name, press RETURN or cursor-up/down to enter another. *Do not leave blank lines between names*—each name must immediately follow the one above.

Before you can enter grades for an assignment, you must first enter the perfect mark (a whole number between 0 and 200) and percentage (weight) of grade for the assignment. To begin entering individual grades, move to the cell (column and row coordinate) for the student and assignment number you want to enter. All grades must be whole numbers between 0 and the perfect mark you've determined for that assignment. To change a student name or grade, go to the appropriate location and type it in. It will automatically replace the old data. If you make an error while entering, use the INST/DEL key as you normally would to erase the previous character.

Calculating Grades

At the top of each column is displayed the assignment number and a place to enter the perfect mark and the percentage (weight) of the grade.

The program uses the perfect mark value for each assignment to calculate a percentage grade for each mark. This feature offers an added convenience: You don't have to figure all grades on a 100 percent base. You can enter a mark based, say, on the actual number of correct answers.

Percentage of grade allows you to give different weights to various tests, finals, midterms, and so on. For example, suppose you give seven tests during a quarter. Two of the seven are unit tests which count more heavily than the other five. All you need to do is assign a higher percentage for the unit tests. Note, however, that the total of all the percentages of grades must equal 100. In our example, we might have the two unit tests each count as 25 percent of the total and the other five tests each count as 10 percent (25, 25, 10, 10, 10, 10, 10, for a 100 percent total).

Once the perfect marks and

percentages of grades are entered, press the £ key to calculate grades. You'll get two types of calculations. *Ave* is the unweighted average (all marks are added together and divided by all the perfect marks added together.) *Fin* is the weighted average, which is calculated with this formula:

$$\frac{\text{mark\#1 * weight}}{\text{perfect mark}} + \frac{\text{mark\#2 * weight}}{\text{perfect mark}} + \frac{\text{mark\#3 * weight}}{\text{perfect mark}} + \text{and so on}$$

If the total percentage of grades doesn't add up to 100, you won't get a figure in the Fin column. Also, any student missing a grade will not have an average in the Fin column.

Press any key to return to the work sheet.

Deleting Grades And Names

Let's say you decide that all the grades for a single assignment should be eliminated. First, move the cursor anywhere on the column of grades to be deleted and press CTRL-D. When prompted, press C for column. Asterisks will fill the column. Now, you can either ignore them or reuse the column.

Likewise, if a student leaves your class and his or her grades are no longer needed, move the cursor to the row containing the student's name and press CTRL-D. But this time, when prompted, press R for row. Instantly, the row disappears and all rows below are moved up.

Deleting grades can serve another very useful purpose. At the end of each quarter or grading period, you can load in a work sheet from disk or tape (see below for loading instructions). Delete all grades, column by column, as discussed above. Then either enter new grades for the new quarter or save out just the names for later use. By saving the names using a different filename on disk or on another tape, you'll preserve the grades from the past quarter for future reference.

To delete a single grade, replace it with an asterisk. To delete all entries, both names and grades, press SHIFT–CLR/HOME. Use all delete commands carefully, as data recovery is impossible.

Gradebook Command Summary

CTRL-A	Sort row or column
CTRL-D	Delete row or column
CTRL-L	Load names and grades from tape or disk
CTRL-P	Print row or column
CTRL-S	Save names and grades to tape or disk
SHIFT-CLR/HOME	Delete all entries
£	Calculate averages
cursor-left/right	Move cursor right
SHIFT-cursorleft/right	Move cursor left
Cursorup/down	Move cursor down
SHIFT-cursorup/down	Move cursor up
CLR/HOME	Move cursor to upper left
←	Toggle cursor speed control

Saving And Loading

Once you've entered some names and grades and are ready to save the file, press CTRL-S. You'll be prompted for a filename (12 characters maximum) and asked to press T or D for tape or disk. If you use a filename that's already on your disk, it will be replaced by this new file.

Remember to save anytime you enter new names or grades. And it's best to use a different filename each time you save.

Loading is also easy: Just press CTRL-L and answer the prompts.

Printing And Sorting

Gradebook will print out any column to any Commodore printer (or any Commodore-emulating printer). To print a column, move the cursor to the desired column and press CTRL-P. *Be sure your printer is turned on before pressing CTRL-P. Otherwise, Gradebook will print to the screen, scrambling the format you've set up.*

If you ask for a column of grades, the name of each student will appear with the student's grade for that one assignment. If the cursor is in the column of student names and you ask for a printout of the column, you'll get a list of student names.

To get a printout of all student averages, press £ to go to calculations. Once the averages are displayed, press CTRL-P.

CTRL-A will sort any column. Move the cursor to the column with the student names, press CTRL-A, and the names will be sorted in alphabetical order. And the grades will move to the proper row. Place the cursor in a column of marks and press CTRL-A, and that list will be arranged in descending order. Again, all corresponding data will be rearranged.

The sorting function gives Gradebook flexibility. You could sort assignment 1 and then print out all the student grades for assignment 1. This would give you a list of names and grades, sorted by grades from highest to lowest. You can save multiple files with the same data, but sorted and arranged in different ways.

Gradebook has only a few commands, but, as with any program, it takes a little getting used to. The first few times you use the program, refer to the table of commands. Once you've used it a bit, you'll find that entering and averaging grades will be quick and easy.

Typing It In

Gradebook is written entirely in machine language, so you must use "MLX," the machine language entry program to type it in. Run MLX and enter the following information when prompted:

Starting address: C000
Ending address: CC27

Once Gradebook has been saved to disk or tape, load it with

LOAD *"filename"*,8,1 for disk

or

LOAD *"filename"* for tape

where *filename* is the name used to save Gradebook using MLX. Once the file is loaded, type NEW, press RETURN, and enter **SYS49152** to activate the program.

See program listing on page 74.

Sprite Designer

André Corbin

This full-featured sprite editor from a Canadian reader lets you design Commodore 64 sprite shapes quickly and conveniently. The program works with either disk or tape.

Nearly every Commodore 64 programmer uses sprites at one time or another. They're the movable graphics shapes that add so much to arcade games and other programs. If you have plenty of time, you can draw the sprites on graph paper, calculate the numbers that make up each sprite, type those numbers into a program, and hope that you didn't make any mistakes along the way. But "Sprite Designer" offers a much faster and easier alternative, allowing you to design sprites directly on the screen, with convenient keyboard controls. You can save a group of sprite shapes to disk or tape and reload them for future use, or you can create a series of BASIC DATA statements containing the sprite data. The program supports both single-color and multicolor sprites and it allows you to work on a group of eight sprites at a time.

Typing It In

Sprite Designer is written in machine language, so you must type it in with the "MLX" machine language entry program published elsewhere in this issue. Read the MLX instructions carefully before you begin. When you run MLX, you'll be asked for a starting address and an ending address for the data you'll be entering. Here are the values to use for Sprite Designer:

Starting address: 84D0
Ending address: 9E47

After you've saved a copy of the program, load it with LOAD "PROGRAM",8,1 for disk or LOAD "PROGRAM",1,1 for tape (replace PROGRAM with the name you used when saving the program). To start the program, type SYS 34000 and press RETURN.

Sprite Designer uses the memory in locations 8192–16383 to store sprite shape data. This work area can hold 16 groups of 8 sprites, allowing you to have a total of 128 sprite shapes in memory at one time.

Main Menu

When you start Sprite Designer, it displays a main menu containing the following choices:

1. INITIALIZE
2. EDIT
3. ANIMATE
4. SAVE
5. LOAD
6. BASIC DATA
7. MONOCHROME
8. MULTICOLORED
0. QUIT

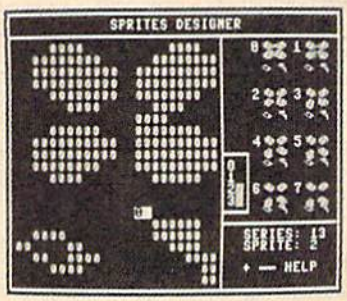
To choose a menu option, simply press the number key next to that option. For instance, the Edit option lets you design a sprite. To choose that option, you press the 2 key. Let's look at the menu options in order.

1. Initialize

Sprite Designer always begins by asking you if you want to initialize, or clear out, the entire sprite workspace. Since this action erases all sprite data in memory, the program asks you to confirm before it continues. If you press Y to answer yes, every dot in all of the sprite shapes is turned on. You can choose this option at any time from the main menu.

2. Edit

Option 2 (Edit) moves you to the sprite-editing screen, which is divided into three areas. In the lower right corner are two numbers that show the sprite series you are working on, as well as the individual sprite you are designing. Each



"Sprite Designer" is a comprehensive utility for designing sprite shapes for use in games and animation. This photo depicts the main editing screen.

series contains eight sprites, and the series are numbered 0–15, corresponding to their respective positions in memory. Within a series, each of the eight sprites is given a number 0–7.

In the upper-right portion of the screen you will see the shapes of all eight sprites in the current series. As you design a sprite, the changes immediately appear in this area.

The left portion of the screen is the editing area, where you actually design the sprite shape. This area is 24 × 21 spaces in size, with each screen space corresponding to a dot in the sprite shape.

Editing Functions

Once you have reached the editing screen, you need to remember only one command: The plus key (+) displays an entire screen of instructions for editing. To return from the help screen to the editing screen, press any key. The asterisk (*) returns you from the editing screen to the main menu.

When you enter the editing screen, the editing area contains the shape of sprite 0 within the current series. The function keys f3–f6 allow you to select any series and any individual sprite within a series. Press f3 to go to the next series,

or f4 (SHIFT-f3) to go back to the previous series. The f5 key advances to the next sprite in the series, while f6 moves to the previous sprite. If you have designed a series of shapes for animation, you can hold down f3 or f4 to move through the entire series quickly.

A reverse-video cursor indicates your position in the editing area. To move the cursor around, press the cursor keys just as you would normally. The size of the cursor depends on whether you are working on a single color or multicolor sprite. For a single-color sprite, the cursor is a square one space in size; for a multicolor sprite, it is a rectangle two spaces in size, containing a number in the range 0–3.

You can turn on every dot in the current sprite shape by pressing CLR. For a single-color sprite, the space bar lets you turn individual dots on and off inside the editing area. You can erase a dot by pressing the space bar and turn a dot on by pressing SHIFT–SPACE (hold down SHIFT and press the space bar).

For a multicolor sprite, a dot can have any of four different colors, including the background color (see "Multicolor," below). Use the numbers 0–3 or the letter keys Z, X, C, and V to color a dot with the desired color.

You also can change the position of the sprite shape within the editing frame, shifting everything one dot to the right, left, up, or down. This is done by holding down the Commodore key and pressing W, Q, R, or E, respectively.

It is possible to flip a sprite shape horizontally or vertically, by pressing Commodore-I or Commodore-K, respectively.

The f1 key allows you to copy a shape from one sprite to any other shape in memory. First, use function keys f3–f6 to put the shape you want to copy into the editing area. Press f1 to copy the shape; then use f3–f6 to select the shape you want to copy to. Press f1 a second time, and the shape is copied instantly.

The f2 key inverts every dot in the current sprite shape. Use f7 and f8 to change the sprite's color.

3. Animate

Option 3 (Animate) lets you cycle through all the sprite shapes in ascending order, beginning with series 0, sprite 0. When you choose this option, Sprite Designer prompts you to enter a delay interval which controls the speed of the animation. The smaller the delay interval, the faster the animation. Press SHIFT–SPACE to stop the animation and return to the main menu. You can also use keys f3–f6 to cycle forward or backward through the sprite shapes, as explained in the preceding section.

4. Save

Sprite Designer allows you to save an entire set of sprite shapes to disk or tape. Enter a filename when prompted; then press D or T to indicate disk or tape, respectively. To cancel the operation, press RETURN when prompted to choose disk or tape. If a disk error occurs, Sprite Designer prints the appropriate error message at the bottom of the screen.

5. Load

This option allows you to reload a set of sprite shapes from disk or tape. Follow the onscreen prompts as explained in the preceding section.

6. Add DATA Lines

Option 6 (BASIC DATA) lets you save sprite shapes in the form of BASIC DATA statements. This is done by adding DATA lines to an existing BASIC program. Activate Sprite Designer and load your existing sprite shapes or design new ones. Once the sprite shapes are in memory, exit Sprite Designer and type NEW. Then load the BASIC program to which you wish to add the sprites. Don't worry that the program will disturb the sprite shapes: Sprite Designer automatically copied them into a safe place (the RAM under BASIC ROM) when you exited the program.

When the BASIC program loads, do not run it. Instead, type SYS 34000 to reactivate Sprite Designer. Do *not* initialize the sprite shapes at this stage, or load any shapes from disk or tape. Instead, choose option 6. The program prompts you to enter the series of sprites that you wish to save. When that is done, you also are given the opportunity to include REMarks along with the DATA lines.

After you answer those questions, the program prompts you to enter a filename and indicate whether to use disk or tape, as in options 4 and 5. At this point, Sprite Designer adds the DATA lines to the program in memory and saves the new program. If you wish to cancel the operation, press RETURN at the filename prompt without typing anything.

After the save is complete, exit Sprite Designer and type NEW (to reset BASIC's internal pointers). In order to work properly, the BASIC program to which you add the DATA lines must not exceed 24 disk blocks (6144 bytes) in length.

7. Monochrome

Each series of eight sprites can be either monochrome (single color) or multicolor. You cannot mix monochrome and multicolor sprites in the same series. When you choose option 7, a 16-color board appears on the screen. To choose a color, press the letter key matching that color. That color will be used for all eight sprites in the designated series.

8. Multicolor

Option 8 allows you to choose colors for a series of multicolor sprites. When you choose this option, a 16-color board appears as described in the previous section. Since a multicolor sprite can have four colors, the program asks you to select four colors, numbered 0–3. When you're drawing in multicolor mode, the cursor contains a number to indicate which color you are drawing in. To switch to a different color, press a number in the range 0–3. A small color board to the right of the drawing area shows you the current drawing colors.

0. Quit

When you exit Sprite Designer, the program copies all the sprite shapes in the workspace into the "hidden" RAM underlying the BASIC ROM chip. This is done to make it easy for you to alternate between using your own BASIC programs and Sprite Designer. You can load and use a BASIC program after using Sprite Designer, then reload and activate Sprite Designer again. If you choose *not* to initialize the work area, Sprite Designer copies the previously stored sprite shapes back into its workspace. This eliminates the need to reload the sprites from disk or tape every time you switch between a BASIC program and Sprite Designer.

See program listing on page 112.

Memo Diary

Jim Butterfield, Associate Editor

Keep track of important dates, holidays, and personal events with this simple, easy to use BASIC program.

"Memo Diary" helps you record and recall birthdays, holidays, appointments, or any other event worth remembering. The program maintains a data file with as many as 100 events whose dates can range from tomorrow to one year in the future. You can record two different types of dates: temporary, one-time events such as appointments which have no importance once they have passed; and permanent, recurring events such as birthdays and anniversaries. By routinely running Memo Diary each time you use your computer, you'll no longer have to worry about forgetting to mail a birthday card to a relative or finding an anniversary gift for a spouse.

The program always shows the correct day of the week when you enter a date, and you need to enter the year only once—the very first time you run the program. After that (for the next 99 years, anyway) Memo Diary keeps track of the year for you. Each time you run the program, it automatically shows all due and overdue events on the screen or printer, and erases one-time events from the calendar after they're displayed.

You can enter temporary or recurring new events and erase existing events whenever you wish. You can also examine all events from the current date forward, or search the entire calendar for events matching a given starting pattern. Finally, Memo Diary saves your calendar either on disk or tape.

Typing The Program

Type in the program and be sure to save a copy and read the instructions before running it.

The first time you run Memo Diary is special. *Do not start the program by entering RUN.* You should type RUN 100 and press RETURN. *If you don't do this, the program will not work correctly.* When you start the program at line 100, Memo Diary lets you enter the correct year without looking for a previous file of events. Thereafter, start the program with RUN in the usual way.

On the first run you'll probably want to enter fixed holidays such as New Year's Day as well as birthdays and anniversaries. These are permanent events that you won't need to enter year after year. A holiday like Thanksgiving should be entered as a one-time event since it falls on a different date each year.

When Memo Diary asks you to enter today's date, you can type in the name of the month (such as OCTOBER) or its number (such as 10). In either case, be careful to enter it correctly. Memo Diary lets you enter any day of the month from 1 to 31, so it won't mind if you specify the date as February 30. Mistakes like these may confuse the calendar file. For instance, if you use the program on July 4 and the next day mistakenly give the date as June 5, the computer thinks you've let almost a whole year go by. To warn you of this, Memo Diary displays HAPPY NEW YEAR. If you see this message when a new year hasn't arrived, stop the program and start over, entering the correct date.

A Memory Jogger

Except for the very first run, Memo Diary always begins by reporting all due and overdue events ("You just missed your anniversary"). Take careful note of these events, since they'll soon be erased from the calendar (if they're temporary events) or moved ahead to next year (if they're permanent). To help jog your memory, Memo Diary also lets you make a copy of the list of events on your printer.

After disposing of due and overdue events, Memo Diary displays five options: You can see future events, add a new event, cancel an event, search for an event, or quit the program. You'll ordinarily want to look ahead to see what's coming in the next week or two. To do this, choose Option 1 (see future events) and supply an appropriate future date when requested. If you enter the current date when looking at future events, Memo Diary assumes you mean the same date *next*

year and gives you everything on file.

When you want to make a new entry, select Option 2 (add new event). First Memo Diary asks whether the new event is one-time or permanent. Then it lets you enter the date and details. Again, the current date is understood as one year from today (it's assumed you don't need to record an event that's happening the same day).

To cancel an event (Option 3), you must know its date. When an event is entered, you're shown every item scheduled for that date, each with its own code number. To cancel an event, type in its code number when prompted.

Option 4 (search for event) lets you search for an event based on the first few letters of the entry. You may find many events in the course of a search. For instance, if the calendar file contains the events CLUB MEETING, CLUB CONFERENCE, and CLUB ELECTION, searching for CLUB displays all three events. In this case you would *not* see the entry CANADIAN CLUB, since CLUB is spotted only if it's in the first word of the entry. Thus, if you plan to search for certain keywords (BIRTHDAY, CHURCH, SOFTBALL, or whatever) keep them at the front of each calendar entry.

After you've finished an option, Memo Diary always returns you to the main menu. Sooner or later you'll be ready to use Option 5 (quit). The program knows when it's time to update the calendar file. If you've erased past and overdue events, added or deleted items, Memo Diary will—with your permission—proceed to update the data file on disk or tape.

The Time Pivot

A program that handles dates can encounter some subtle paradoxes. Does August come before April, or after it? The correct answer is *both*. Memo Diary could resolve this difficulty by adding a year designation to every event, but that complicates the handling of permanent events, which don't belong to a specific year. This is not a trivial problem: If you schedule a new event for August, the program must decide whether to add the event to the calendar ahead of an existing April event, or after it. Without a year designation, how can anyone tell?

The problem is solved by using a *pivot* date, usually the same as the current date. If today is July 4, August does indeed come before April. On the other hand, if today is November 11, April comes before August. Since the calendar always looks one year into the future, everything is kept in order.

However, there's one case in which the pivot date can't be the current date. Each time the program begins, it must measure the time lapse since its last use. For example, say that you last used the program on August 20, 1985 and next use it on September 4, 1985. On the first run (August 20) Memo Diary uses August 20 as the pivot. That way an event dated September 1 is seen ahead of another item dated in October.

On the second run (September 4) the September 1 event is reported as past due and either erased from the calendar (if it's temporary) or moved ahead to September 1 of next year (if it's permanent). Once this is done, the pivot date moves forward to September 4, meaning that a September 1 event now belongs *after* an item dated in October. Don't worry if this sounds confusing: It works out more simply in practice than in theory.

The day of the week is worked out with a simple formula. If you haven't seen it before, here's a hint on how it works. The calendar is modified to make March 1 the first day of the "adjusted year." This way, leap year with its extra February 29 date doesn't break up the sequence of days: The extra leap day just gets pasted onto the year's end. Though the math is a bit convoluted, you may find it interesting to trace the logic of this routine (it starts at line 2150).

Expanding The Calendar

Memo Diary can keep track of a maximum of 100 events. In practice it's wise to limit the number to 80 or 90 to leave room for permanent events that move automatically from the front to the back of the list. If you need more than 100 events, change the L$ value in the DIM statement. Line 150 contains the value L$(100). You can increase the 100 to whatever number you like, but don't get carried away. Since Memo Diary uses string arrays, a very large value may cause garbage collection delays. There's no particular limit to the number of events allowed for a particular date.

Program Notes

Let's take a look at the program's major features. Line 90 prepares Memo Diary to read a file. The variable F is a *Boolean* (logical) variable that's defined as *true* here, to let you read the calendar file on a normal run. When you enter at line 100 on the first run, F is *false* (like every other undefined variable) and no file is read.

DATA statements in lines 110–140 hold the names of the months of the year and days of the week; the names are read into the arrays M$ and W$. Line 150 dimensions the L$ array for 100 items. Lines 230–250 call for a reading of the calendar file if appropriate. This is done in the subroutine at line 3010. When Memo Diary reads this file, it detects and reports the last date the file was used. Line 260 asks for today's date; the subroutine at line 1670 asks for and accepts the date.

Now it's time to search for due and overdue events. Using the previous date as a pivot, the subroutine at line 1960 scans for all events up to today's date. The program reports these events, erases them, or moves them ahead as needed, and proceeds to the main menu. Line 680 begins a main activity loop: It prompts with the menu, asks for a choice, then goes to the appropriate subroutine. Line 850 lets you see future events. Since the pivot date is now today, the program scans to the requested future date to see how many events fall into the today-to-future-date range.

Line 940 lets you add a new event. After asking ANNUAL OR ONE-TIME? the program requests the event's date and then asks for details. After adding a year designation to the date of one-time events, the new event is inserted into the proper sequence. Line 1210 lets you cancel an event. Memo Diary asks for a date and then lists all events that match that date. At line 1350, the program asks which event to delete. Note that the number you supply must be in the correct range.

Line 1450 begins the search-for-an-event routine. After it receives a search string (P$), the program looks for a match. When it scans through the calendar, it must look in different places depending on whether the event is one-time or permanent. That's because one-time events carry a year designation, making their dates three characters longer.

A Horrible Mistake?

Line 1570 handles the quit option; the flag F9 registers activity. If you haven't changed any of the data, there's no need to update the calendar file. Before scratching the old file and writing the new one, the program asks whether you're ready. That way, if you made some horrible mistake, you can cancel the file update.

The main loop ends at line 1580 and is followed by several subroutines. The routine starting at line 1590 writes a new calendar file when appropriate, and line 1670 begins the date input routine. The date is formed into a string (D8$) to allow for easy searches or entry. The subroutine at line 1930 reads the calendar file. The first item in the file is always the most recent date of use; the remaining data is events.

The subroutine at line 1960 scans all events to see which have dates between the pivot date (D9$) and a second date (D8$). There are three dates involved: event, pivot, and the second date, which makes the comparison a bit messy. Boolean variables keep everything in order. Eventually, the variable F0 indicates the date is in range, and the variable L0 indicates when the last event is found within the date range.

The routine starting at line 4020 displays the information, on the printer if desired.) The date is given complete with the day of the week, and events falling on the same day are grouped together. The weekday calculation begins at line 2150. The weekday variable, W, ranges from 0 to 6, so 0 means Sunday. As written, this routine is good for years ranging from 85 (1985) to 84 (2084). If you want to plan more than 99 years in advance, you'll need to modify the routine.

See program listing on page 77.

The Construction Set

Fred Karg

This unique program has the fun elements of a game, but it's not a game. A creative toy? Maybe. If that sounds too vague, it could also be called a tool for computer-aided design. Whatever you call it, you'll find it's a lot of fun for children—and adults. A joystick is required.

All our lives we build things, often just for the fun of it. Toddlers play with building blocks. Grade schoolers create elaborate structures with Tinker Toys, Lego blocks, and Erector sets. Teenagers put together model cars and planes. Adults have basement workshops for making things out of wood. Even the process of writing a computer program can be described as building something from parts.

"The Construction Set" is unlike most other computer programs. It can hold up to four different sets of building blocks which can be combined easily in any way you like. No one wins or loses and there's no high score. But it's great fun to create your own interesting picture.

Special Typing Instructions

The program listings include a BASIC program and three programs in MLX format. Type in Program 1 (The Construction Set Main Program) first. Datassette owners should change **DI=8** to **DI=1** in line 160.

The three MLX listings (Programs 2–4) are not machine language programs; they're character sets for three different sets of building blocks. Follow these directions to create each of the character sets:

1. Load the MLX entry program (found elsewhere in this issue). If you haven't used it before, be sure to read the directions carefully.

2. Type RUN and answer the prompts:

```
Starting Address: 7000
Ending Address:   7803
```

3. Type in Program 2, 3, or 4. When you've finished, save the program.

4. Repeat these steps for each of the character sets you wish to use.

You don't have to type in all of the character sets, but you need at least one for The Construction Set to work properly.

Line 170 of the main program keeps track of how many character sets will be loaded and what their

names are. Currently, it's set up for 3. If you enter only one or two sets of building blocks, change the 3 to a 2 or 1 and follow it with a comma and the name you used to save the character set file.

Tape users should be especially careful with line 170. Save all of the character sets to the same tape, one after another. Remember which one was saved first and make sure that filename is the first in line 170. For example, if you typed in and saved the first two sets, and saved them as HOUSE and MONSTERS, line 170 of the main program should read:

```
170 DATA 2,"HOUSE","MONSTERS"
```

Remember to modify line 160 as well.

Putting Together A Picture

When you first load and run The Construction Set, you'll see the menu of commands on screen while the program loads the character sets.

Any time you want to check the menu, press f7 and the list of commands will appear (press f7 to go back to the work area).

The work screen is divided into two parts. At the bottom, you should see several building blocks. A pointing hand symbol floats above the blocks. Use the cursor keys (left and right) to move the hand back and forth to select the block you wish to place on the screen. Press the plus and minus keys to see more building blocks in the same set. Some blocks are full size, others are shorter.

The top part of the screen is the work space. At first the screen is blank, except for a window (a hollow rectangle). With a joystick in port 2, move the window around the screen. Then press the joystick fire button to place a building block on the screen in the same position as the rectangle. To erase it, tap the INST/DEL key. Whichever block the hand is pointing to at the bottom of the screen is the one that's stamped on the screen.

The process of creating a design is fairly easy. First move to the set of blocks you want with the plus and minus keys, then use the cursor keys to move the hand to the pattern you want. After selecting a block, position the window and press the fire button. Patterns can be repeated as often as you like.

If you forget the commands, press f7 to see the menu. If you want to start over, clear the screen by pressing the shifted CLR/HOME key.

When you're finished, you can experiment with different colors for different parts of the screen. Press the left-arrow key (right above CTRL) and then one of the following keys.

(←) f1: Change border color.
(←) f3: Change screen color.
(←) f5: Change character colors.

Four Character Sets, Four Screens

The Construction Set has room for four different character sets. Each defines a set of building blocks. The three accompanying the program are "House" for making brick buildings, "Creature" for putting together people (and monsters), and "Shapes" for abstract patterns. These custom character sets were created with "Ultrafont +" from the July 1984 GAZETTE. You can create your own Construction Set building blocks if you have a copy of Ultrafont +. You could invent building blocks for electronic circuits, quilts, landscapes, flags, or a variety of other patterns.

Each set of building blocks has its own screen. You can travel between the workspaces of each of the character sets by pressing f1 (as mentioned above, if you press the left-arrow key first, f1 changes the border color). The pictures you've created are preserved when you switch back and forth. So if you press f1 four times, you'll come back to the screen where you started and the picture will still be there.

An interesting modification for children is to load the same set of building blocks into all four workspaces. For example, you could change line 170 to **170 DATA 4, "CREATURE","CREATURE","CREATURE","CREATURE"** (tape users will have to use MLX to save the Creature font four times in a row on a single tape). Then have the child make up a story and build four pictures to illustrate the plot. When the characters have been put together, use f1 to step through the pictures as the child tells the story.

See program listings on page 117.

Only NRI teaches you to service all computers as you build your own fully IBM-compatible microcomputer

NEW! Train with the newest Sanyo 880 Series Computer—it's fully IBM-compatible and runs almost twice as fast!

With computers firmly established in offices—and more and more new applications being developed for every facet of business—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 rate than for any other occupation.

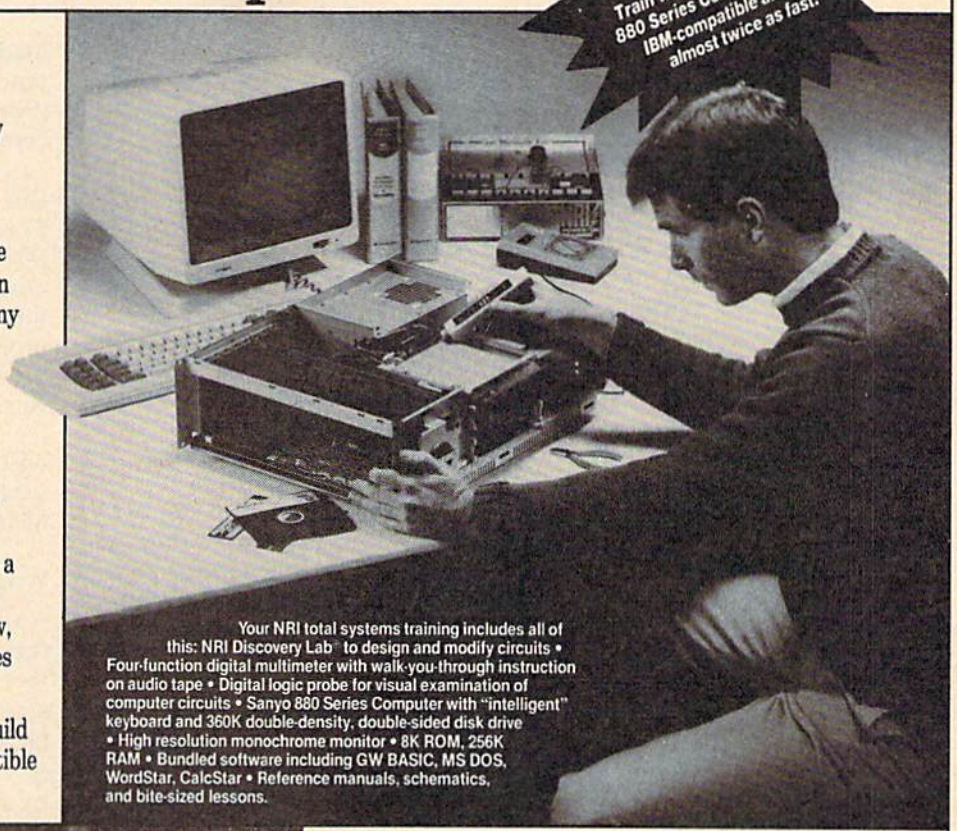
Total systems training

No computer stands alone... it's part of a total system. And if you want to learn to service and repair computers, you have to understand computer *systems.* Only NRI includes a powerful computer system as part of your training, centered around the new, fully IBM-compatible Sanyo 880 Series computer.

As part of your training, you'll build this highly rated, 16-bit, IBM-compatible computer system. You'll assemble Sanyo's "intelligent" keyboard, install the power supply and disk drive and interface the high-resolution monitor. The 880 Computer has two operating speeds: standard IBM speed of 4.77 MHz and a remarkable turbo speed of 8 MHz. It's confidence-building, real-world experience that includes training in programming, circuit design and peripheral maintenance.

No experience necessary—NRI builds it in

Even if you've never had any previous training in electronics, you can succeed with NRI training. You'll start with the basics, then rapidly build on them to master such concepts as digital logic, microprocessor design, and computer memory. You'll build and test advanced electronic circuits using the exclusive NRI Discovery Lab®, professional digital multimeter, and logic probe. Like your computer, they're all yours to keep as part of your training. You even get some of the most popular software, including WordStar, CalcStar, GW Basic and MS DOS.



Your NRI total systems training includes all of this: NRI Discovery Lab® to design and modify circuits • Four-function digital multimeter with walk-you-through instruction on audio tape • Digital logic probe for visual examination of computer circuits • Sanyo 880 Series Computer with "intelligent" keyboard and 360K double-density, double-sided disk drive • High resolution monochrome monitor • 8K ROM, 256K RAM • Bundled software including GW BASIC, MS DOS, WordStar, CalcStar • Reference manuals, schematics, and bite-sized lessons.

NRI is the only technical school that trains you on a total computer system. You'll install and check keyboard, power supply, disk drive, and monitor, following step-by-step directions.

Send for 100-page free catalog

Send the coupon today for NRI's 100-page, full-color catalog, with all the facts about at-home computer training. Read detailed descriptions of each lesson, each experiment you perform. See each piece of hands-on equipment you'll work with and keep. And check out NRI training in other high-tech fields such as Robotics, Data Communications, TV/Audio/ Video Servicing, and more.

SEND COUPON TODAY FOR FREE NRI CATALOG!

NRI SCHOOLS
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, NW, Washington, DC 20016
We'll give you tomorrow.

For Career courses approved under GI Bill ☐ check for details.

☑ CHECK ONE FREE CATALOG ONLY

- ☐ Computer Electronics
- ☐ TV/Audio/Video Servicing
- ☐ Satellite Electronics
- ☐ Robotics & Industrial Control
- ☐ Data Communications
- ☐ Industrial Electronics
- ☐ Communication Electronics
- ☐ Electronic Design Technology
- ☐ Telephone Servicing
- ☐ Digital Electronics Servicing
- ☐ Basic Electronics
- ☐ Electricians
- ☐ Appliance Servicing
- ☐ Small Engine Repair
- ☐ Air Conditioning, Heating, & Refrigeration
- ☐ Locksmithing & Electronic Security
- ☐ Building Construction
- ☐ Automotive Servicing
- ☐ Photography
- ☐ Bookkeeping & Accounting

Name (Please print) Age

Street

City/State/Zip

Accredited by the National Home Study Council 231-028

128 DOS Wedge

Shawn K. Smith

This utility makes it easier for 128 users to communicate with their disk drive—the 1541 or 1571. Similar to DOS wedges for the VIC and 64, its commands should be familiar to many Commodore owners.

Commodore disk drives are intelligent peripherals. They have their own RAM, ROM, and CPU. The computer gives commands to the disk drive, and the drive decides how to carry them out.

Unfortunately, the interface between the user and disk drive isn't always friendly. To aid owners of the VIC and 64 (and the 128 in 64 mode), Commodore supplies the VIC-20 Wedge and the C-64 Wedge on the *Test/Demo* disk that comes with the 1541 disk drive. Unfortunately, no wedge has been made available for the Commodore 128. Of course, BASIC 7.0 has new disk commands, but these are longer than the DOS wedge equivalents and are unfamiliar to people who have previously used the wedge on the 64 or VIC.

Simple And Efficient

To demonstrate the efficiency of the DOS wedge, let's look at a quick example. Assume that the light on your disk drive is blinking, and you want to read the error channel. Without the wedge, you must type this (in BASIC 2.0):

```
10 OPEN1,8,15:INPUT#1,E,D$,T$,
   S$:CLOSE1:PRINT E;D$;T$;S$
```

With the wedge activated, simply type an at sign (@) and press RETURN. The equivalent BASIC 7.0 command is PRINT DS$. This is preferable to the BASIC 2.0 version, but it requires seven more keystrokes than the wedge.

"128 DOS Wedge" works only in direct mode. If you need to access the drive from within a program, you should use the special commands BASIC 7.0 offers. In direct mode, however, one keystroke is all you need to perform a variety of useful disk functions.

Typing It In

Because 128 DOS Wedge is written entirely in machine language, it must be entered with "MLX," the machine language entry program found elsewhere in this issue. Be sure to read and understand the instructions for using MLX before you begin entering the data for 128 DOS Wedge. When you run MLX, you'll be asked for a starting address and an ending address for the data you'll be entering. The correct values for 128 DOS wedge are

Starting address: 1A00
Ending address: 1AEF

After you finish typing in the data, be sure to save a copy before you leave MLX. When you're ready to use the wedge program, you can load and execute it using

BLOAD"*filename*":SYS 6656

or

BOOT "*filename*"

Substitute the filename you used when you saved the 128 DOS Wedge data for *filename* in the commands above.

With the wedge in memory, you can save a copy of it to another disk with this statement:

BSAVE"*filename*",P6656 TO P6896

Now, with 128 DOS Wedge loaded and activated, you have a powerful disk management system. (See the table for a list of available commands and explanations.)

128 DOS Wedge is stored in the free RAM area at locations $1A00–1AED, so it doesn't interfere with BASIC. However, any other ML program or data stored in this area will cause conflicts. The program also uses the RS-232 input buffer (at $0C00) for analyzing command strings.

If you need to disable the wedge, use the ! command. The wedge can be reactivated with SYS 6656.

Following is a list of 128 DOS Wedge default settings.

- All loads, saves, verifies, and directory listings are directed to device 8 (U8), drive 0 (D0).
- All disk commands (@ or >) default to device 8. The drive number (if applicable) should be supplied by you.
- The % command loads a file beginning at the address from which it was saved. If you want to load a binary file into a bank other than zero, use BASIC's BLOAD command instead.

See program listing on page 121.

128 DOS Wedge Commands

- **Drive Status: @ or >** Use this command when the drive light is blinking.
- **Send Command: @*command* or >*command*** This sends a command to the disk drive.
  Examples: @S0:*filename* (scratches a file)
  @S0:*filename1,filename2,filename3*
  @R0:*newname=oldname* (renames a file)
- **Directory: * or $** This command lists the directory. Wildcards are allowed.
  Examples: * (lists entire directory)
  $*=S (lists all sequential files)
- **Load BASIC program: / *filename*** loads a BASIC program into memory.
- **Load and run BASIC program: ↑ *filename*** loads and runs a BASIC program.
- **Load a binary file: % *filename*** loads a binary file into memory.
- **Save file: ← *filename*** saves a file to disk.
- **Verify file: ; *filename*** verifies a program.
- **Disable wedge: !** turns the wedge off.
- **Reactivate wedge: SYS 6656** turns the wedge back on.

128 Instant Keywords

Shawn K. Smith

Save time and typing effort with this short utility for the Commodore 128. Up to 52 keywords can be entered, each an easy-to-remember, two-key combination.

"Instant Keywords" can drastically reduce the time it takes to type in a program. This utility prints a BASIC 7.0 keyword when the Commodore or SHIFT key is pressed in conjunction with a letter key. For instance, pressing the SHIFT and L keys displays the keyword LOOP. A total of 52 keywords can be displayed in this fashion. Refer to the chart for a list of the key combinations. Also, pressing the SHIFT or Commodore key while in quote mode displays the standard graphics characters rather than a BASIC keyword.

Instant Keywords is short and easy to use. Although it contains mostly machine language (ML), you don't have to know any ML to use it. In fact, you can just type it in and run it as a BASIC program. First, type in the program and then save a copy. When you run it, the BASIC loader stores the ML in an area of RAM which is determined by the value S in line 100 (changing the value of S will relocate the utility). Once the data is stored in RAM, the utility is activated, the address to deactivate/reactivate it is displayed, and the loader is erased from memory. Pressing RUN/STOP and RESTORE is another way to deactivate the program.

Modifying The Program

Readers may wish to rearrange the utility to support a different set of keywords. This can be accomplished with minor changes to the utility. But first, a quick background about keywords is in order. BASIC 7.0 contains 130 plus commands or keywords. Most of the keywords (including all of the keywords in the 64's BASIC 2.0) are represented by one-byte tokens. For instance, the command PRINT is stored in the computer with a token value of $99 (153 decimal). Because the 128 has a larger vocabulary, the designers of the 128 decided to use two-byte tokens to represent some of the new commands. All of the new two-byte commands use $CE or $FE as the first byte of the token. Instant Keywords will allow you to use any keyword except those that begin with $CE as the first token value. (This eliminates the use of only eight keywords.)

The last 52 hexadecimal values in the loader (beginning with 0B in line 200) are the token values of the keywords displayed by Instant Keywords. The first 26 hex values are for the SHIFT key (the token for SHIFT-A is the first, and the token for SHIFT-Z is the twenty-sixth). The last 26 values are for the Commodore key. If you plan to add tokens for any two-byte commands, leave off the first byte ($FE)—the program knows that it is a two-byte command and will adjust itself accordingly. If you're unsure of the token value of any keyword, type in this program:

```
10 GOTO30
20 REM **PLACE KEYWORD HERE**
30 ={B}15:B=PEEK(45)+12+PEEK(4
   6)*256:PRINT"KEYWORD VALUE(
   S) = ";
40 H=PEEK(B):PRINTRIGHT$(DIM(H
   ),2);" ";
50 IFH=254ORH=206THENB=B+1:GOT
   040
```

In line 20, type the keyword for which you want to find the token value. Run the program and it will display the token value for the keyword you've inserted.

Letter	SHIFT	Commodore
A	SLEEP	STR$
B	BEGIN	BEND
C	CHR$	COLOR
D	DOPEN	DCLOSE
E	ELSE	ENVELOPE
F	FOR	FILTER
G	GOTO	GOSUB
H	HEX$	DEC
I	INPUT	INSTR
J	JOY	PLAY
K	DRAW	CHAR
L	LOOP	LOCATE
M	MID$	MOVSPR
N	NEXT	COLLISION
O	TAB(	SPC(
P	PRINT	PAINT
Q	GSHAPE	SSHAPE
R	RETURN	RESTORE
S	SPRITE	SOUND
T	THEN	TEMPO
U	USING	UNTIL
V	READ	DATA
W	WHILE	WINDOW
X	POKE	PEEK
Y	GRAPHIC	CIRCLE
Z	LEFT$	RIGHT$

See program listing on page 126.

64 Mode Speed-Up
For The 128

Gary Lamon

Once you get used to the 128's fast mode, 64 mode seems especially slow. This short program offers a way to significantly speed things up. For the 128 in 64 mode only; the program doesn't work on a standard 64.

The more you use a computer, the more you wonder if it couldn't be just a bit faster—especially when it's in the middle of a time-consuming task like alphabetizing a list of 800 names. If you own a Commodore 128, you can use the FAST command to double the speed of programs running in 80 columns. Although it also works in 40 columns, the screen goes blank. When you type GO 64, you give up access to the FAST command, but you don't have to give up fast mode. There are several interesting ways to squeeze more speed out of the Commodore 128's 64 mode. First, let's look at some background information.

Every computer has an internal clock which paces the processor. The faster the clock's speed, the more instructions the computer can execute in a given time. A Commodore 64 contains a 6510 microprocessor with a clock speed of about 1 megahertz (MHz), one million cycles per second. On the other hand, the Commodore 128 uses an 8502 microprocessor that's compatible with the 6510 but can run at a speed of either 1 or 2 MHz. When you're using the 64 mode on your 128, the system automatically sets the speed of the 8502 so that the machine performs exactly like a Commodore 64.

It seems a waste that 128 users cannot make use of this additional speed when running their old 64 programs in 64 mode. But there is a way. We *can* double the computer's speed in 64 mode with a few simple POKEs:

POKE 53296,1 (double speed)
POKE 53296,0 (normal speed)
POKE 53296,3 (double speed and screen off)

If you try the first or third of these POKEs in 64 mode, you'll indeed find that your programs run at twice normal speed; but there's a problem. The screen fills with a flashing checkerboard pattern (if you use the first POKE) or goes completely blank (if you use the second). The regular screen is still there, but it cannot be read. The problem is that the 40-column video chip (the VIC II) cannot keep up with the 8502 when the 8502 is running at 2 MHz. The third POKE works well on a 64 program that does, say, a great deal of number crunching. With this kind of program, it's probably not important to have video for part of the program's execution.

There's another way of achieving a significant speed increase while retaining an almost normal picture. The program accompanying this article—"64 Mode Speed Up"—provides approximately a 20 percent speed increase and leaves the screen readable. After typing in the program, save a copy. While in 64 mode, load and run the program and then type NEW. Your machine is now 20 percent faster. To return to normal speed, type SYS 49236 or press RUN/STOP–RESTORE. To get back to fast speed, type SYS 49152. If you'd like to check this, write a short BASIC program with a large loop (such as: 10 FOR I = 1 TO 30000:NEXT) and time it to measure the speed increase. (*Note: You should return to regular speed before all disk or tape operations).*

How It Works

You may have noticed a flash at the top of the screen while at fast speed. This is normal. But what causes this flash and how does the program work?

The program works by using a machine language *raster interrupt* routine in locations 49152–49258. The raster can be thought of as a sort of paintbrush that paints the picture on the video screen. The raster paints one line at a time across the screen starting at the top left and then moves down one line at a time. The program takes advan-

tage of the fact that we can see only raster lines 51 to 251. The computer is interrupted when the raster is at line 251 (the last visible line) and told to speed up to 2 MHz. This speed is maintained until the raster reaches line 51 (the first line we can see) and then reduced to 1 MHz. While the screen is "painted," the computer is running at regular speed. The result is a computer that runs faster, and you don't have to sacrifice the screen.

But what causes the flash at the top of the screen? Occasionally the computer is performing a task and does not want to be interrupted quite yet, so a few raster lines are done at the 2 MHz speed. (Remember what happened when you typed POKE 53296,1?)

Two memory locations within the interrupt program can be used to speed up the system even more:

POKE 49257, top raster line
POKE 49258, bottom raster line

As an example, try this with the fast mode operating (after SYS 49152): **POKE 49257,150**. You'll find that the flashing garbage will expand to fill the upper half of the screen, but the lower half will remain normal. The computer will now run about 1.6 times faster than a normal 64. You can expand or contract the screen any way you like with the two POKEs above. The more "garbage" visible, the faster the computer. One good way to visibly check the speed of the computer is to load a BASIC program and LIST it at the fast speed and at regular speed. The listing will scroll by considerably faster with the interrupt operating.

See program listing on page 122.

Mozart Magic

James Bagley

Based on a musical game devised by the composer Mozart, this delightful program for the Commodore 128 composes its own minuets in the style of Mozart himself.

This Commodore 128 program is a translation of a game by Wolfgang Amadeus Mozart. It composes a complete, original minuet at random. Mozart delighted in games of chance, so it was only natural that he should combine his two interests and produce an activity known as *Musikalisches Wuerfelspiel*, or musical craps. The idea was not original with Mozart, but his effort was the most successful.

Making Music

Type in and save the program; then run it. After it plays an introduction and initializes, the program displays a menu. You can choose a different instrument for each voice, but most songs sound best if you choose the same instrument for all three voices. Some of the instruments such as the drum and xylophone may sound strange or faint; they are included for the sake of completeness, so you can hear what all the 128's instruments sound like.

The next menu allows you to change the tempo. Press F to increase the speed at which the minuet is played, press S to decrease the speed, and press E to exit the routine. The tempo always defaults to 8. The main menu reappears after the minuet is finished.

The program itself is structured to reflect the composer's original technique. Mozart set up two grids of 8 columns and 11 rows. The columns were numbered 1–8, and the rows were numbered 2–12. On the first throw of the dice, he scanned down the first column to the row numbered the same as the sum of the two die. At this intersection was a number. He then copied down a measure of music corresponding to this number and repeated the process until he reached the eighth column of the first part.

In the eighth column of the grid, each number referred to a measure of music with two sets of notes. Because the music modulated to the dominant, the lower notes served for the first ending and the upper notes were for the second ending. Since these measures were all the same, M2$(1) is used in the program for the first ending and M2$(2) for the second ending of the first part of the minuet.

See program listing on page 115.

128 File Viewer

Jeffrey D. Partch

This Commodore 128 utility packs a double punch. It can print the contents of any disk file or disassemble any machine language program directly from disk. Neither operation disturbs the program currently in memory. A disk drive is required.

Have you ever wanted to know the contents of a mysterious file in the disk directory, or needed to look at another file during a programming session? All too often, getting that information requires a lot of saving, loading, and listing. "128 File Viewer" allows you to display the contents of any program (PRG) or sequential (SEQ) disk file on the screen without harming the program in memory. It can also disassemble any machine language program directly from disk—again, without disturbing the current program. Since it adds a new command to the Commodore 128's BASIC, this program is very easy to use.

Because File Viewer is written entirely in machine language, it must be typed in using the "MLX" machine language entry program found elsewhere in this issue. Be sure to read and understand the instructions for using MLX before you begin entering the data for File Viewer. When you run MLX, you'll be asked for a starting address and an ending address for the data you'll be entering. Here are the addresses for File Viewer:

Starting address: 1350
Ending address: 17E7

When you finish entering the data, be sure to save a copy to disk before you leave MLX. Once you have a completed copy of File Viewer on disk, you can activate it with a command of the form:

BOOT "FILE VIEWER",D*drive*,U*device*

Of course, you should replace FILE VIEWER in this statement with the filename you used when saving the data with MLX. If your disk drive is device 8, the normal device number for Commodore drives, you can omit everything after the closing quotation mark (just use BOOT "FILE VIEWER"). The first optional parameter is used to specify the drive number for dual-drive systems with drive 1 in addition to drive 0. (Commodore 1541 and 1571 drives are always drive 0.) The second parameter is used to specify a device number other than 8. For instance, BOOT "FILE VIEWER",D0,U9 boots the program from a disk in a drive addressed as device 9.

When the familiar READY prompt reappears, File Viewer has been installed and is ready to use. (Note that pressing the reset switch deactivates File Viewer.) This program works as an extension of the Commodore 128's BASIC, so it's as easy to use as any other BASIC statement. Here is the general syntax:

VIEW *"filename,type"*,D*drive*,U*device*

Again, the last two parameters are not needed if you are using a single 1541 or 1571 disk drive addressed as device 8.

The *type* parameter is also optional in most cases, since File Viewer ordinarily determines for itself whether the file is a sequential or program file. If necessary, however, you can specify the type by including an *S* for sequential files or a *P* for program files. For instance, the statement VIEW "SAMPLE,S" displays the contents of the sequential file SAMPLE, while VIEW "SAMPLE,P" displays the contents of a program file of the same name.

File Viewer ordinarily reads and displays the entire file. You can slow the display by pressing the Commodore key or pause it completely by pressing NO SCROLL. Press STOP if you wish to terminate the display before you reach the end of the file.

Disassembly

To aid machine language programmers, File Viewer also includes a disassembly option. Here's the syntax to use:

VIEW *"filename,M"*

The *M* stands for machine language. It's not a Commodore file type, but simply a signal to File Viewer that you wish to disassemble the file rather than print it to the screen. When you choose this option, File Viewer reads the file from

disk and disassembles it to the screen in standard 6502 assembler format. Just as with the display option, you can slow the disassembly with the Commodore key, pause it with NO SCROLL, or cut it off by pressing STOP.

Redirecting Output

In most cases you'll want to look at a file on the screen. However, File Viewer also lets you divert the normal screen output to a printer or other peripheral device. This option is most useful for machine language disassemblies, since it allows you to create a hardcopy printout of the program which can be studied at leisure. However, you can also use it as a quick way to print a text file without loading it into memory.

Diverting output requires that you open a logical file to the desired device. For instance, say that you want to send the contents of a file to the printer. The statement OPEN 1,4 opens logical file 1 to the printer (which is usually device 4). Once the logical file is open, you must tell File Viewer where to send its output. This is done by adding the logical file number to the VIEW statement:

VIEW# *file number,"filename,type",*
Ddrive, Udevice

The logical file number must match the one you used when you opened the logical file, and must be in the range 1–127. As with normal format for the statement, the type, drive number, and device number parameters are optional. For example, if you open a file to the printer with OPEN 1,4, this statement makes a hardcopy printout of the file SAMPLE from a disk in the drive addressed as device 8:

```
VIEW#1, "SAMPLE"
```

In special cases, you may wish to divert output to a disk file, or even to a modem via the RS-232 interface. File Viewer isn't picky about where it sends output, as long as you have properly opened a logical file to the device. The *Commodore 128 System Guide* explains the syntax needed to open a file to disk or RS-232 interface.

Special Concerns

The VIEW statement works only in direct mode; you should not attempt to add it to a program. VIEW does not accept string or numeric variables in place of its parameters (you can't use a statement like F$ = "SAMPLE": VIEW F$).

The file display mode of File Viewer is intended primarily for looking at text files—that is, files that consist of printable character codes. It is possible to view other types of files such as tokenized BASIC programs, but the display may be difficult or impossible to read in such cases. For instance, if the file contains values equivalent to control-code characters, displaying the file may clear the screen, change the printing color, and so forth. If you try to print a hardcopy of such a file, some of the values may be interpreted as spurious printer control codes, causing strange behavior such as unwanted form feeds or a change in printing mode. Similarly, disassembling a file that doesn't contain a machine language program produces meaningless results, but does no real harm.

The machine language for File Viewer occupies memory locations 4944–6114 ($1350–$17E2). If you disturb the contents of this area while File Viewer is active, the computer will probably lock up. The program also uses locations 4864–4937 ($1300–$1349) and 250– 254 ($FA–$FE) for temporary storage. You may use these locations for your own purposes; however, every VIEW statement will overwrite the contents of these areas.

File Viewer also maintains a six-byte bank-switching routine beginning at location 2048 ($0800) in both banks 0 and 1. This is necessary because some BASIC errors may occur while the system is operating in bank 1. In bank 0, this area is at the bottom of the BASIC runtime stack, where it isn't likely to be disturbed unless you run a program that uses a very large number of nested GOSUBs or FOR-NEXT loops. To accommodate this routine in bank 1, File Viewer bumps the start of variables up to address 2054 ($0806) when you first BOOT the program, thereby protecting the bank 1 copy of the switching routine from being destroyed by the system. This results in a slight reduction of the amount of space available for scalar variables, but should have no noticeable effect on the operation of BASIC.

See program listing on page 102.

64 RAMdisk

Hubert Cross

This Commodore 64 utility creates an electronic disk drive that's much faster than an ordinary disk drive. Since it uses "hidden" memory, the RAMdisk doesn't reduce the amount of programming space available for your use. No machine language knowledge is needed to use the program.

A RAM disk is a familiar device to many personal computer owners. In simple terms, a RAM disk emulates a disk drive entirely in the computer's RAM (Random Access Memory), allowing you to store and retrieve files much faster than you can from a mechanical disk drive. The Amiga, for instance, includes a built-in RAM disk as part of its system software; and RAM disks are popular utility programs for computers such as the Atari ST and IBM PC/PCjr.

"64 RAMdisk" is a RAM disk for the Commodore 64 which doesn't subtract a single byte from the space normally available for programming. You control the RAMdisk with simple BASIC commands, and the program is compatible many disk utilities.

Typing In 64 RAMdisk

Because 64 RAMdisk is written in machine language, the program must be typed with the "MLX" machine language entry program printed elsewhere in this issue. Here are the addresses you need to enter 64 RAMdisk with MLX:

Starting address: 0801
Ending address: 12A8

64 RAMdisk is designed to load and run exactly like a BASIC program. Load the program with LOAD "RAMDISK",8 for disk or LOAD "RAMDISK" (or simply LOAD) for tape. After the program loads, type RUN and press RETURN. The program relocates its code to the safe memory area beginning at location 49152 and prints the message RAM DISK ACTIVATED. 64 RAMdisk is now ready to use.

RAMdisk Commands

Following is a list of 64 RAMdisk commands. All of these commands work in BASIC direct mode (when you aren't running a program).

DIR. This command displays a directory of the files in the RAMdisk. For instance, if you type DIR and press RETURN, 64 RAMdisk prints a directory of the RAMdisk. The number of bytes free for use is printed at the bottom of the directory display.

NAME. The NAME command can be used to change the RAMdisk's name, which appears in reversed characters at the top of the directory display.

RSAVE. The RSAVE command saves a file to the RAMdisk, storing a copy of the BASIC program currently in memory. Here is the correct syntax for the command:

```
RSAVE "PROGRAM"
```

Of course, you should substitute the name of your program for PROGRAM in this example. If you forget to specify a filename, 64 RAMdisk prints the error message MISSING FILENAME and doesn't save anything. The filename can be any combination of 15 or fewer characters.

RLOAD. The RLOAD command copies any program in the RAMdisk back to BASIC. For example, the statement RLOAD "EXAMPLE" loads the program named EXAMPLE from the RAMdisk, storing a copy of it in the usual BASIC program space.

SCRATCH. The SCRATCH command deletes a file from the RAMdisk. For instance, the statement SCRATCH "PROGRAM" removes the file named PROGRAM from the RAMdisk. If you attempt to scratch a file that doesn't exist in the RAMdisk, the program prints the error message FILE NOT FOUND and doesn't scratch anything.

RENAME. This command changes the name of a file in the RAMdisk. Here is the syntax to use:

```
RENAME "OLDNAME","NEWNAME"
```

The RENAME command requires two filenames: the name of the existing file (OLDNAME in this example) and the new name which you want that file to have (NEWNAME in this example). The filenames are separated with a comma. The error message FILE NOT FOUND appears if you attempt to rename a nonexistent file. The error message FILE EXISTS appears if

you try to use a new name which already exists in the RAMdisk. (Every file in the RAMdisk must have a unique name.)

REPLACE. This command replaces the designated RAMdisk file with the BASIC program in memory, using the same filename. For instance, REPLACE "TEST" deletes the program TEST from the RAMdisk and saves the BASIC program in memory, using the filename TEST. This is normally done when you have made changes to a program and wish to save the revised version with the same name.

RNEW. The RNEW command does a NEW of the RAMdisk, erasing every file that it contains. *Be very careful when using this command,* since the program does not ask you to confirm this action, and there is no easy way to recover files after an RNEW. If you include a name after RNEW, this command renames the RAMdisk with the name specified. For instance, this command erases everything in the RAMdisk and renames it as MYDISK:

```
RNEW "MYDISK"
```

EXIT. This command disables the RAMdisk and gives you the option to save the entire contents of the RAMdisk in a single tape or disk file. When you type EXIT and press RETURN, the program prints this prompt:

```
EXIT RAM DISK
ARE YOU SURE? (Y/N)
```

If you type N at this prompt, nothing happens. If you type Y at this prompt, the program copies the 64 RAMdisk machine language program, as well as all the files in the RAMdisk, back into the BASIC program space. This is done so that you can save the entire RAMdisk as a single file, using a normal SAVE command from BASIC. The advantage of this method is that you can reload the RAMdisk program *and all of the saved files* in one operation, at the beginning of your next programming session. You do not need to use any special tricks to save this file. Here is a typical SAVE command:

```
SAVE "CABOODLE",8
```

This example saves 64 RAMdisk and the entire contents of the RAMdisk under the filename CABOODLE. If you replace the 8 with a 1, the file is saved to tape instead of disk. (The resulting file will be considerably longer than 64 RAMdisk itself, since it contains a copy of every file in the RAMdisk, as well as a copy of 64 RAMdisk.)

Once you have saved a master file, you can reload 64 RAMdisk and the individual files with the same load and run commands you would ordinarily use to activate the RAMdisk. In this case, for instance, you would use these commands:

```
LOAD "CABOODLE",8
RUN
```

If you type RUN and press RETURN at this point, the program moves the 64 RAMdisk code to its normal location beginning at location 49152, then transfers each of the saved files to the RAMdisk, too. When the startup message appears, the RAMdisk is ready to use and all of the files are in place.

Notice that the filename used with EXIT has no connection with the name of the RAMdisk itself (see NAME) or the names of individual files contained in the RAMdisk. As a practical matter, however, you will probably want to use a name that reminds you what individual files the master file contains. If you previously used NAME to give the RAMdisk a meaningful name, you can use the same name when saving the entire disk with EXIT.

RAMdisk Notes

64 RAMdisk is designed as a convenience for saving and loading BASIC programs, not as a total replacement for a disk or tape drive. Thus, it supports only one type of file—a BASIC program (PRG) file—and only one form of file access (saving and loading). You cannot use the RAMdisk for other types of files, such as sequential (SEQ) files. And, for instance, you cannot OPEN a file in the RAMdisk for reading or writing, even though those are legitimate operations for program (PRG) files on tape or floppy disk.

Like all RAMdisks, 64 RAMdisk is volatile, meaning that it disappears completely, together with all its contents, when you turn off the computer. For this reason, you should make frequent backup copies of RAMdisk programs on disk or tape. To make a backup copy, load the program into BASIC memory with RLOAD; then save it to disk or tape in the usual way.

This program occupies the memory area beginning at location 49152 ($C000), so you cannot use it with any other machine language program or utility that occupies the same space. Because this program uses the "hidden" RAM underlying the 64's ROM (Read-Only Memory) chips, it is also incompatible with programs which use that area of RAM. 64 RAMdisk *does* work with either TurboTape or TurboDisk, but not with both at the same time, since those programs are incompatible with one another. You must relocate TurboDisk, as explained in the TurboDisk article, before using it with 64 RAMdisk; the best place to put TurboDisk is as close as possible to the top of BASIC RAM.

The 64 has a total of 16K (16,384 bytes) of RAM under its ROM chips. Half of this lies under the BASIC ROM, and the other half lies under the Kernal operating system ROM. Not all of this RAM can be used for file storage with 64 RAMdisk. Every program stored in the RAMdisk requires an extra 18 bytes for a directory entry, 15 bytes for a filename entry, 1 byte as a filename marker, and 2 bytes for a pointer to the beginning of the next program.

Programs are stored beginning at the bottom of the RAM under BASIC and growing upward, toward higher memory locations. The directory begins at the top of the RAM under the Kernal and grows downward, toward lower memory locations. If you fill the 8K space under BASIC with programs, 64 RAMdisk uses as much of the RAM under the Kernal as needed. If you try to save a program that's bigger than the amount of free space left in the RAMdisk, the program prints the error message RAM DISK FULL and doesn't save anything.

It's theoretically possible to create so many individual files that the RAMdisk directory would fill all of the RAM under the Kernal ROM. However, since it would require more than 454 files to overflow the directory, 64 RAMdisk does not check for this unlikely situation, and does not print an error message if it occurs.

See program listing on page 100.

Off-Screen Trace

Brent Dubach

BASIC programmers will appreciate this testing and debugging utility. It traces program execution line by line and displays the trace on an alternate screen so there's no interference with your program display. It also allows you to control the speed of the execution. For the Commodore 64 and 128 (in 64 mode).

Tracing the path of program execution is helpful when you're debugging a program. Most TRACE commands in BASIC languages print executing line numbers on the same screen as the program being traced. While acceptable for some programs, in others this approach so disrupts formatted text output that it loses much of its debugging value. For example, in programs that use a high-resolution graphics screen, tracing execution becomes awkward since the normal text screen cannot be seen so long as the program has the machine in the hires graphics mode.

Here's a trace utility for the Commodore 64 that avoids these problems by setting up its own display screen. Your own program output is not disturbed, and the trace information can be seen at the touch of a key. And the line displayed is the line *after* the line which has just been executed.

How To Use The Trace

Since "Off-Screen Trace" is written entirely in machine language (ML), you'll need to use MLX to type it in. When prompted for the starting and ending addresses, respond with the following:

Starting address: 8800
Ending address: 8AFF

Be sure to save a copy before running the program.

Although Off-Screen Trace is written in ML, you don't need to know anything about ML in order to use it. Once in memory, the trace program is activated by a **SYS 34816** command and deactivated by **SYS 34970**. These commands should be issued only in direct mode and not from within a program. If you want to trace only a part of the program, you may use RUN or GOTO followed by the line number at which you want to begin and then either a STOP command or the STOP key to halt execution. Tracing may be turned off at any time after it has been switched on. *Be sure, however, that you do not follow a SYS 34816 with another SYS 34816 without deactivating tracing in between.* If you do this, you'll need to turn the machine off and back on and start all over.

Once Off-Screen Trace is activated, pressing CTRL-O (for Other screen) will allow you to see a separate trace display that shows the line numbers in which statements are being executed. Pressing CTRL-O again returns you to whatever output screen your program is using. This keypress is not "debounced" by the program and therefore has a very light touch. Use a quick, crisp keypress to toggle between the two displays.

Controlling Execution Speed

Because of the overhead involved in tracing execution, programs always run more slowly while being traced. This utility allows you to control the speed of execution with the space bar. It allows statements to be executed only when it sees that the space bar is held down. By holding it down, you can keep the program executing at maximum speed, and you can also step through statements one at a time with a quick single press of the space bar. Regardless of which screen is in view at the time you press the space bar, your own program screen will be reinstated before the next statement in the BASIC program is executed. Another CTRL-O will retrieve the updated trace display.

Instead of an unstructured sequence of line numbers, the trace display shows executing line numbers in an outline format that reflects the organization of your program. Each time a GOSUB is encountered, the trace display is indented two spaces. With each RETURN it is moved two spaces to the left of the left margin. Thus, you can easily tell by monitoring the line number display whether your program is entering and exiting subroutines as you intended.

Memory Considerations

The program uses memory at the top of the range normally available for BASIC program text and variables. In addition to memory for machine language itself, memory is needed for the separate display screen and for saving certain important information when toggling between displays. When activated with SYS 34816, the trace program protects itself and its display screen from incursions by BASIC by setting 33792 as the top of memory available to BASIC. This still allows BASIC programs that need almost 32K of memory and leaves all the typically used sprite and machine-language areas available. You should note that some of this range of memory is the same as that used by "MetaBASIC," so you should disable "MetaBASIC" before loading and using this trace facility.

See program listing on page 98.

COMPUTE! Books'

WAREHOUSE CLEARANCE SALE

You can order any of the following groups of books for $9.95*. Or, choose any three books from anywhere in this ad and pay $9.95* for all three. Each additional book after 3 is just $2.00. (If you want only one or two books, the cost is $4 each.)

Check the books or groups of books you want and mail this coupon with your payment to:

COMPUTE! Books
Customer Service
PO Box 5038
F.D.R. Station
New York, NY 10150

* Add $2 per order for shipping and handling.

Name _____

Address _____

City _____ State _____ Zip _____

☐ VISA ☐ Mastercard
Acct. No. _____ Exp.Date _____
☐ U.S. money order
☐ Check drawn on a U.S. bank

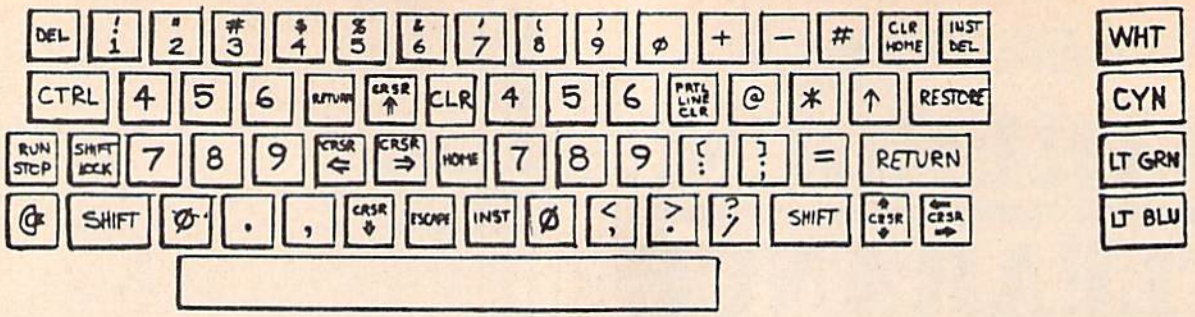
Subtotal $_____
Sales Tax (NC residents add 5%; NY residents add 8.25%) $_____
Shipping and handling $2.00 $ 2.00

Total payment enclosed $_____

Offer good while quantities last. All sales final.
Please allow 4–6 weeks for delivery.

☐ **Group I**	**$9.95**
☐ COMPUTE!'s Beginner's Guide to Commodore 64 Sound	238654X
☐ COMPUTE!'s First Book of Commodore 64 Sound and Graphics	2386213
☐ COMPUTE!'s Reference Guide to Commodore 64 Graphics	2386299
☐ **Group II**	**$9.95**
☐ Every Kid's First Book of Robots	2386051
☐ Commodore 64 Games for Kids	238637X
☐ COMPUTE!'s Computing Together	2386515
☐ All About the Commodore 64, Vol.1	238640X
☐ **Group III**	**$9.95**
☐ COMPUTE!'s Commodore Collection, Vol. 2	2386701
☐ COMPUTE!'s Commodore Collection, Vol. 1	2386558
☐ COMPUTE!'s First Book of Commodore 64	2386205
☐ **Group IV**	**$9.95**
☐ COMPUTE!'s Third Book of Commodore 64	2386728
☐ Creating Arcade Games on the Commodore 64	2386361
☐ COMPUTE!'s First Book of Commodore 64 Games	2386345
☐ **Group V**	**$9.95**
☐ COMPUTE!'s Data File Handler for the Commodore 64	2386868
☐ COMPUTE!'s Telecomputing on the Commodore 64	0092
☐ Commodore Peripherals: A User's Guide	2386566
☐ **Group VI**	**$9.95**
☐ COMPUTE!'s First Book of VIC	2386078
☐ COMPUTE!'s Second Book of VIC	2386167
☐ COMPUTE!'s Third Book of VIC	2386434
☐ Mapping the VIC	2386248
☐ **Group VII**	**$9.95**
☐ COMPUTE!'s VIC Games for Kids	2386353
☐ Creating Arcade Games on the VIC	2386256
☐ COMPUTE!'s Second Book of VIC Games	2386574
☐ COMPUTE!'s First Book of VIC Games	2386132

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company



64+

Barry L. Camp

The new commands that this utility adds to the 64 make editing and programming much more enjoyable. "64+" alters the operating system itself to create a new working environment.

"64+" is a full-scale enhancement to the 64's operating system. It works by copying the entire contents of the BASIC and Kernal ROMs into the underlying RAM and then making modifications to add new features.

64+ consists largely of new CHR$ and keyboard commands that perform useful functions. Other capabilities include a more useful USR function, preprogrammed function keys, disk drive defaults, and a numeric keypad.

Typing It In

Since 64+ is a machine language program, you will need to type it in using "MLX," the machine language entry program found elsewhere in this issue. Be sure you are familiar with MLX before you begin typing in 64+. When MLX asks for starting and ending addresses, respond with the following:

Starting address: 0801
Ending address: 0C40

Type in the data, and be sure to save a copy before leaving MLX.

Although 64+ is written in machine language, it loads and runs just like a BASIC program. To get started, load 64+ and type RUN. You will notice some changes immediately. The lowercase character set is switched in and the screen colors change. These visual cues also appear every time you press RUN/STOP–RESTORE. This different visual appearance serves as a reminder that that you are operating under 64+. Try out the new commands and keystrokes provided by 64+. The accompanying quick-reference table and keyboard map are all you need to begin experimenting.

Most of the new commands can be used in three ways. First, you can use them in immediate mode by typing the proper combination of keys (especially helpful when you are editing a BASIC program). Second, you can use the CHR$ function in a PRINT statement to print the corresponding character code for the command. For example, PRINT CHR$(7) will ring a bell in 64+. Finally, you can type the key combinations within quotes in PRINT statements. In quote mode, the commands appear as reverse letters. The quick-reference table has all the information you need to use the commands in any of the three ways mentioned.

New Characters

One of the most powerful features of 64+ is the addition of 15 new characters. These can be accessed by the CHR$ function or typed directly from the keyboard. Machine language programmers can use Kernal routines to access these new characters.

Here is a description of the characters which may not be self-explanatory. To find the corresponding keystroke and the character that appears when you use the command within a string, refer to the quick-reference table and the keyboard map.

Tab
This is a version of the comma delimeter used in PRINT statements. PRINT "Hello","there" and PRINT "Hello there" both have the same effect. Press the STOP key for this character. In immediate mode, the STOP key tabs the cursor.

Line home
This moves the cursor back to the beginning of the current logical line. For example, when editing a BASIC line, hold CTRL and press the back-arrow key to place the cursor on the first digit of the line number.

Partial screen clear
Clears everything below the current cursor position.

Partial line clear
Clears from the cursor position to the end of the current logical line. This is especially powerful for editing BASIC lines. Depending on how the cursor is positioned, it is possible either to trim unwanted or unneeded statements off the end of a line (like REM or STOP statements), or to delete the entire line by erasing all but the line number (and then press RETURN).

Delete
This is similar to the normal delete except that instead of pulling part of a line back over the unwanted characters to the left, the cursor remains stationary, and unwanted characters are pulled into the cursor from the right. Use the CTRL or Commodore key in conjunction with the

INST/DEL key for this "black hole" delete.

Repeating-key control
If you want all keys to repeat, use CHR$(22). If you don't want any to repeat, use CHR$(23). The default is CHR$(24), which corresponds to the setting of a normal 64—only the cursor keys.

Set color RAM
This sets every location on the screen to the current cursor color.

Escape
This may prove to be the most useful of the new characters. It cancels insert, reverse, and quote modes. It also resets the default key-repeat values and resets the SID chip registers to their default values.

Additional function-key characters
Normally there are eight function keys: f1, f3, f5, and f7 are unshifted, while f2, f4, f6, and f8 are obtained in conjunction with the the SHIFT key. 64+ adds four more function keys (f9–f12), which are entered by pressing the Commodore key in conjunction with the function keys.

Other Features

In addition to the new characters, these features are also part of 64+:

• Freeze. Whenever the SHIFT-LOCK key is engaged, the screen stops scrolling, effectively freezing it. This is handy for examining listings or anything else that is too large to fit on one screen. To continue normal scrolling, release the key.

• Four preprogrammed function keys. By holding down CTRL and pressing one of the function keys, the corresponding one of the following four BASIC keywords is automatically printed to the screen: DATA (f1), LIST (f3), RUN (f5), SYS (f7).

• Disk drive defaults. Device 8 is now the default device for LOAD, SAVE, and VERIFY. It is no longer necessary to add ,8 to the command for disk loads and saves. (However, it is now necessary to add ,1 to the SAVE, LOAD, or VERIFY command when using tape.) Note also that SHIFT-RUN/STOP has also been altered to facilitate the new device change. That key combination will now load and run the first program on the disk.

• End-of-line warning system. When you are near the end of an 80-character logical line, 64+ notifies you with a tone. This is especially useful when entering large amounts of data while looking away from the screen.

• Audible error messages. A bell-like tone sounds whenever an error occurs in a BASIC program. This is intended for those times when you are unable to see error messages (when programming with a graphics screen, for example).

• Numeric keypads for quick data entry. The keyboard map shows a special keyboard configuration, accessible by holding down the Commodore key and pressing CTRL. Notice that a # appears under the cursor. This is to let you know that the keyboard is in *numeric mode* (which affects unshifted keys only). Now you have one-touch access to important control characters, 4 of the 16 available colors (for text), and the new numeric keypads. Notice that the left group of numbers is in a different order than the right. This allows you a choice between the telephone- and calculator-style keypad formats. When you are ready to exit numeric mode, just press Commodore-CTRL again. This time a \ will appear, and the keyboard will be back to normal.

Keyboard Map

Quick Reference Table

Description	Character Code	Keyboard Access	Appears As
Tab	3	RUN/STOP	reverse c
Line Home	6	CTRL-1 or Commodore-1	reverse f
Bell (Tone)	7	CTRL-G	reverse g
Partial Screen Clear	11	CTRL-CLR/HOME or Commodore-CLR/HOME	reverse k
Partial Line Clear	16	CTRL-RETURN or Commodore-RETURN	reverse p
Delete	21	CTRL-INST/DEL or Commodore-INST/DEL	none
All Repeat	22	CTRL-+	reverse v
No Repeat	23	CTRL-−	reverse w
Default Repeat	24	CTRL-£	reverse x
Set Color RAM	26	CTRL-Z or Commodore-0	reverse z
Escape	27	SHIFT-=, CTRL-=, or Commodore-=	none
f9	128	Commodore-f1	reverse —
f10	130	Commodore-f3	reverse b
f11	132	Commodore-f5	reverse d
f12	143	Commodore-f7	reverse o

A Useful USR

Machine language programmers will appreciate the new application of the rarely used USR function. It can now be used to find the values of system vectors. To use it, simply pass the address holding the low byte of the vector. For example,

```
NMI = USR(65530):PRINT NMI
```

would return the value 65091, the starting address of the NMI routine. Calculating available memory is also a breeze, with this short formula:

```
MEM = USR(51)−USR(49)
```

This example subtracts the values of the vectors that point to the top and bottom of available memory.

Memory Conflicts

Be careful when writing programs that use POKE statements, because BASIC and the Kernal are now in RAM. Also, 64+ uses location 787 ($0313) for various flags, as well as locations 51617–52223 ($C9A1–$CBFF). Stay away from these areas of memory to avoid a crash. One commonly used utility that works with no conflicts is the *DOS Wedge*, which resides in memory immediately following the locations used by 64+.

64+ will probably not work with your favorite word processor. Word processors are generally self-contained machine language programs that provide their own special editing functions and other features. For instance, *SpeedScript* and 64+ will not work together.

See program listing on page 98.

Printer Wedge

James Chandler, Jr.

If you have a 64 and a Commodore 1525, MPS-801, or MPS-803 printer, here's an easy way to create, save, and print out your own custom characters. With "Printer Wedge," you can have true descenders, foreign language character sets, and customized graphics.

The Commodore 1525, MPS-801, and MPS-803 printers are inexpensive and generally reliable printers. But the printed characters leave much to be desired—no descenders. If you own one of these printers, you know how difficult it is to discern the difference between a lowercase and capital "P" or a lowercase "g" and the number "9." This program, "Printer Wedge," lets you define a new character set—including true descenders or foreign language characters.

The program is written in BASIC, but it POKEs a machine language "wedge" into memory. You don't have to know machine language to make it work, however. The wedge prints the user-defined characters in conjunction with many programs, including *SpeedScript*.

Printer Wedge characters can be any width from one to ten pixels, and any number of characters can be created, from one replacement character to an entire upper/lowercase font. Also, foreign language character sets can easily be defined (within the limits of printer resolution).

The program also has another mode which allows substitute character strings of up to 40 characters. For example, one could specify text replacements for LISTed graphics symbols (cursor controls, colors, etc.), allowing the printing of more readable program listings.

Creating Custom Characters

When you load and run Printer Wedge, you get a character creation screen. Characters are created on a grid seven high by ten wide. Move around the grid using the cursor keys, and plot dots with f7. Delete dots with f8. When the character is complete, assign the character with f1. The computer then prompts you for the key of the character. Press the key representing the character and the program appends a DATA statement to itself describing that character. That character will no longer be available to you—it will be replaced by the new character you've designed. It then loops back to the character creation screen.

When you're through creating the characters you wish to use, press f2. The program asks for a starting address, then POKEs the wedge to the specified location. You are then given the option of sending sample lines to the printer so you can inspect the new characters.

If you wish to alter any of the characters you've created, run the program again and press f2 when you finish. Since the character set is stored as part of Printer Wedge, you can create many character sets and save each by its own name, such as PWEDGE/ELITE or PWEDGE/SPANISH.

Create characters starting from the left column of the character creation grid. When printing, the wedge inserts one blank column at the right of a user-defined character. This allows proportional spacing of characters. For replacement characters to be in the same seven by five format of the 1525, 801, 803 character set, do not allow two adjacent blank columns in the first five columns of the character creation grid. In addition, do not leave column five blank.

To specify characters that cannot meet the above requirements, calculate the column values as in the 1525, 801, or 803 users' manual. Append a DATA statement to Printer Wedge using the following format:

*[line number 3000 + 10 * ASCII]* **DATA** *[ASCII value of character],[number of character columns],[column 1],[column 2],...,[last column]*

You can find the ASCII value of a character by typing (in immediate mode):

PRINT ASC("*[character]***")**

The line 10000 DATA 0 tells Printer Wedge it has read and POKEd all characters into memory. A replacement character for the double quote (SHIFT-2) should normally be specified (line 3340). Otherwise, printing a double quote character causes a graphic nightmare.

I've included three character sets as options for you to try. Any of these can be appended to Program 1. Character Set 1 (Program 2) replaces the lowercase g, p, and q with less ambiguous characters. Character Set 2 (Program 3) proportionally spaces without descenders. Character Set 3 (Program 4) squeezes characters into the top six rows, allowing one row for descenders. Program 5 allows for converting graphics symbols to text.

Character Strings

Press f3 to assign a string of characters to a key. This can be useful in a couple of ways. You can make program listings more readable by replacing the reverse-heart that means "clear the screen" with the easier to read string [CLR]. To do this, type in the string [CLR] and assign it to the SHIFT-CLR/HOME key (by pressing f3). After entering all replacement strings in this manner, save PWEDGE/LIST. Programs can then be listed to disk or tape, then read back and printed with a short BASIC routine. To do this, first open a file to tape or disk, then enter **CMD** *(file number)*:**LIST**. After the file is created, type **PRINT#** *(file number)*:**CLOSE** *(file number)*. To read the file (and list it to the printer), open the file for input, open a line to the printer, use GET# to get characters, and PRINT# to send it to the printer.

SpeedScript allows the use of the Commodore key subset of the keyboard (left graphics symbols on the keys). [Commodore-D] could represent "Dear Friends," [Commodore-Y] could represent "Yours Truly," and so on. This would allow automation of the repetitive parts of letters or of any document form you use frequently.

See program listings on page 123.

One-Touch Function Keys

Keith Ashcraft

This utility programs the 64's function keys for a variety of useful tasks. And best of all, it provides a menu of its functions on the top screen line. A disk drive is required.

The Commodore 64's function keys are a highly useful feature. Because they're programmable, they can be tailored to the needs of any user. "One-Touch Function Keys" adds four functions—including a directory listing—to your 64. And you don't have to worry about forgetting which key does what; they're always listed at the top of the screen.

Because the program is written entirely in machine language, it must be typed in using "MLX," the machine language entry program found elsewhere in this issue. Be sure to read and understand the instructions for using MLX before you begin typing in the program. When you run MLX, you'll be asked to provide starting and ending addresses for the data you'll be entering. The correct values are:

Starting address: C000
Ending address: C11F

After you finish typing in the program, be sure to save a copy before leaving MLX. To load One-Touch Function Keys, use a statement of the form LOAD "*filename*",8,1. (Substitute ,1,1 for the ,8,1 if you're using tape instead of disk.) Type **SYS 49152** to activate it. You should see the definition line on the top of the screen.

Here's a list of the functions that One-Touch Function Keys adds to the 64:

- **f1**—The Directory
  Press f1 at any time for a directory listing. RUN/STOP halts the listing.
- **f3**—List
  The BASIC program currently in memory is listed when f3 is pressed.
- **f5**—Save
  Press f5 for SAVE to be printed on the screen.
- **f7**—Load
  Press f7 for LOAD to be printed on the screen.

When you use Load or Save (f5 and f7), you must insert the filename of the program and press RETURN. For example, if you wanted to load a file named GAME.OBJ, you would press f7 and then "GAME.OBJ",8,1.

RUN/STOP-RESTORE disables the program, but typing SYS 49152 reactivates it. Try to avoid using One-Touch Function Keys with other programs that use the RAM area at 49152.

See program listing on page 100.

Sequential File Editor

Paul Piciocchi

This menu-based program makes modifying sequential files easy, with options for editing, inserting, deleting, resaving, and printing out sequential files. For the Commodore 64, Plus/4, and 16. A disk drive is required.

Sequential files can hold a wide variety of information. When a game saves the all-time high score, chances are it puts the information in a sequential file. When you save an adventure game position to disk, the data probably goes into a sequential file. Programs that include filenames and addresses often store the data in sequential files.

Modifying a sequential file can often be a tedious chore. Sometimes you can load and run a word processor to do the job. But the file might not conform to the word processor's format (some use ASCII characters stored in sequential files, others might use screen codes in program files). Or you might not own a word processor. You're not left with much of a choice if that's the case: Either you'll have to retype the contents of the whole file or you'll have to write a program that can read the file into memory for you to make modifications—not an easy task. This program, "Sequential File Editor," provides a simple solution—it reads a sequential file into memory and then offers a menu with several editing options. It runs on the Commodore 64, Plus/4, and 16.

Using The Editor

After typing in the program, save a copy to disk. To use it, load it and type RUN.

You're first asked for a filename and disk device number (which should be 8, unless you have multiple drives). Next you're asked if what you typed is correct. If you reply Y, the program starts to read in the file (if that file is not on the disk, you'll be told). While the program reads the file, the contents are displayed as individual lines, along with assigned line numbers. (A carriage return in the sequential file determines the end of each line.) If you plan to do complex editing, you may want to write down the line numbers you plan to modify.

Once the file is read, you're given a simple menu. The commands include the following:

1. List File. You're asked if you wish to see the file with line numbers (so you can find the line you wish to edit) or without line numbers (to see the final product). Press either W or O. While listing, you can pause the output by pressing the space bar or abort the output by pressing A. Resume the listing by pressing the space bar again.

2. Edit Line, and **3. Insert Line.** If you select either of these, you're asked for a line number. If you choose to edit, you're shown the line as it is. Otherwise, the program will make space within the file to insert that line. Next you're asked to change or write the line. In either case, pressing RETURN keeps the line as it is. (When editing or inserting lines, you can enter up to 254 characters. The program will prevent you from going beyond this limit. Control characters are not allowed as input, except for DEL—CHR$(20)—which deletes single characters, and CTRL-X, which erases an entire line.) The program works with sequential files as large as 3000 lines. On the Commodore 16, however, the limit is 100 lines, due to limited memory.

4. Delete Line. This allows you to remove a line from the file. After specifying the line number, you're shown the line and asked if you are certain you wish to delete it.

5. Rerun Program. If you finish with a file and wish to edit another, you can rerun Sequential File Editor. Make sure to use the next command (resave file) if you want the changes you've made to be written to disk. The rerun option starts over with the initial filename prompt.

6. Resave File After Editing. It's important that you remember to do this. When you finish editing a

file, you must resave the file to disk to make the changes permanent. Should you forget, the file will remain as it was before editing. When prompted for a name to resave the file, you can type an asterisk (*) to resave with the original filename. This command doesn't use the sometimes unreliable save-with-replace function; it actually scratches the old file before the new information is written out.

7. Print File To Printer. If you wish to have a printout of the file, choose this option. You're asked for the number of lines per page (this number is usually 66). The program will format the file into pages, and leave five blank lines on the top and bottom of each sheet. If you do not want the file formatted into pages, enter 0 as the number of lines per page on your paper.

8. Exit Program. Once you've finished editing and have resaved the file to disk (option 6), you can exit the program.

Word Of Warning

Because strings are limited to a maximum length of 255 characters, this program may crash with an error in line 190 when a file contains 256 or more characters without a RETURN character (CHR$(13)). The following line will allow such files to be read into the program:

```
205 IF LEN(A$(LN))=255 THEN PRINT
    A$;:GOTO 230
```

Note that if you add this line and subsequently read a file and write it back to disk, the new file may contain some additional CHR$(13) characters.

See program listing on page 99.

64Key

Daniel Bingamon

This might be the last program you type in the traditional way. 64Key allows you to print a BASIC instruction just by pressing one shifted key.

"64Key" is a Commodore 64 version of Thomas Henry's "VICKey," originally published in the August 1982 issue of COMPUTE!. It allows you to print a BASIC keyword just by pressing one shifted key. For example, pressing SHIFT-A prints ASC, SHIFT-B prints STEP, SHIFT-C gives CHR$, and so on. It makes typing in long programs faster and easier.

To use 64Key, type in the program, SAVE it, and type RUN. If you have mistyped a DATA statement, the program will tell you so. The program POKEs 64Key into the $C000 block of free RAM. Type SYS 52557 to activate 64Key. You will get the READY message back as though nothing has happened. But try typing a SHIFTed letter. Presto! A keyword appears.

Now instead of typing out a keyword, you can simply hit the appropriate key as shown in the table. Of course, you can always type the keyword normally (you must if the keyword is not in the table). Be sure to type NEW before typing in your program.

64Key checks location $D4 (212) to see if the editor is in quote mode. If you hit a SHIFTed letter within quotes, you will get the normal graphics character or capital letter instead of a keyword. Typing SYS 52557 turns 64Key on or off. So, if you want to deactivate it, type SYS 52557 a second time. You can also deactivate it by pressing the RUN/STOP and RESTORE keys together. 64Key will remain in memory until you turn the computer off.

"64Key" uses the same area of memory as the DOS Wedge program supplied with the 1541 demo disk. To use these two valuable utilities together, change the following lines:

```
20 FOR I=51789 TO 51967
50 IF X<>23743 THEN PRINT "THERE
   IS AN ERROR IN YOUR DATA
   STATEMENTS":END
60 PRINT"SYS 51789 TO
   ACTIVATE":END
```

Next, change the DATA element 205 to 202 in the following lines: 100, 120, 130, 140, 150, 190, 220, 300, and 320. Finally, remove the ,0 from the end of line 430 and delete line 440. These changes relocate 64Key to the area immediately above the Wedge, allowing the two to coexist in harmony and still leaving locations 49152–51788 free for other uses.

BASIC Keywords

A	ASC	N	NEXT
B	STEP	O	OPEN
C	CHR$	P	POKE
D	DIM	Q	PEEK
E	END	R	RIGHT$
F	FOR	S	STR$
G	GET	T	TAB(
H	STOP	U	USR
I	INPUT	V	VAL
K	GOSUB	X	READ
L	LEFT$	Y	RESTORE
M	MID$	Z	SYS

See program listing on page 91.

No-SYS Loader

Walter L. Smith, Jr.

Now machine language programs can be as easy to handle as BASIC programs. With this 64 utility, running, loading, and saving most machine language programs is a snap. A disk drive is required.

Machine language (ML) programs are more difficult to load, run, and copy than their BASIC counterparts. "No-SYS Loader" narrows the gap between the two, allowing most ML programs to be treated like BASIC programs.

No-SYS Loader works by combining the ML program with a short BASIC loader to produce a new program that can be loaded, saved, and run like a BASIC program. When the program is run, the loader transfers the ML to its correct location, performs a NEW, and, if necessary, resets certain BASIC pointers.

Typing It In

No-SYS Loader is written in machine language. It must be entered using the "MLX" machine language entry program found elsewhere in this issue. When you run MLX, you'll be asked for the starting and ending addresses of the data you'll be entering. For No-SYS Loader, respond with the following values:

Starting address: 0801
Ending address: 0A88

After entering the data, be sure to save a copy of the program.

To use No-SYS Loader, first, load it and type RUN. The program asks for a filename. Make sure that a disk containing a copy of the target program is in the drive; then enter the appropriate filename. No-SYS Loader finds the file on disk and tells you its load address.

No-SYS Loader next asks for the SYS address that starts the program. Either enter the address, or, since the load address and SYS address are the same, you may simply press RETURN.

No-SYS Loader now gives you the opportunity to insert a remark into the program. *Be sure that the remark you enter contains no more than 19 characters.* The program doesn't check the length of your input, and if you enter too many characters, the resulting program will crash when run.

Now insert the disk on which you want the converted program saved. Enter the filename for the converted file. The converted program is saved to disk.

To make sure that the program is working correctly, enter these lines:

```
LOAD"filename",8
LIST
```

You should see these lines on the screen:

```
10 SYS2088
20 REM your comment
```

The SYS address will always be 2088. This is the address of the loader, not the ML program. Run the program to see if it works properly.

Now that the program has been converted, it's easy to make copies. For example, to make a copy, type

```
LOAD"OLD NAME",8
```

Then change disks (if desired) and type

```
SAVE"NEW NAME",8
```

Notes

A limitation of this program is that it will not convert programs that load below address 2304 ($0900). You'll find, however, that there are few ML programs that load below this address.

If the converted program is to be used as a utility for a BASIC program, the converted program must be loaded and run first, or else it will erase any BASIC program currently in memory.

See program listing on page 102.

Programming Books

from COMPUTE!

COMPUTE! Books offers a line of programming books for the intermediate to advanced Commodore 64 and 128 users. These reference books take you beyond BASIC and into machine language programming, helping you learn about memory maps, addresses, the new *GEOS*, and time-saving routines you can add to your own programs.

COMPUTE!'s 128 Programmer's Guide

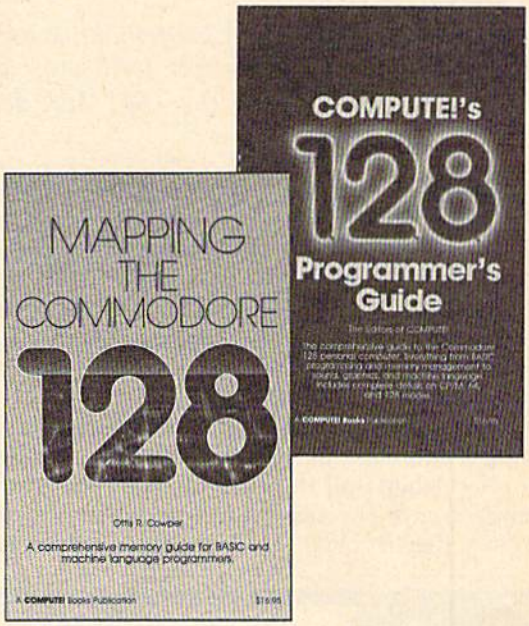
Editors of COMPUTE!
ISBN 0-87455-031-9 444 pages
A complete guide to the Commodore 128, this book explores BASIC 7.0, shows you how to create graphics and sounds, explains how to program peripherals, and introduces you to machine language programming.
$17.95

Mapping the Commodore 128

Ottis R. Cowper
ISBN 0-87455-060-2 704 pages
The comprehensive memory map and programmer's guide that provides a detailed explanation of the inner workings of the Commodore 128 including memory management, BASIC 7.0, I/O chip register, the operating system, system RAM, and more.
$19.95

Machine Language Routines for the Commodore 128 and 64

Todd Heimarck and Patrick Parrish
ISBN 0-87455-085-8 592 pages
This collection of machine language routines is a must for every Commodore 128 and 64 machine language programmer. Scores of these routines can simply be inserted into your own programs. Included is the assembly language code with easy-to-understand documentation and instructions. *There is a companion disk available for $12.95 that includes all the programs in the book* (858BDSK).
$18.95



Programming the Commodore 64 Revised: The Definitive Guide

Raeto Collin West
ISBN 0-87455-081-5 642 pages
This bestselling, encyclopedic reference guide which covers the Commodore 64 in its entirety has been updated to include information on the new Commodore 64C and *GEOS*, from Berkeley Softworks. *There is also a disk available for $12.95 which includes the programs in the book* (507BDSK).
$24.95

Mapping the Commodore 64 and 64C

Sheldon Leemon
ISBN 0-87455-082-3 324 pages
An update of the bestselling memory map and programming guide that's a necessity for intermediate and advanced programmers. This definitive sourcebook has been expanded and now covers the new icon-based *GEOS* (Graphics Environment Operating System) with clear descriptions of how to make it work for you. For BASIC and machine language programmers of both the Commodore 64 and 64C.
$16.95

These books are now available at your local book or computer store.
You can also order directly from COMPUTE! by calling toll free 800-346-6767 (in NY call 212-887-8525) or by mailing your order to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Customer Service hours are 10:00 am–12:30 pm and 1:30 pm–3:00 pm EST, Monday–Friday
Please include $2.00 postage and handling per book or disk. NC residents add 5 percent sales tax, and NY residents add 8.25 percent sales tax. Please allow 4–6 weeks for delivery.

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company

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

64 AutoBoot Maker

Rene Suarez

Here's a Commodore 64 boot-making program with a twist: It's compatible with most commercial accelerators for the 1541 disk drive.

Unlike most other computers, the Commodore 64 does not automatically *boot* (load and run) programs from disk. After loading a program, you must type either a RUN or SYS command to start it up. However, you've probably used commercial software that automatically loads and runs when you enter a statement such as LOAD"BOOT",8,1. Wouldn't it be nice to run your own programs this way?

Such programs are said to *autoboot* because everything is automatic after you enter the LOAD command. In effect, they pull themselves up by their own bootstraps; hence the term. Since autoboots require less typing, they're ideal for frequently used programs or programs intended for people who aren't familiar with the computer. The only catch is that autoboots can be very tricky to write. And though there are programs that construct them for you, the resulting package may not be compatible with the popular accelerator programs for the 1541 disk drive.

"64 Autoboot Maker" gives you the best of both worlds—an autoboot that works with most accelerated drives and normal drives, too. (Unfortunately, it won't work with the "TurboDisk" accelerator published in the July 1985 issue of the GAZETTE.) You can make an autoboot for any BASIC or machine language program, and even disable the RUN/STOP and RESTORE keys if you like.

Preparing An Autoboot

Type in 64 Autoboot Maker and save a copy, then run it. First it asks for the name of the target program (the one you want the autoboot to load and run). Since Autoboot Maker looks for the program on disk, insert the correct disk in the drive when you enter the filename. Next, enter a name for the autoboot. Of course, this name should be different from the original filename. For example, if the target program is named LASER, you could call the autoboot LASER/BOOT.

At this point, you must enter the memory address where the program begins. If the target program is BASIC, press RETURN without entering a number (BASIC programs always start in the same place). If the target program is machine language, enter the memory address where it normally begins. The address may be entered in decimal or hexadecimal (a hex number must begin with the $ symbol).

Finally, you have the option of making the autoboot disable the RUN/STOP and RESTORE keys. (Warning: With these keys disabled, you may not be able to break out of the autobooted program without turning off the computer.) Enter Y if you want them disabled and N if you don't. Once this is done, the disk drive spins for a short time and the status of the drive appears on the screen.

To test the newly created autoboot package, simply type LOAD *"filename"*,8,1 and press RETURN. Replace *filename* with the name of your autoboot, and don't forget the ,1 after the 8. The target program automatically loads and runs. (If it doesn't, one reason may be that it loads into the same memory area as the autoboot itself—locations $02A7–0303.)

Note that you must repeat the procedure to create the same autoboot file on another disk.

See program listing on page 75.

The Automatic Proofreader

Philip I. Nelson

"The Automatic Proofreader" helps you type in program listings for the 128, 64, Plus/4, and 16 and prevents nearly every kind of typing mistake.

Type in the Proofreader *exactly* as listed. Since the program can't check itself, type carefully to avoid mistakes. Don't omit any lines, even if they contain unfamiliar commands. After finishing, save a copy or two on disk or tape before running it. This is important because the Proofreader erases the BASIC portion of itself when you run it, leaving only the machine language portion in memory.

Next, type RUN and press RETURN. After announcing which computer it's running on, the Proofreader displays the message "Proofreader Active". Now you're ready to type in a BASIC program.

Every time you finish typing a line and press RETURN, the Proofreader displays a two-letter checksum in the upper-left corner of the screen. Compare this result with the two-letter checksum printed to the left of the line in the program listing. If the letters match, it's almost certain the line was typed correctly. If the letters don't match, check for your mistake and correct the line.

The Proofreader ignores spaces not enclosed in quotes, so you can omit or add spaces between keywords and still see a matching checksum. However, since spaces inside quotes are almost always significant, the Proofreader pays attention to them. For example, **10 PRINT"THIS IS BASIC"** will generate a different checksum than **10 PRINT"THIS ISBA SIC"**.

A common typing error is transposition—typing two successive characters in the wrong order, like PIRNT instead of PRINT or 64378 instead of 64738. The Proofreader is sensitive to the *position* of each character within the line and thus catches transposition errors.

The Proofreader does *not* accept keyword abbreviations (for example, ? instead of PRINT). If you prefer to use abbreviations, you can still check the line by LISTing it after typing it in, moving the cursor back to the line, and pressing RETURN. LISTing the line substitutes the full keyword for the abbreviation and allows the Proofreader to work properly. The same technique works for rechecking programs you've already typed in.

If you're using the Proofreader on the Commodore 128, Plus/4, or 16, *do not perform any GRAPHIC commands while the Proofreader is active.* When you perform a command like GRAPHIC 1, the computer moves everything at the start of BASIC program space—including the Proofreader—to another memory area, causing the Proofreader to crash. The same thing happens if you *run* any program with a GRAPHIC command while the Proofreader is in memory.

Though the Proofreader doesn't interfere with other BASIC operations, it's a good idea to disable it before running another program. However, the Proofreader is purposely difficult to dislodge: It's not affected by tape or disk operations, or by pressing RUN/STOP– RESTORE. The simplest way to disable it is to turn the computer off then on. A gentler method is to SYS to the computer's built-in reset routine (SYS 65341 for the 128, 64738 for the 64, and 65526 for the Plus/4 and 16). These reset routines erase any program in memory, so be sure to save the program you're typing in before entering the SYS command.

If you own a Commodore 64, you may already have wondered whether the Proofreader works with other programming utilities like "MetaBASIC." The answer is generally yes, *if you're using a 64 and activate the Proofreader after installing the other utility.* For example, first load and activate MetaBASIC, then load and run the Proofreader.

When using the Proofreader with another utility, you should disable *both* programs before running a BASIC program. While the Proofreader seems unaffected by most utilities, there's no way to promise that it will work with any and every combination of utilities you might want to use. The more utilities activated, the more fragile the system becomes.

The New Automatic Proofreader

```
10 VEC=PEEK(772)+256*PEEK(773)
   :LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADE
   R FOR ";:IF VEC=42364 THEN
   {SPACE}PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VI
   C-20"
40 IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
60 SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
70 FOR J=0 TO 166:READ BYT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +BYT:NEXT
80 IF CHK<>20570 THEN PRINT "*
   ERROR* CHECK TYPING IN DATA
    STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+L
   F,LB:POKE SA+HF,HB:NEXT
110 IF CHK<>22054 THEN PRINT "
   *ERROR* RELOAD PROGRAM AND
   {SPACE}CHECK FINAL LINE":EN
   D
120 POKE SA+149,PEEK(772):POKE
    SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
   14,22:POKE SA+18,23:POKESA+
   29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"
   PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
   EEK(LO)+256*PEEK(HI))-1,0:N
   EW
160 DATA 120,169,73,141,4,3,16
   9,3,141,5,3
170 DATA 88,96,165,20,133,167,
   165,21,133,168,169
180 DATA 0,141,0,255,162,31,18
   1,199,157,227,3
190 DATA 202,16,248,169,19,32,
   210,255,169,18,32
200 DATA 210,255,160,0,132,180
   ,132,176,136,230,180
210 DATA 200,185,0,2,240,46,20
   1,34,208,8,72
220 DATA 165,176,73,255,133,17
   6,104,72,201,32,208
230 DATA 7,165,176,208,3,104,2
   08,226,104,166,180
240 DATA 24,165,167,121,0,2,13
   3,167,165,168,105
250 DATA 0,133,168,202,208,239
   ,240,202,165,167,69
260 DATA 168,72,41,15,168,185,
   211,3,32,210,255
270 DATA 104,74,74,74,74,168,1
   85,211,3,32,210
280 DATA 255,162,31,189,227,3,
   149,199,202,16,248
290 DATA 169,146,32,210,255,76
   ,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
   77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
   116,117,151,128,129,167,136
   ,137
```

How To Type In *COMPUTE!'s Gazette* Programs

Each month, COMPUTE!'s Gazette publishes programs for the Commodore 128, 64, Plus/4, and 16. Each program is clearly marked by title and version. Be sure to type in the correct version for your machine. All 64 programs run on the 128 in 64 mode. Be sure to read the instructions in the corresponding article. This can save time and eliminate any questions which might arise after you begin typing.

We frequently publish two programs designed to make typing easier: The Automatic Proofreader, and MLX, designed for entering machine language programs.

When entering a BASIC program, be especially careful with DATA statements as they are extremely sensitive to errors. A mistyped number in a DATA statement can cause your machine to "lock up" (you'll have no control over the computer). If this happens, the only recourse is to turn your computer off then on, erasing what was in memory. So be sure to *save a program before you run it.* If your computer crashes, you can always reload the program and look for the error.

Special Characters

Most of the programs listed in each issue contain special control characters. To facilitate typing in any programs from the GAZETTE, use the following listing conventions.

The most common type of control characters in our listings appear as words within braces: {DOWN} means to press the cursor down key; {5 SPACES} means to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing another key), the character is underlined. For example, A means hold down the SHIFT key and press A. You may see strange characters on your screen, but that's to be expected. If you find a number followed by an underlined key enclosed in braces (for example, {8 A}), type the key as many times as indicated (in our example, enter eight SHIFTed A's).

If a key is enclosed in special brackets, [ ], hold down the Commodore key (at the lower left corner of the keyboard) and press the indicated character.

Rarely, you'll see a single letter of the alphabet enclosed in braces. This can be entered on the Commodore 64 by pressing the CTRL key while typing the letter in braces. For example, {A} means to press CTRL-A.

The Quote Mode

Although you can move the cursor around the screen with the CRSR keys, often a programmer will want to move the cursor under program control. This is seen in examples such as {LEFT}, and {HOME} in the program listings. The only way the computer can tell the difference between direct and programmed cursor control is *the quote mode.*

Once you press the quote key, you're in quote mode. This mode can be confusing if you mistype a character and cursor left to change it. You'll see a reverse video character (a graphics symbol for cursor left). In this case, you can use the DELete key to back up and edit the line. Type another quote and you're out of quote mode. If things really get confusing, you can exit quote mode simply by pressing RETURN. Then just cursor up to the mistyped line and fix it.

When You Read:	Press:	See:
{CLR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + ↑ CRSR ↓	
{DOWN}	↑ CRSR ↓	
{LEFT}	SHIFT + ← CRSR →	
{RIGHT}	← CRSR →	
{RVS}	CTRL + 9	
{OFF}	CTRL + 0	
{BLK}	CTRL + 1	
{WHT}	CTRL + 2	
{RED}	CTRL + 3	
{CYN}	CTRL + 4	

When You Read:	Press:	See:
{PUR}	CTRL + 5	
{GRN}	CTRL + 6	
{BLU}	CTRL + 7	
{YEL}	CTRL + 8	
{ F1 }	f1	
{ F2 }	SHIFT + f1	
{ F3 }	f3	
{ F4 }	SHIFT + f3	
{ F5 }	f5	
{ F6 }	SHIFT + f5	
{ F7 }	f7	
{ F8 }	SHIFT + f7	

When You Read:	Press:	See:
←	←	
↑	SHIFT + ↑	

For Commodore 64 Only

When You Read:	Press:	See:
[1]	COMMODORE + 1	
[2]	COMMODORE + 2	
[3]	COMMODORE + 3	
[4]	COMMODORE + 4	
[5]	COMMODORE + 5	
[6]	COMMODORE + 6	
[7]	COMMODORE + 7	
[8]	COMMODORE + 8	

UTILITIES UNLIMITED, Inc.

12305 N.E. 152nd Street
Brush Prairie, Washington 98606

Orders taken 24 hrs. a day; seven days a week.

If you wish to place your order by phone, please call **206-254-6530**. Add $3.00 shipping & handling; $2.00 for COD on all orders. Visa, M/C accepted. Dealer Inquiries Invited.

WORLD'S BIGGEST PROVIDER OF C64/128 UTILITIES

Software Submissions Invited

We are looking for HACKER STUFF: print utilities, parameters, telecommunications, and the unusual.

We now have over 1,000 parameters in stock!

Ad copy by RWS Ind

SUPER PARAMETERS 500 PAK

At a time when many programmers are cutting back on writing parameters, we are going full steam! After getting several requests for parameters on many of the "oldies but goodies" **WE DID JUST THAT! ANOTHER 1st: 500 DIFFERENT PARAMETERS** all in one pack. And these parameters are not on any of our regular 100 paks. This **FANTASTIC 5-DISK SET** is a value of over $100 for less than 1/2. **$49.95**

SUPER PARAMETERS 100 PAKS: 1 thru 4 ★

. . . **YOU GET 100 OF THE BEST PARAMETERS, ANYWHERE.** Professionally done, easy to use, reliable, fast and full of performance. This package can remove all protection. Includes titles from all major software publishers for the C64/128. **$19.95@**

SUPER PARAMETERS 10 PAKS: 1 thru 40 ★

These are the early releases for your convenience until we have enough for a new 100 PAK **$4.95@**

ALL PARAMETERS ARE TOP QUALITY—NO FILLER OR JUNK

PARAMETERS CONSTRUCTION SET

The company that has **THE MOST PARAMETERS** is about to do something **UNBELIEVABLE**. We are giving you more of our secrets. Using this **VERY EASY** program, it will not only **READ, COMPARE** and **WRITE PARAMETERS FOR YOU**; it will also **CUSTOMIZE** the disk with your name. It will impress you as well as your friends. The **"PARAMETER CONSTRUCTION SET"** is like nothing you've ever seen. In fact you can even **READ PARAMETERS** that you may have already written; then by using your construction set rewrite it with your new **CUSTOMIZED MENU. DON'T WAIT** for months for that one special parameter: **WRITE IT!** **$34.95**

REMEMBER FOLKS, YOU SAW IT HERE FIRST!

THE FINAL CARTRIDGE III — C-64 and C-128

THE #1 SELLING CARTRIDGE IN EUROPE! After reviewing all utilities of this type, including: Icepick, Capture, and super snap shot, we determined that **THE FINAL CARTRIDGE** is just that: **THE LAST CARTRIDGE THAT YOU WILL EVER NEED. EXTENSIVE** programming utilities for both beginner and the hardcore programmer alike. The most impressive utility for **MEMORY-TO-DISK BACKUP** that we have ever seen. We were able to copy almost every program we tried with just the **PRESS OF THE BUTTON**. Now, you may have seen a comparison of **FINAL CARTRIDGE** to super snap shot by a dealer. It is our opinion that they only printed the few items they felt were in its favor **(NOT ACCURATELY)** and failed to list all **THE FAR SUPERIOR FEATURES OF THE FINAL CARTRIDGE**. Come on guys, let's be fair. To get a much better **UNBIASED OPINION**, see great reviews of **FINAL CARTRIDGE** in:

RUN: Sept. 87, pg. 78-80: *"A conventional review doesn't do The Final Cartridge justice—fun at this price is a rarity."*

GAZ: July 87, pg. 44: *"I can't begin to think of a cartridge which does so many useful things."*

AHOY: Nov. 87, pg. 63: *"Overall we were quite pleased."*

INFO: Aug. 87, pg. 33: *"!!A super hacker's cartridge!!"*

(NOW w/CLOCK)

READ THEM, THEN YOU DECIDE! ONLY $54.95

SIMILAR PRODUCTS MAY APPEAR ELSEWHERE LATER; JUST REMEMBER YOU SAW THEM HERE FIRST!!

SUPER NIBBLERS COMBO! ★

For those who need a really super nibbler copier, for those hard to backup programs, we give you the **★ULTIMATE COPY PROGRAM★** and it works well with both C64 or C128. This package includes both a Super Fast File copier and 100 parameters. **ONLY $24.95**

SUPER PARAMETERS—GEOS PAK

This #1 seller will standardize your **GEOS** series utilities to work with anyone else's **GEOS**. Parameters are included for these: **FONTPACK 1, DESKPACK 1, WRITERS WORKSHOP, GEODEX & GEOFILE. THE ORIGINAL STANDARDIZER** for just **$9.95**

(Another first!) THE 128 SUPERCHIP

There is an empty socket inside your 128 just waiting for our **SUPER CHIP** to give you 32k worth of great **BUILT-IN UTILITIES**, all at just the **TOUCH OF A FINGER**. You get built-in features: **FILE COPIER, NIBBLER, TRACK & SECTOR EDITOR, SCREEN DUMP**, and even a 300/1200 baud **TERMINAL PROGRAM** that's 1650, 1670 and Hayes compatible. Best of all, it doesn't use up any memory. To use, simply touch a function key, and it responds to your command.

ALSO INCLUDES 100 parameters—**FREE!**

All this for **$29.95**

ADULT GAME & GRAPHICS DATA DISKS

GAME: A very unusual game to be played by a very **OPEN MINDED** adult. It includes a **CASINO** and **HOUSE OF ILL REPUTE**. Please, you **MUST BE 18** to order **EITHER ONE**.

DATA ★: This **POPULAR** disk works with **PRINT SHOP** and **PRINT MASTER**.

$24.95 ea.

V-MAX, RAPID LOCK DIGITAL SOLUTIONS™

SUPER SPECIALS—PRICED TO MOVE

THE "ORIGINAL" PARAMETERS CROSS REFERENCE over 4,000 parameters listed complete with sources	$19.95
1541 Mash (for drive alignment)	$14.95
64 BBS: for 1 or 2 drives	$19.95
64 MODEM prg.: w/wargames auto dialer	$14.95
N-CODER & D-CODER: together for only	$14.95
GRAPHIC LABEL MAKER: the original	$12.95
D-COMPILER: for the famous BLITZ!	$14.95
Parameter Kruncher	$9.95
HEAVY DUTY REPAIRABLE 64 POWER SUPPLIES Most expensive C64 repairs are due to a faulty power supply. Don't wait: replace yours now!!	$39.95
CLOSE-OUT OF KRACKER-JAX 64/128 RESET BOARDS	$3.95
DISK NOTCHERS: double disk capacity	$4.95
SECURITY KEYS: lost your dongle? (For golf or bowling games)	$3.95

LOCK PICK—THE BOOKS—for the C64 and C128

Lock Pik 64/128 was put together by our crack team, as a tool for those who have a desire to see the **INTERNAL WORKINGS** of a parameter. The books give you **STEP-BY-STEP INSTRUCTIONS** on breaking protection for backup of 100 popular program titles. Uses **HESMON** and **SUPEREDIT**. Instructions are so clear and precise that anyone can use it. **★OUR BOOK TWO IS NOW AVAILABLE★**

BOOK 1: Includes Hesmon and a disk with many utilities such as: **KERNAL SAVE, I/O SAVE, DISK LOG FILE** and lots more, all with instructions on disk. A long-time favorite of many.

BOOK 2: 100 NEW PARAMETERS. Hesmon on disk, and more utilities to include: **A GENERAL OVERVIEW ON HOW TO MAKE PARAMETERS** and a **DISK SCANNER**.

$29.95 each OR BUY BOTH FOR **ONLY $49.95**

C-64 and C-128 BURST ROM CHIPS

Lets your C-64 run in **"BURST MODE"** when using a 1571 or 1581 disk drive. Loads 100 blks in 6 sec. (1571) and 4 sec. (1581). Features: fast directory, **SEQ.** and **REL.** file access and built-in **DOS WEDGE**. Simple installation Kernal ROM U4 must be socketed. C-128 version has same features as above. For use in 64 mode.

C-64 **$29.95** C-128 **$35.95**

CALL or WRITE FOR FREE CATALOG.

★ = AN IBM and ST VERSION IS ALSO AVAILABLE

MLX Machine Language Entry Program For Commodore 64 and 128

Ottis R. Cowper, Technical Editor

"MLX" is a labor-saving utility that allows almost fail-safe entry of machine language programs. Included are versions for the Commodore 64 and 128.

Type in and save some copies of whichever version of MLX is appropriate for your computer (you'll want to use it to enter future ML programs from COMPUTE!'s GAZETTE). Program 1 is for the Commodore 64, and Program 2 is for the 128 (128 MLX can also be used to enter Commodore 64 ML programs for use in 64 mode). When you're ready to enter an ML program, load and run MLX. It asks you for a starting address and an ending address. These addresses appear in the article accompanying the MLX-format program listing you're typing.

If you're unfamiliar with machine language, the addresses (and all other values you enter in MLX) may appear strange. Instead of the usual decimal numbers you're accustomed to, these numbers are in *hexadecimal*—a base 16 numbering system commonly used by ML programmers. Hexadecimal—hex for short—includes the numerals 0–9 and the letters A–F. But don't worry—even if you know nothing about ML or hex, you should have no trouble using MLX.

After you enter the starting and ending addresses, you'll be offered the option of clearing the workspace. Choose this option if you're starting to enter a new listing. If you're continuing a listing that's partially typed from a previous session, don't choose this option.

A functions menu will appear. The first option in the menu is ENTER DATA. If you're just starting to type in a program, pick this. Press the E key, and type the first number in the first line of the program listing. If you've already typed in part of a program, type the line number where you left off typing at the end of the previous session (be sure to load the partially completed program before you resume entry). In any case, make sure the address you enter corresponds to the address of a line in the listing you are entering. Otherwise, you'll be unable to enter the data correctly. If you pressed E by mistake, you can return to the command menu by pressing RETURN alone when asked for the address. (You can get back to the menu from most options by pressing RETURN with no other input.)

Entering A Listing

Once you're in Enter mode, MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight data bytes and a checksum. Although an MLX-format listing appears similar to the "hex dump" listings from a machine language monitor program, the extra checksum number on the end allows MLX to check your typing. (Commodore 128 users *can* enter the data from an MLX listing using the built-in monitor if the rightmost column of data is omitted, but we recommend against it. It's much easier to let MLX do the proofreading and error checking for you.)

When you enter a line, MLX recalculates the checksum from the eight bytes and the address and compares this value to the number from the ninth column. If the values match, you'll hear a bell tone, the data will be added to the workspace area, and the prompt for the next line of data will appear. But if MLX detects a typing error, you'll hear a low buzz and see an error message. The line will then be redisplayed for editing.

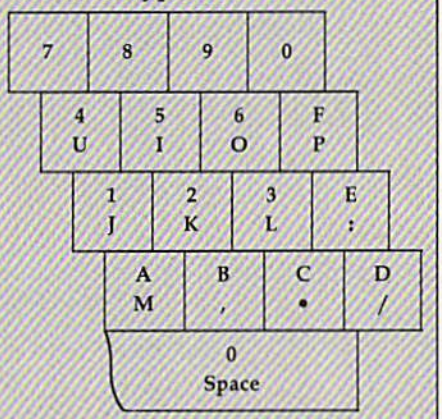
Invalid Characters Banned

Only a few keys are active while you're entering data, so you may have to unlearn some habits. You *do not* type spaces between the columns; MLX automatically inserts these for you. You *do not* press RETURN after typing the last number in a line; MLX automatically enters and checks the line after you type the last digit.

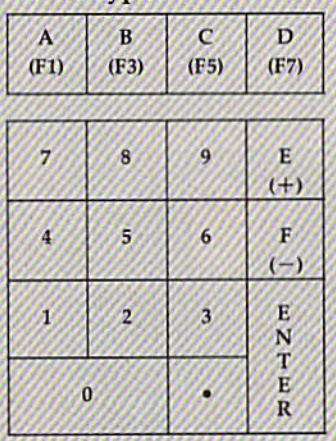
Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. To simplify typing, 128 MLX redefines the function keys and + and − keys on the numeric keypad so that you can enter data one-handed. (The 64 version incorporates the keypad modification from the March 1986 "Bug-Swatter" column, lines 485–487.) In either case, the keypad is active only while entering data. Addresses must be entered with the normal letter and number keys. The figures above show the keypad configurations for each version.

MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

64 MLX Keypad



128 MLX Keypad



Editing Features

To correct typing mistakes before finishing a line, use the INST/DEL key to delete the character to the left of the cursor. (The cursor-left key also deletes.) If you mess up a line really badly, press CLR/HOME to start the line over. The RETURN key is also active, but only before any data is typed on a line. Pressing RETURN at this point returns you to the command menu. After you type a character of data, MLX disables RETURN until the cursor returns to the start of a line. Remember, you can press CLR/HOME to quickly get to a line

number prompt.

More editing features are available when correcting lines in which MLX has detected an error. To make corrections in a line that MLX has redisplayed for editing, compare the line on the screen with the one printed in the listing, then move the cursor to the mistake and type the correct key. The cursor left and right keys provide the normal cursor controls. (The INST/DEL key now works as an alternative cursor-left key.) You cannot move left beyond the first character in the line. If you try to move beyond the rightmost character, you'll reenter the line. During editing, RETURN is active; pressing it tells MLX to recheck the line. You can press the CLR/HOME key to clear the entire line if you want to start from scratch, or if you want to get to a line number prompt to use RETURN to get back to the menu.

Display Data

The second menu choice, DISPLAY DATA, examines memory and shows the contents in the same format as the program listing (including the checksum). When you press D, MLX asks you for a starting address. Be sure that the starting address you give corresponds to a line number in the listing. Otherwise, the checksum display will be meaningless. MLX displays program lines until it reaches the end of the program, at which point the menu is redisplayed. You can pause the display by pressing the space bar. (MLX finishes printing the current line before halting.) Press space again to restart the display. To break out of the display and get back to the menu before the ending address is reached, press RETURN.

Other Menu Options

Two more menu selections let you save programs and load them back into the computer. These are SAVE FILE and LOAD FILE; their operation is quite straightforward. When you press S or L, MLX asks you for the filename. You'll then be asked to press either D or T to select disk or tape.

You'll notice the disk drive starting and stopping several times during a load or save (save only for the 128 version). Don't panic; this is normal behavior. MLX opens and reads from or writes to the file instead of using the usual LOAD and SAVE commands (128 MLX makes use of BLOAD). Disk users should also note that the drive prefix 0: is automatically added to the filename (line 750 in 64 MLX), so this should *not* be included when entering the name. This also precludes the use of @ for Save-with-Replace, so remember to give each version you save a different name. The 128 version makes up for this by giving you the option of scratching the existing file if you want to reuse a filename.

Remember that MLX saves the entire workspace area from the starting address to the ending address, so the save or load may take longer than you might expect if you've entered only a small amount of data from a long listing. When saving a partially completed listing, make sure to note the address where you stopped typing so you'll know where to resume entry when you reload.

MLX reports the standard disk or tape error messages if any problems are detected during the save or load. (Tape users should bear in mind that Commodore computers are never able to detect errors during a save to tape.) MLX also has three special load error messages: INCORRECT STARTING ADDRESS, which means the file you're trying to load does not have the starting address you specified when you ran MLX; LOAD ENDED AT *address*, which means the file you're trying to load ends before the ending address you specified when you started MLX; and TRUNCATED AT ENDING ADDRESS, which means the file you're trying to load extends beyond the ending address you specified when you started MLX. If you see one of these messages and feel certain that you've loaded the right file, exit and rerun MLX, being careful to enter the correct starting and ending addresses.

The 128 version also has a CATALOG DISK option so you can view the contents of the disk directory before saving or loading.

The QUIT menu option has the obvious effect—it stops MLX and enters BASIC. The RUN/STOP key is disabled, so the Q option lets you exit the program without turning off the computer. (Of course, RUN/STOP-RESTORE also gets you out.) You'll be asked for verification; press Y to exit to BASIC, or any other key to return to the menu. After quitting, you can type RUN again and reenter MLX without losing your data, as long as you don't use the clear workspace option.

The Finished Product

When you've finished typing all the data for an ML program and saved your work, you're ready to see the results. The instructions for loading and using the finished product vary from program to program. Some ML programs are designed to be loaded and run like BASIC programs, so all you need to type is LOAD *"filename"*,8 for disk (DLOAD *"filename"* on the 128) or LOAD *"filename"* for tape, and then RUN. Such programs will usually have a starting address of 0801 for the 64 or 1C01 for the 128. Other programs must be reloaded to specific addresses with a command such as LOAD *"filename"*,8,1 for disk (BLOAD *"filename"* on the 128) or LOAD *"filename"*,1,1 for tape, then started with a SYS to a particular memory address. On the Commodore 64, the most common starting address for such programs is 49152, which corresponds to MLX address C000. In either case, you should always refer to the article which accompanies the ML listing for information on loading and running the program.

An Ounce Of Prevention

By the time you finish typing in the data for a long ML program, you may have several hours invested in the project. Don't take chances—use our "Automatic Proofreader" to type the new MLX, and then test your copy *thoroughly* before first using it to enter any significant amount of data. Make sure all the menu options work as they should. Enter fragments of the program starting at several different addresses, then use the Display option to verify that the data has been entered correctly. And be sure to test the Save and Load options several times to ensure that you can recall your work from disk or tape. Don't let a simple typing error in the new MLX cost you several nights of hard work.

Program 1: MLX For Commodore 64

```
SS 10 REM VERSION 1.1: LINES 8
      30,950 MODIFIED, LINES 4
      85-487 ADDED
EK 100 POKE 56,50:CLR:DIM IN$,
      I,J,A,B,A$,B$,A(7),N$
DM 110 C4=48:C6=16:C7=7:Z2=2:Z
      4=254:Z5=255:Z6=256:Z7=
      127
CJ 120 FA=PEEK(45)+Z6*PEEK(46)
      :BS=PEEK(55)+Z6*PEEK(56
      ):H$="0123456789ABCDEF"
SB 130 R$=CHR$(13):L$="{LEFT}"
      :S$=" ":D$=CHR$(20):Z$=
      CHR$(0):T$="{13 RIGHT}"
CQ 140 SD=54272:FOR I=SD TO SD
      +23:POKE I,0:NEXT:POKE
      {SPACE}SD+24,15:POKE 78
      8,52
FC 150 PRINT"{CLR}"CHR$(142)CH
      R$(8):POKE 53280,15:POK
      E 53281,15
EJ 160 PRINT T$" {RED}{RVS}
      {2 SPACES}{8 @}
      {2 SPACES}"SPC(28)"
      {2 SPACES}{OFF}{BLU} ML
      X II {RED}{RVS}
      {2 SPACES}"SPC(28)"
      {12 SPACES}{BLU}"
FR 170 PRINT"{3 DOWN}
      {3 SPACES}COMPUTE!'S MA
      CHINE LANGUAGE EDITOR
      {3 DOWN}"
JB 180 PRINT"{BLK}STARTING ADD
```

```
      RESS[4]";:GOSUB300:SA=A
      D:GOSUB1040:IF F THEN18
      0
GF 190 PRINT"{BLK}{2 SPACES}EN
      DING ADDRESS[4]";:GOSUB
      300:EA=AD:GOSUB1030:IF
      {SPACE}F THEN190
KR 200 INPUT"{3 DOWN}{BLK}CLEA
      R WORKSPACE [Y/N][4]";A
      $:IF LEFT$(A$,1)<>"Y"TH
      EN220
PG 210 PRINT"{2 DOWN}{BLU}WORK
      ING...";:FORI=BS TO BS+
      EA-SA+7:POKE I,0:NEXT:P
      RINT"DONE"
DR 220 PRINTTAB(10)"{2 DOWN}
      {BLK}{RVS} MLX COMMAND
      {SPACE}MENU {DOWN}[4]":
      PRINT T$"{RVS}E{OFF}NTE
      R DATA"
BD 230 PRINT T$"{RVS}D{OFF}ISP
      LAY DATA":PRINT T$"
      {RVS}L{OFF}OAD FILE"
JS 240 PRINT T$"{RVS}S{OFF}AVE
       FILE":PRINT T$"{RVS}Q
      {OFF}UIT{2 DOWN}{BLK}"
JH 250 GET A$:IF A$=N$ THEN250
HK 260 A=0:FOR I=1 TO 5:IF A$=
      MID$("EDLSQ",I,1)THEN A
      =I:I=5
FD 270 NEXT:ON A GOTO420,610,6
      90,700,280:GOSUB1060:GO
      TO250
EJ 280 PRINT"{RVS} QUIT ":INPU
      T"{DOWN}[4]ARE YOU SURE
       [Y/N]";A$:IF LEFT$(A$,
      1)<>"Y"THEN220
EM 290 POKE SD+24,0:END
JX 300 IN$=N$:AD=0:INPUTIN$:IF
      LEN(IN$)<>4THENRETURN
KF 310 B$=IN$:GOSUB320:AD=A:B$
      =MID$(IN$,3):GOSUB320:A
      D=AD*256+A:RETURN
PP 320 A=0:FOR J=1 TO 2:A$=MID
      $(B$,J,1):B=ASC(A$)-C4+
      (A$>"@")*C7:A=A*C6+B
JA 330 IF B<0 OR B>15 THEN AD=
      0:A=-1:J=2
GX 340 NEXT:RETURN
CH 350 B=INT(A/C6):PRINT MID$(
      H$,B+1,1);:B=A-B*C6:PRI
      NT MID$(H$,B+1,1);:RETU
      RN
RR 360 A=INT(AD/Z6):GOSUB350:A
      =AD-A*Z6:GOSUB350:PRINT
      ";";
BE 370 CK=INT(AD/Z6):CK=AD-Z4*
      CK+Z5*(CK>Z7):GOTO390
PX 380 CK=CK*Z2+Z5*(CK>Z7)+A
JC 390 CK=CK+Z5*(CK>Z5):RETURN
QS 400 PRINT"{DOWN}STARTING AT
      [4]";:GOSUB300:IF IN$<>
      N$ THEN GOSUB1030:IF F
      {SPACE}THEN400
EX 410 RETURN
HD 420 PRINT"{RVS} ENTER DATA
      {SPACE}":GOSUB400:IF IN
      $=N$ THEN220
JK 430 OPEN3,3:PRINT
SK 440 POKE198,0:GOSUB360:IF F
       THEN PRINT IN$:PRINT"
      {UP}{5 RIGHT}";
GC 450 FOR I=0 TO 24 STEP 3:B$
      =S$:FOR J=1 TO 2:IF F T
      HEN B$=MID$(IN$,I+J,1)
HA 460 PRINT"{RVS}"B$L$;:IF I<
      24THEN PRINT"{OFF}";
HD 470 GET A$:IF A$=N$ THEN470
FK 480 IF(A$>"/"ANDA$<":")OR(A
      $>"@"ANDA$<"G")THEN540
GS 485 A=-(A$="M")-2*(A$=",")-
      3*(A$=".")-4*(A$="/")-5
      *(A$="J")-6*(A$="K")
FX 486 A=A-7*(A$="L")-8*(A$=":
      ")-9*(A$="U")-10*(A$="I
      ")-11*(A$="O")-12*(A$="
      P")
CM 487 A=A-13*(A$=S$):IF A THE
      N A$=MID$("ABCD123E456F
      0",A,1):GOTO 540
MP 490 IF A$=R$ AND((I=0)AND(J
      =1)OR F)THEN PRINT B$;:
      J=2:NEXT:I=24:GOTO550
KC 500 IF A$="{HOME}" THEN PRI
      NT B$:J=2:NEXT:I=24:NEX
      T:F=0:GOTO440
MX 510 IF(A$="{RIGHT}")ANDF TH
      ENPRINT B$L$;:GOTO540
GK 520 IF A$<>L$ AND A$<>D$ OR
      ((I=0)AND(J=1))THEN GOS
      UB1060:GOTO470
HG 530 A$=L$+S$+L$:PRINT B$L$;
      :J=2-J:IF J THEN PRINT
      {SPACE}L$;:I=I-3
QS 540 PRINT A$;:NEXT J:PRINT
      {SPACE}S$;
PM 550 NEXT I:PRINT:PRINT"{UP}
      {5 RIGHT}";:INPUT#3,IN$
      :IF IN$=N$ THEN CLOSE3:
      GOTO220
QC 560 FOR I=1 TO 25 STEP3:B$=
      MID$(IN$,I):GOSUB320:IF
       I<25 THEN GOSUB380:A(I
      /3)=A
PK 570 NEXT:IF A<>CK THEN GOSU
      B1060:PRINT"{BLK}{RVS}
      {SPACE}ERROR: REENTER L
      INE [4]":F=1:GOTO440
HJ 580 GOSUB1080:B=BS+AD-SA:FO
      R I=0 TO 7:POKE B+I,A(I
      ):NEXT
QQ 590 AD=AD+8:IF AD>EA THEN C
      LOSE3:PRINT"{DOWN}{BLU}
      ** END OF ENTRY **{BLK}
      {2 DOWN}":GOTO700
GQ 600 F=0:GOTO440
QA 610 PRINT"{CLR}{DOWN}{RVS}
      {SPACE}DISPLAY DATA ":G
      OSUB400:IF IN$=N$ THEN2
      20
RJ 620 PRINT"{DOWN}{BLU}PRESS:
       {RVS}SPACE{OFF} TO PAU
      SE, {RVS}RETURN{OFF} TO
       BREAK[4]{DOWN}"
KS 630 GOSUB360:B=BS+AD-SA:FOR
      I=BTO B+7:A=PEEK(I):GOS
      UB350:GOSUB380:PRINT S$
      ;
CC 640 NEXT:PRINT"{RVS}";:A=CK
      :GOSUB350:PRINT
KH 650 F=1:AD=AD+8:IF AD>EA TH
      ENPRINT"{DOWN}{BLU}** E
      ND OF DATA **":GOTO220
KC 660 GET A$:IF A$=R$ THEN GO
      SUB1080:GOTO220
EQ 670 IF A$=S$ THEN F=F+1:GOS
      UB1080
AD 680 ONFGOTO630,660,630
CM 690 PRINT"{DOWN}{RVS} LOAD
      {SPACE}DATA ":OP=1:GOTO
      710
PC 700 PRINT"{DOWN}{RVS} SAVE
      {SPACE}FILE ":OP=0
RX 710 IN$=N$:INPUT"{DOWN}FILE
      NAME[4]";IN$:IF IN$=N$
      {SPACE}THEN220
PR 720 F=0:PRINT"{DOWN}{BLK}
      {RVS}T{OFF}APE OR {RVS}
      D{OFF}ISK: [4]";
FP 730 GET A$:IF A$="T"THEN PR
      INT"T{DOWN}":GOTO880
HQ 740 IF A$<>"D"THEN730
HH 750 PRINT"D{DOWN}":OPEN15,8
      ,15,"I0:":B=EA-SA:IN$="
      0:"+IN$:IF OP THEN810
SQ 760 OPEN 1,8,8,IN$+",P,W":G
      OSUB860:IF A THEN220
FJ 770 AH=INT(SA/256):AL=SA-(A
      H*256):PRINT#1,CHR$(AL)
      ;CHR$(AH);
PE 780 FOR I=0 TO B:PRINT#1,CH
      R$(PEEK(BS+I));:IF ST T
      HEN800
FC 790 NEXT:CLOSE1:CLOSE15:GOT
      O940
GS 800 GOSUB1060:PRINT"{DOWN}
      {BLK}ERROR DURING SAVE:
      [4]":GOSUB860:GOTO220
MA 810 OPEN 1,8,8,IN$+",P,R":G
      OSUB860:IF A THEN220
GE 820 GET#1,A$,B$:AD=ASC(A$+Z
      $)+256*ASC(B$+Z$):IF AD
      <>SA THEN F=1:GOTO850
RX 830 FOR I=0 TO B:GET#1,A$:P
      OKE BS+I,ASC(A$+Z$):IF(
      I<>B)AND ST THEN F=2:AD
      =I:I=B
FA 840 NEXT:IF ST<>64 THEN F=3
FQ 850 CLOSE1:CLOSE15:ON ABS(F
      >0)+1 GOTO960,970
SA 860 INPUT#15,A,A$:IF A THEN
       CLOSE1:CLOSE15:GOSUB10
      60:PRINT"{RVS}ERROR: "A
      $
GQ 870 RETURN
EJ 880 POKE183,PEEK(FA+2):POKE
      187,PEEK(FA+3):POKE188,
      PEEK(FA+4):IFOP=0THEN92
      0
HJ 890 SYS 63466:IF(PEEK(783)A
      ND1)THEN GOSUB1060:PRIN
      T"{DOWN}{RVS} FILE NOT
      {SPACE}FOUND ":GOTO690
CS 900 AD=PEEK(829)+256*PEEK(8
      30):IF AD<>SA THEN F=1:
      GOTO970
SC 910 A=PEEK(831)+256*PEEK(83
      2)-1:F=F-2*(A<EA)-3*(A>
      EA):AD=A-AD:GOTO930
KM 920 A=SA:B=EA+1:GOSUB1010:P
      OKE780,3:SYS 63338
JF 930 A=BS:B=BS+(EA-SA)+1:GOS
      UB1010:ON OP GOTO950:SY
      S 63591
AE 940 GOSUB1080:PRINT"{BLU}**
       SAVE COMPLETED **":GOT
      O220
XP 950 POKE147,0:SYS 63562:IF
      {SPACE}ST>0 THEN970
FR 960 GOSUB1080:PRINT"{BLU}**
       LOAD COMPLETED **":GOT
      O220
DP 970 GOSUB1060:PRINT"{BLK}
      {RVS}ERROR DURING LOAD:
      {DOWN}[4]":ON F GOSUB98
      0,990,1000:GOTO220
PP 980 PRINT"INCORRECT STARTIN
      G ADDRESS (";:GOSUB360:
      PRINT")":RETURN
GR 990 PRINT"LOAD ENDED AT ";:
      AD=SA+AD:GOSUB360:PRINT
       D$:RETURN
FD 1000 PRINT"TRUNCATED AT END
      ING ADDRESS":RETURN
RX 1010 AH=INT(A/256):AL=A-(AH
      *256):POKE193,AL:POKE1
      94,AH
FF 1020 AH=INT(B/256):AL=B-(AH
      *256):POKE174,AL:POKE1
      75,AH:RETURN
FX 1030 IF AD<SA OR AD>EA THEN
      1050
HA 1040 IF(AD>511 AND AD<40960
```

```
)OR(AD>49151 AND AD<53
248)THEN GOSUB1080:F=0
:RETURN
HC 1050 GOSUB1060:PRINT"{RVS}
{SPACE}INVALID ADDRESS
{DOWN}{BLK}":F=1:RETU
RN
AR 1060 POKE SD+5,31:POKE SD+6
,208:POKE SD,240:POKE
{SPACE}SD+1,4:POKE SD+
4,33
DX 1070 FOR S=1 TO 100:NEXT:GO
TO1090
PF 1080 POKE SD+5,8:POKE SD+6,
240:POKE SD,0:POKE SD+
1,90:POKE SD+4,17
AC 1090 FOR S=1 TO 100:NEXT:PO
KE SD+4,0:POKE SD,0:PO
KE SD+1,0:RETURN
```

Program 2: MLX For Commodore 128

```
AE 100 TRAP 960:POKE 4627,128:
DIM NL$,A(7)
XP 110 Z2=2:Z4=254:Z5=255:Z6=2
56:Z7=127:BS=256*PEEK(4
627):EA=65280
FB 120 BE$=CHR$(7):RT$=CHR$(13
):DL$=CHR$(20):SP$=CHR$
(32):LF$=CHR$(157)
KE 130 DEF FNHB(A)=INT(A/256):
DEF FNLB(A)=A-FNHB(A)*2
56:DEF FNAD(A)=PEEK(A)+
256*PEEK(A+1)
JB 140 KEY 1,"A":KEY 3,"B":KEY
5,"C":KEY 7,"D":VOL 15
:IF RGR(0)=5 THEN FAST
FJ 150 PRINT"{CLR}"CHR$(142);C
HR$(8):COLOR 0,15:COLOR
4,15:COLOR 6,15
GQ 160 PRINT TAB(12)"{RED}
{RVS}{2 SPACES}{9 @}
{2 SPACES}"RT$;TAB(12)"
{RVS}{2 SPACES}{OFF}
{BLU} 128 MLX {RED}
{RVS}{2 SPACES}"RT$;TAB
(12)"{RVS}{13 SPACES}
{BLU}"
FE 170 PRINT"{2 DOWN}
{3 SPACES}COMPUTE!'S MA
CHINE LANGUAGE EDITOR
{2 DOWN}"
DK 180 PRINT"{BLK}STARTING ADD
RESS{4}";:GOSUB 260:IF
{SPACE}AD THEN SA=AD:EL
SE 180
FH 190 PRINT"{BLK}{2 SPACES}EN
DING ADDRESS{4}";:GOSUB
260:IF AD THEN EA=AD:E
LSE 190
MF 200 PRINT"{DOWN}{BLK}CLEAR
{SPACE}WORKSPACE [Y/N]?
{4}":GETKEY A$:IF A$<>"
Y" THEN 220
QH 210 PRINT"{DOWN}{BLU}WORKIN
G...";:BANK 0:FOR A=BS
{SPACE}TO BS+(EA-SA)+7:
POKE A,0:NEXT A:PRINT"D
ONE"
DC 220 PRINT TAB(10)"{DOWN}
{BLK}{RVS} MLX COMMAND
{SPACE}MENU {4}{DOWN}":
PRINT TAB(13)"{RVS}E
{OFF}NTER DATA"RT$;TAB(
13)"{RVS}D{OFF}ISPLAY D
ATA"RT$;TAB(13)"{RVS}L
{OFF}OAD FILE"
HB 230 PRINT TAB(13)"{RVS}S
{OFF}AVE FILE"RT$;TAB(1
3)"{RVS}C{OFF}ATALOG DI
SK"RT$;TAB(13)"{RVS}Q
{OFF}UIT{DOWN}{BLK}"
AP 240 GETKEY A$:A=INSTR("EDLS
CQ",A$):ON A GOTO 340,5
50,640,650,930,940:GOSU
B 950:GOTO 240
SX 250 PRINT"STARTING AT";:GOS
UB 260:IF(AD<>0)OR(A$=N
L$)THEN RETURN:ELSE 250
BG 260 A$=NL$:INPUT A$:IF LEN(
A$)=4 THEN AD=DEC(A$)
PP 270 IF AD=0 THEN BEGIN:IF A
$<>NL$ THEN 300:ELSE RE
TURN:BEND
MA 280 IF AD<SA OR AD>EA THEN
{SPACE}300
PM 290 IF AD>511 AND AD<65280
{SPACE}THEN PRINT BE$;:
RETURN
SQ 300 GOSUB 950:PRINT"{RVS} I
NVALID ADDRESS {DOWN}
{BLK}":AD=0:RETURN
RD 310 CK=FNHB(AD):CK=AD-Z4*CK
+Z5*(CK>Z7):GOTO 330
DD 320 CK=CK*Z2+Z5*(CK>Z7)+A
AH 330 CK=CK+Z5*(CK>Z5):RETURN
QD 340 PRINT BE$;"{RVS} ENTER
{SPACE}DATA ":GOSUB 250
:IF A$=NL$ THEN 220
JA 350 BANK 0:PRINT:F=0:OPEN 3
,3
BR 360 GOSUB 310:PRINT HEX$(AD
)+":";:IF F THEN PRINT
{SPACE}L$:PRINT"{UP}
{5 RIGHT}";
QA 370 FOR I=0 TO 24 STEP 3:B$
=SP$:FOR J=1 TO 2:IF F
{SPACE}THEN B$=MID$(L$,
I+J,1)
PS 380 PRINT"{RVS}"B$+LF$;:IF
{SPACE}I<24 THEN PRINT"
{OFF}";
RC 390 GETKEY A$:IF (A$>"/" AN
D A$<":") OR(A$>"@" AND
A$<"G") THEN 470
AC 400 IF A$="+" THEN A$="E":G
OTO 470
QB 410 IF A$="-" THEN A$="F":G
OTO 470
FB 420 IF A$=RT$ AND ((I=0) AN
D (J=1) OR F) THEN PRIN
T B$;:J=2:NEXT:I=24:GOT
O 480
RD 430 IF A$="{HOME}" THEN PRI
NT B$:J=2:NEXT:I=24:NEX
T:F=0:GOTO 360
XB 440 IF (A$="{RIGHT}") AND F
THEN PRINT B$+LF$;:GOT
O 470
JP 450 IF A$<>LF$ AND A$<>DL$
{SPACE}OR ((I=0) AND (J
=1)) THEN GOSUB 950:GOT
O 390
PS 460 A$=LF$+SP$+LF$:PRINT B$
+LF$;:J=2-J:IF J THEN P
RINT LF$;:I=I-3
GB 470 PRINT A$;:NEXT J:PRINT
{SPACE}SP$;
HA 480 NEXT I:PRINT:PRINT"{UP}
{5 RIGHT}";:L$="
{27 SPACES}"
DP 490 FOR I=1 TO 25 STEP 3:GE
T#3,A$,B$:IF A$=SP$ THE
N I=25:NEXT:CLOSE 3:GOT
O 220
BA 500 A$=A$+B$:A=DEC(A$):MID$
(L$,I,2)=A$:IF I<25 THE
N GOSUB 320:A(I/3)=A:GE
T#3,A$
AR 510 NEXT I:IF A<>CK THEN GO
SUB 950:PRINT:PRINT"
{RVS} ERROR: REENTER LI
NE ":F=1:GOTO 360
DX 520 PRINT BE$;B=BS+AD-SA:FO
R I=0 TO 7:POKE B+I,A(I
):NEXT I
XB 530 F=0:AD=AD+8:IF AD<=EA T
HEN 360
CA 540 CLOSE 3:PRINT"{DOWN}
{BLU}** END OF ENTRY **
{BLK}{2 DOWN}":GOTO 650
MC 550 PRINT BE$;"{CLR}{DOWN}
{RVS} DISPLAY DATA ":GO
SUB 250:IF A$=NL$ THEN
{SPACE}220
JF 560 BANK 0:PRINT"{DOWN}
{BLU}PRESS: {RVS}SPACE
{OFF} TO PAUSE, {RVS}RE
TURN{OFF} TO BREAK{4}
{DOWN}"
XA 570 PRINT HEX$(AD)+":";:GOS
UB 310:B=BS+AD-SA
DJ 580 FOR I=B TO B+7:A=PEEK(I
):PRINT RIGHT$(HEX$(A),
2);SP$;:GOSUB 320:NEXT
{SPACE}I
XB 590 PRINT"{RVS}";RIGHT$(HEX
$(CK),2)
GR 600 F=1:AD=AD+8:IF AD>EA TH
EN PRINT"{BLU}** END OF
DATA **":GOTO 220
EB 610 GET A$:IF A$=RT$ THEN P
RINT BE$:GOTO 220
QK 620 IF A$=SP$ THEN F=F+1:PR
INT BE$;
XS 630 ON F GOTO 570,610,570
RF 640 PRINT BE$"{DOWN}{RVS} L
OAD DATA ":OP=1:GOTO 66
0
BP 650 PRINT BE$"{DOWN}{RVS} S
AVE FILE ":OP=0
DM 660 F=0:F$=NL$:INPUT"FILENA
ME{4}";F$:IF F$=NL$ THE
N 220
RF 670 PRINT"{DOWN}{BLK}{RVS}T
{OFF}APE OR {RVS}D{OFF}
ISK: {4}";
SQ 680 GETKEY A$:IF A$="T" THE
N 850:ELSE IF A$<>"D" T
HEN 680
SP 690 PRINT"DISK{DOWN}":IF OP
THEN 760
EG 700 DOPEN#1,(F$+",P"),W:IF
{SPACE}DS THEN A$=DS$:G
OTO 740
JH 710 BANK 0:POKE BS-2,FNLB(S
A):POKE BS-1,FNHB(SA):P
RINT"SAVING ";F$:PRINT
MC 720 FOR A=BS-2 TO BS+EA-SA:
PRINT#1,CHR$(PEEK(A));:
IF ST THEN A$="DISK WRI
TE ERROR":GOTO 750
GC 730 NEXT A:CLOSE 1:PRINT"
{BLU}** SAVE COMPLETED
{SPACE}WITHOUT ERRORS *
*":GOTO 220
RA 740 IF DS=63 THEN BEGIN:CLO
SE 1:INPUT"{BLK}REPLACE
EXISTING FILE [Y/N]{4}
";A$:IF A$="Y" THEN SCR
ATCH(F$):PRINT:GOTO 700
:ELSE PRINT"{BLK}":GOTO
660:BEND
GA 750 CLOSE 1:GOSUB 950:PRINT
"{BLK}{RVS} ERROR DURIN
G SAVE: {4}":PRINT A$:G
OTO 220
FD 760 DOPEN#1,(F$+",P"):IF DS
THEN A$=DS$:F=4:CLOSE
{SPACE}1:GOTO 790
```

```
PX 770 GET#1,A$,B$:CLOSE 1:AD=
       ASC(A$)+256*ASC(B$):IF
       {SPACE}AD<>SA THEN F=1:
       GOTO 790
KB 780 PRINT"LOADING ";F$:PRIN
       T:BLOAD(F$),B0,P(BS):AD
       =SA+FNAD(174)-BS-1:F=-2
       *(AD<EA)-3*(AD>EA)
RQ 790 IF F THEN 800:ELSE PRIN
       T"{BLU}** LOAD COMPLETE
       D WITHOUT ERRORS **":GO
       TO 220
ER 800 GOSUB 950:PRINT"{BLK}
       {RVS} ERROR DURING LOAD
       : {4}":ON F GOSUB 810,8
       20,830,840:GOTO220
QJ 810 PRINT"INCORRECT STARTIN
       G ADDRESS (";HEX$(AD);"
       )":RETURN
DP 820 PRINT"LOAD ENDED AT ";H
       EX$(AD):RETURN
EB 830 PRINT"TRUNCATED AT ENDI
       NG ADDRESS ("HEX$(EA)")
       ":RETURN
FP 840 PRINT"DISK ERROR ";A$:R
       ETURN
KS 850 PRINT"TAPE":AD=POINTER(
       F$):BANK 1:A=PEEK(AD):A
       L=PEEK(AD+1):AH=PEEK(AD
       +2)
XX 860 BANK 15:SYS DEC("FF68")
       ,0,1:SYS DEC("FFBA"),1,
       1,0:SYS DEC("FFBD"),A,A
       L,AH:SYS DEC("FF90"),12
       8:IF OP THEN 890
FG 870 PRINT:A=SA:B=EA+1:GOSUB
        920:SYS DEC("E919"),3:
       PRINT"SAVING ";F$
AB 880 A=BS:B=BS+(EA-SA)+1:GOS
       UB 920:SYS DEC("EA18"):
       PRINT"{DOWN}{BLU}** TAP
       E SAVE COMPLETED **":GO
       TO 220
CP 890 SYS DEC("E99A"):PRINT:I
       F PEEK(2816)=5 THEN GOS
       UB 950:PRINT"{DOWN}
       {BLK}{RVS} FILE NOT FOU
       ND ":GOTO 220
GQ 900 PRINT"LOADING ...{DOWN}
       ":AD=FNAD(2817):IF AD<>
       SA THEN F=1:GOTO 800:EL
       SE AD=FNAD(2819)-1:F=-2
       *(AD<EA)-3*(AD>EA)
JD 910 A=BS:B=BS+(EA-SA)+1:GOS
       UB 920:SYS DEC("E9FB"):
       IF ST>0 THEN 800:ELSE 7
       90
XB 920 POKE193,FNLB(A):POKE194
       ,FNHB(A):POKE 174,FNLB(
       B):POKE 175,FNHB(B):RET
       URN
CP 930 CATALOG:PRINT"{DOWN}
       {BLU}** PRESS ANY KEY F
       OR MENU **":GETKEY A$:G
       OTO 220
MM 940 PRINT BE$"{RVS} QUIT
       {4}";RT$;"ARE YOU SURE
       {SPACE}[Y/N]?":GETKEY A
       $:IF A$<>"Y" THEN 220:E
       LSE PRINT"{CLR}":BANK 1
       5:END
JE 950 SOUND 1,500,10:RETURN
AF 960 IF ER=14 AND EL=260 THE
       N RESUME 300
MK 970 IF ER=14 AND EL=500 THE
       N RESUME NEXT
KJ 980 IF ER=4 AND EL=780 THEN
        F=4:A$=DS$:RESUME 800
DQ 990 IF ER=30 THEN RESUME:EL
       SE PRINT ERR$(ER);" ERR
       OR IN LINE";EL
```

SOAR to new heights with flight simulation books from COMPUTE!

This collection of flight guidebooks helps you get the most out of flight simulation software. Designed to be used with Flight Simulator, Flight Simulator II, F-15 Strike Eagle, and other fighter-jet simulations, these books contain scores of scenarios that lead you across the U.S., around the world, into battle, and back to your favorite landing strips.

Flying on Instruments with *Flight Simulator*
Jonathan M. Stern 251 pages
Here's fascinating hands-on instruction on flying with instruments in *Flight Simulator* and *Flight Simulator II*. How to fly in all kinds of weather—when the cloud ceiling is low or the visibility is reduced. How to read and fly the instrument approach charts that instrument-rated pilots use. How to take off, fly to another airport, and land—all while the clouds are a mere 200 feet off the ground. And how to navigate by using the impressive instruments available in *Flight Simulator* and *Flight Simulator II*. More than 130 actual instrument approach procedure charts. For all versions of *Flight Simulator* and *Flight Simulator II* (Apple, Amiga, Atari, Atari ST, Commodore, IBM PC and compatibles, Macintosh). *Flight Simulator* or *Flight Simulator II* required.
Perfect-bound edition:
$9.95 ISBN 0-87455-091-2
Split-wire-bound edition:
$12.95 ISBN 0-87455-103-X

COMPUTE!'s Flight Simulator Adventures for the Macintosh, Amiga, and Atari ST
David Florance, Tom R. Halfhill, and Philip I. Nelson 227 pages
Fly through mists into mystery. This collection of 48 exciting, new, customized simulator scenarios for the Macintosh, Commodore Amiga, and Atari ST personal computers puts you in the pilot's seat. The new generation of *Flight Simulator* and *Flight Simulator II* offers better graphics, easier-to-use controls, and more airports. With this book, you'll experience hair-raising rescue missions, dangerous night landings, sightseeing side trips, and one-of-a-kind flights of fancy. Make *COMPUTE!'s Flight Simulator Adventures* your copilot and thrill to hours of flight.
Flight Simulator or *Flight Simulator II* required.
$12.95 ISBN 0-87455-100-5

Learning to Fly with *Flight Simulator*
John Rafferty
Learn to fly like a simulator expert in the imaginary world of *Flight Simulator* and *Flight Simulator II*. You'll learn all the fundamentals—from takeoffs to landings—in a single, detailed flight. Then you'll develop those flying skills by practicing standard rate turns, executing a flight plan, navigating from point A to point B, and more. Learn how to make precision landings and use the more advanced instruments in your aircraft. To cap your training—and turn you into a professional simulator pilot—you'll learn how to read and use the actual FAA approach charts commercial pilots fly with. This complete flight-instruction tutorial was written by a professional pilot.
$12.95 ISBN 0-87455-115-3

40 Great Flight Simulator Adventures
Charles Gulick 152 pages
This bestselling volume offers 40 exciting, customized flight simulator scenarios for *Flight Simulator* and *Flight Simulator II* on the Apple II, IBM, Commodore 64, and Atari personal computers.
$10.95 ISBN 0-87455-022-X

40 More Great Flight Simulator Adventures
Charles Gulick 236 pages
Forty more challenging flight simulator scenarios for Microsoft's *Flight Simulator* and SubLOGIC's *Flight Simulator II*. Explore the world from the air. For the Apple II, IBM, Commodore 64, and Atari personal computers.
$12.95 ISBN 0-87455-043-2

Jet Fighter School: Air Combat Simulator Tactics and Maneuvers
Richard G. Sheffield 177 pages
Learn actual combat maneuvers and much more with this hands-on book for popular jet-combat simulators like *F-15 Strike Eagle, Jet,* and *Ace*. Information about jet fighter pilots and their aircraft, especially the F-15, provides an excellent introduction to the subject. There's even detailed data about the weapons available to jet-combat simulator pilots. The heart of the book demonstrates, step-by-step, numerous offensive and defensive combat flight maneuvers, from the Hi Yo-Yo to the Defensive Spiral Drive. Tactical tips on *F-15 Strike Eagle* scenarios round out the book. *F-15 Strike Eagle* or other jet-combat flight simulator required.
Perfect-bound edition:
$9.95 ISBN 0-87455-092-0
Split-wire-bound edition:
$12.95 ISBN 0-87455-104-8

Ask for COMPUTE! Books at your local book or computer store.
To order direct, call toll free **800-346-6767** (in N.Y. call 212-887-8525) or write **COMPUTE! Books**, P.O. Box 5038, F.D.R. Station, New York, NY 10019.
Customer Service hours are 10:00 am–12:30 pm and 1:30 pm–3:00 pm EST.

Please include $2.00 shipping and handling in the U.S. or $5.00 airmail. NC residents add 5 percent sales tax and NY residents add 8.25 percent sales tax. Please allow 4–6 weeks for delivery.

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company

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

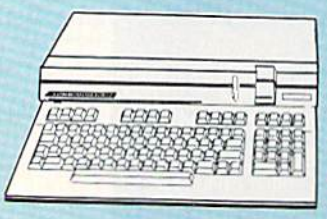
Lyco Computer

Marketing & Consultants

Since 1981

Order processed within 24 hours.

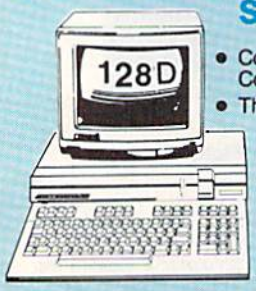
COMMODORE 128D System



Commodore 128D Plus Thomson 4120 Monitor $659.95

$439.95

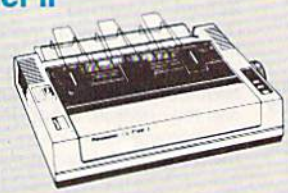
COMMODORE 128D System Special



- Commodore 128D Computer/Drive
- Thompson 4120 Monitor

$649.95

Panasonic 1091 Model II



$179.95

Quantities Limited

Smarteam 1200 Baud Modem

$89.95

Hayes Compatible

COMMODORE HARDWARE

1541 C Disk Drive	$164.95
1571 Disk Drive	$215.95
1581 Disk Drive	$LOWEST
Indus GT C-64 Drive	$169.95
Blue Chip 64C Drive	$139.95
Blue Chip C 128 Drive	$195.99
1802C Monitor	$189.95
2002 Monitor	$239.95
1901 Monitor	$CALL
64 C Computer	$LOW
128D Computer/Drive	$439.95
1670 Modem	$CALL
C-1351 Mouse	$32.95
1700 RAM	$109.95
C-1750 RAM	$169.95
1764 RAM C64	$124.95

Attention Educational Institutions:

If you are not currently using our educational service program, please call our representatives for details.

PC COMPATIBLE HARDWARE

PC 5¼ Drive	$119
MS DOS + Basic Soft	$59
BCM 12G Gr. Monitor	$89
BCM 12A Am. Monitor	$89
BCM 14C Color RGB Monitor	$269
BCC CG Color Card	$94.99
BCPC PC/XT Comp	$559
Avatex Graphics Card	$169
Zuckerboard	$CALL
Iomega Bernoulli	$CALL
Hercules	$CALL
AST	$CALL

Seagate HARD DRIVES

20 meg kit	$275.95*
30 meg kit	$299.95*
40 meg kit	$489.95*
*Drives include controller	
40 meg w/o controller	$425.95

COMMODORE 64C COMPUTER

~~$299.95~~

Call For December Special!

Priced too low to print.

COMMODORE 64C System

$459.95

- Commodore 64 C Computer
- Commodore 1541 C Drive
- Star NP 10 Printer

BLUE CHIP PERSONAL COMPUTER

- 100% IBM PC/XT compatible!

$479

(Green, amber & color monitors available)

BLUE CHIP special

- Blue Chip Personal Computer
- Seagate 20 Meg Hard Drive

$749.95

(Hard drive must be installed)

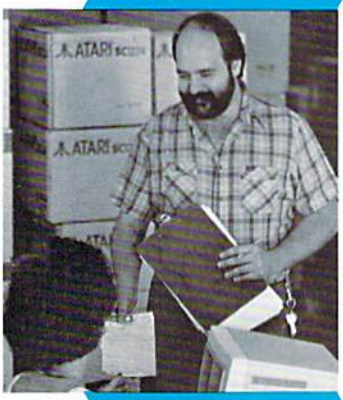
1-800-233-8760

Lyco Means Total Service.

Mark "Mac" Bowser, Sales Manager

I would personally like to thank all of our past customers for helping to make Lyco Computer one of the largest mail order companies and a leader in the industry. Also, I would like to extend my personal invitation to all computer enthusiasts who have not experienced the services that we provide. Please call our trained sales staff at our toll free number to inquire about our diverse product line and weekly specials.

Call Lyco

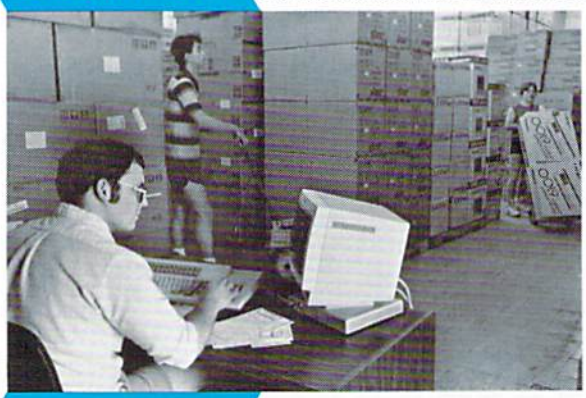


First and foremost our philosophy is to keep abreast of the changing market so that we can provide you with not only factory-fresh merchandise but also the newest models offered by the manufacturers at the absolute best possible prices. And we offer the widest selection of computer hardware, software and accessories.

Feel free to call Lyco if you want to know more about a particular item. I can't stress enough that our toll-free number is not just for orders. Many companies have a toll-free number for ordering, but if you just want to ask a question about a product, you have to make a toll call. Not at Lyco. Our trained sales staff is knowledgeable about all the products we stock and is happy to answer any questions you may have. We will do our best to make sure that the product you select will fit your application. We also have Saturday hours — one more reason to call us for all your computer needs.

Order Now

Once you've placed your order with Lyco, we don't forget about you. Our friendly, professional customer service representatives will find answers to your questions about the status of an order, warranties, product availability, or prices.



Lyco Computer stocks a multimillion dollar inventory of factory-fresh merchandise. Chances are we have exactly what you want right in our warehouse. And that means you'll get it fast. In fact, orders are normally shipped within 24 hours. Free shipping on prepaid cash orders over $50, and there is no deposit required on C.O.D. orders. Air freight or UPS Blue/Red Label shipping is available, too. And all products carry the full manufacturers' warranties.

Here's How

I can't see why anyone would shop anywhere else. Selection from our huge in-stock inventory, best price, service that can't be beat — we've got it all here at Lyco Computer.

TO ORDER, CALL TOLL-FREE: 1-800-233-8760
New PA Wats: 1-800-233-8760
Outside Continental US Call: **1-717-494-1030**

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

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

Risk-Free Policy: • full manufacturers' warranties • no sales tax outside PA • prices show 4% cash discount; add 4% for credit cards • APO, FPO, international: add $5 plus 3% for priority • 4-week clearance on personal checks • we check for credit card theft • sorry, compatability not guaranteed • return authorization required • price/availability subject to change • prepaid orders under $50 in Continental US, add $3.00

MasterCard VISA

New Item!

Computer Area Light Center
- Heavy-duty weighted base for extra stability
- Built-in storage for disks, pens, and paper
- Two-hinged desktop arm for optimal positioning

Introductory price: $29.95

(includes delivery)

Disc Storage

QVS-10 5¼	$3.99
QVS-75 5¼	$16.95
QVS-40 3½	$11.95

Great News

Peak Peripherals RS232 Modem adapter for C64/128
- Adapts std DB25 Modem to your Commodore computer

Only $19.95

Printer Interfaces

Xetec Jr	$38.95
Xetec Supergraphics	$58.95
PPI	$45.95
Cardco GWhiz	$35.95
Cardco Super G	$45.95

Special Purchase

Data Share-Messenger Modem
300 Baud Auto Dial/Auto Answer Modem

Only $29.95

Great Bargain

Contriver M-1 Mouse
- Works with Geos
- Complete with Graphic Utility Software

Only $32.95

Mouse Pad

- provides clean, dust-free surface for your mouse
- No-mar backing to protect your desk surface

9x11 .... $10.95

ACCESSORIES — SOFTWARE

Diskettes

1/4

xell:
DD $8.99
DD $9.99

hus:
DD $6.45
DD $6.95

C:
DD $7.99
HD $16.95
heric DSDD $6.99

rbatim:
DD $9.95
DD $12.95

5

xell:
DD $12.50
DD $17.99

rbatim:
DD $12.99
DD $18.99

C:
DD $11.99
DD $14.99
heric SSDD $13.95
heric DSDD $15.95

Free
Disk Storage Case
w/purchase of 5 boxes

Joysticks

: 3 ..... $9.95
: 2 ..... $10.95
: 5 ..... $12.95
: 1 + IBM/AP $26.95
inomy $5.95
Stick $6.95
ck Max $10.95
is $11.99
Vay $19.99
handle $16.75
IXJ $13.95
IXJ-Apple/PC $28.95
ner 909 ..... $24.95
o IBM/AP $29.95
ntriver Joystick Bd .. $34.95

Dustcovers

1081 ..... $9.95
1091 ..... $9.95
on FX286 ..... $14.95
on FX86 ..... $9.95
on LX86 ..... $9.95
r NX10 ..... $9.95
r SR10 ..... $9.95
r SR15 ..... $14.95
-182-192 ..... $9.95
kosha SP series $9.95
Color $14.95
28 ..... $9.95
1 ..... $9.95
ga $14.95

Printer Paper

sheet laser $9.95
0 sheet laser $19.95
sheet ivory, 20lb $9.95

Printer Paper

150 sheet white, 20lb $9.95
1000 sh. Gr. bar 14" ... $29.95
1000 mailing labels $9.95

Surge Suppressors

QVS Surge 6-outlet $24.95
QVS Surge 6-outlet with EMI/RFI $32.95
QVS Surge 6-outlet with indicator $29.95
QVS 6-out. Pow. Strip . $14.95

Drive Maintenance

Nortronics 5¼ Drive Cleaner $9.95
Nortronics 5¼ Drive Cleaner Kit $17.95

Switch Boxes

Cent '25' AB $49.95
Cent '36' AB $49.95
RS232 ABC $52.95
Cent ABC $54.95
RS232 ABCD $59.95
Cent ABCD $59.95

Color Ribbons Available

Video Tape

SKC T120 VHS Video Tape:
each $4.49
3 pack $12.75
10 pack $39.90

Closeouts!

Factory Fresh Special Pricing Call for Availability

Quantum Link ..... $SAVE
Leading Edge ..... $SAVE
Computer Cover-up (dustcovers) ..... $SAVE
Novation modems ..... $SAVE
Anchor modems ..... $SAVE
Certron ..... $SAVE
Toshiba VCR's ..... $SAVE
Teknika Monitors ..... $SAVE
Panasonic Monitors ..... $SAVE
Dennison Diskettes ..... $SAVE

Great savings on Epyx, Cardco, Eastern House, High Tech Expressions, Timeworks, Spinaker, Scarborough, Continental, and much more!!

Generic Ribbons

Save up to 50% off brand prices! Call for your make and model!

COMMODORE

Access:
Echelon $25.95
Leader Board $22.95
Mach 5 ..... $19.95
Mach – 128 ..... $28.95
10th Frame $22.95
Triple Pack $11.95
Wld. Cl. Leader Brd. ... $22.95
Famous Courses #1 ... $11.95
Famous Courses #2 ... $11.95

Action Soft:
Up Periscope $18.95
Thunder Chopper ..... $NEW

Activision:
Aliens $19.95
Champion. Basketball . $19.95
Hacker $8.95
Hacker 2 ..... $19.95
Labyrinth $19.95
Music Studio $19.95
Tass Times $16.95
Titanic $11.95
Leather Goddesses $22.95
Stationfall $19.95
Lurking Horror $19.95
Top Fuel Eliminator $15.95

Berkeley Softworks:
Geos $35.95
Deskpak I $20.95
Fontpak I $17.95
Geodex $23.95
Geofile $29.95
Writers Workshop $29.95
Geo Calc $29.95

Broderbund:
Bank St. Writer $27.95
Carmen San Diego $19.95
Graphic Lib. I, II, III $13.95
Karateka $12.95
Print Shop $25.49
Print Shop Compan. $22.95
Print Shop Paper $12.95

Epyx:
Create A Calendar $15.95
Destroyer $22.95
Fastload $19.95
Football $11.95
Movie Monster $11.95
Sub Battle $22.95
Winter Games $22.95
California Games $22.95
Super Cycle $11.95
Graphics Scrapbook $11.95
Str. Sports Basketball . $22.95
Wld.'s Great Baseball . $11.95
Summer Games II $22.95
Vorpol Utility Kit $11.95
World Games $22.95

Firebird:
Colossus Chess IV $19.95
Elite $18.95
Golden Path $22.95
Guild of Thieves $22.95
Pawn $22.95
Talking Teacher $22.95
Tracker $22.95
Starglider $22.95
Sentry $22.95

Microleague:
Microleag. Baseball $22.95
General Manager $16.95
Stat Disk $13.95
'86 Team Disk $11.95
Microleag. Wrestling ..... $New

Microprose:
F-15 Strike Eagle $19.95
Gunship $19.95
Kennedy Approach $13.95
Silent Service $19.95

COMMODORE

Solo Flight $13.95
Top Gunner $13.95

Origin:
Autoduel $28.95
Ultima III $22.95
Ultima IV $33.95
Moebius $22.95

Springboard:
Newsroom $29.95
Certificate Maker $29.95
Clip Art Vol. #1 ..... $17.95
Clip Art Vol. #2 ..... $23.95
Clip Art Vol. #3 ..... $17.95
Graphics Expander $21.95

Strategic Simulations:
Gemstone Healer $16.95
Gettysburg $33.95
Kampfgruppe $33.95
Phantasie II $22.95
Phantasie III $22.95
Ring of Zelfin $22.95
Road War 2000 ..... $22.95
Shard of Spring $22.95
Wizards Crown $22.95
War in the S. Pacific ... $33.95
Wargame Constr. $16.95
Battlecruiser $33.95
Gemstone Warrior $7.95
Battle of Antetiem $28.95
Colonial Conquest $22.95
Computer Ambush $33.95
B-24 ..... $19.95

Sublogic:
Flight Simulator II $31.49
Jet Simulator $24.95
Night Mission Pinball ... $18.95
Scenery Disk 1-6 ..... $12.95

Timeworks:
Partner C64 ..... $24.95
Partner 128 ..... $39.95
Swift Calc 128 ..... $29.95
Wordwriter 128 ..... $29.95

Unison World:
Art Gallery 1 or 2 ..... $14.95
Print Master $17.95

AMIGA

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

Activision:
Champ. Basketball $27.95
Championship Golf $24.95
GFL Football $27.95
Hacker $16.95
Hacker 2 ..... $24.95
Music Studio $29.95
Tass Times $24.95

Epyx:
Apshai Trilogy $13.95
Rogue $22.95
Winter Games $22.95
World Games $22.95

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

Microprose:
Silent Service $24.95

Sublogic:
Flight Simulator II $32.95
Jet Simulator $32.95
Scenery Disk $17.95

AMIGA

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

IBM

Access:
Wld. Cl. Leader Board . $27.95
10th Frame $27.95

Activision:
Pebble Beach Golf $22.95
Champ. Baseball $22.95
Champ. Basketball $22.95
Zork Trilogy $39.95
Leather Goddesses $22.95
Moonmist $22.95

Broderbund:
Ancient Art of War $25.95
Print Shop $32.95
Print Shop Compan. $31.95
Graphic Lib. I or II $19.95
Karateka $19.95
Toy Shop $22.95
Bank St. Writer + ... $44.95

Epyx:
Apshai Trilogy $11.95
Create A Calendar $15.95
Destroyer $22.95
Movie Monster $11.95
Rogue $16.95
St. Sports Basketball ... $22.95
Sub Battle Simulator ... $22.95
Winter Games $22.95
World Games $22.95

Firebird:
Starglider $25.95
Guild of Thieves $25.95

Microleague:
Microleag. Baseball $22.95
General Manager $16.95
Stat Disk $13.95
'86 Team Disk $11.95

Microprose:
Conflict in Vietnam $22.95
Crusade in Europe $22.95
Decision in Desert $22.95
F-15 Strike Eagle $22.95
Silent Service $22.95
Gunship $27.95

Origin:
Ultima I $22.95
Ultima III $22.95
Ultima IV $33.95
Moebius $33.95
Ogre $16.95

Strategic Simulations:
Road War 2000 ..... $22.95
Kampfgruppe $33.95
Shard of Spring $22.95
Battle of Antetiem $33.95
Computer Baseball $7.95
Gettysburg $33.95
Wizards Crown $22.95
Chickamauga $33.95

Sublogic:
Jet Simulator $31.49
Scenery Japan $17.95
Scenery San Fran. $17.95
Scenery #1-#6 ..... $64.95
Flight Simulator $34.95

Timeworks:
Swiftcalc $39.95
Wordwriter $34.95

Unison World:
Art Gallery 2 ..... $14.95
News Master $49.95
Print Master (+) $29.95
Fonts & Borders $17.95

e shipping on Prepaid cash orders over $50 in the Continental US.

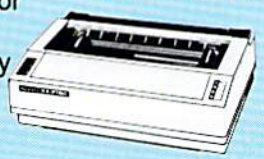
Price Guarantee

Since 1981, we have led the industry by continuing to offer the lowest national prices while providing quality service. Many companies have come and gone trying to imitate our quality and service. If by some oversight we do not have the lowest prices advertised on the products you desire, then we would appreciate the opportunity to rectify this oversight.

New 1080i Model II

- Now 150 cps Draft Mode
- NLQ Mode
- Friction & Tractor Feed
- 2-Year Warranty

Panasonic 1080i *Model II*



$159.95

Monitors

Thomson:

230 Amber TTL/12"	$85
450 Amber TTL/15"	$129.95
4120 CGA	$199.95
4160 CGA	$259.95
4460 EGA	$319.95
4375 UltraScan	$459.95

Zenith:

ZVM 1220	$89
ZVM 1230	$89

Blue Chip:

BCM 12" Green TTL	$79
BCM 12" Amber TTL	$89
BCM 14" Color	$259.95

NEC:

Multisync	$549

Save $90 over NEC Multisync with Thomson 4375 UltraScan $459.95

THOMSON 4120 Monitor

- 14" RGBI/video composite/analog
- Compatible with IBM and Commodore
- RGB data cable included

$199.95

Modems

Avatex:

1200i PC Card	$99
1200hc Modem	$99.95
2400	$219
2400i PC Card	$CALL

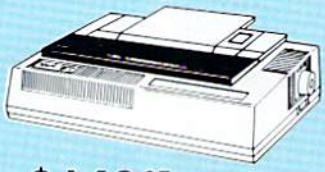
Hayes:

Smartmodem 300	$149.95
Smartmodem 1200	$285.95
Smartmodem 2400	$425.95

Wide Carriage Special

Star Micronics Inc NX-10

- 120 cps Draft
- EZ Front Panel Selection
- NLQ Mode

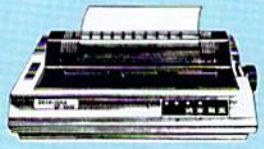


$149.95

Great Performance Great Price

SEIKOSHA SP 180Ai

- 100 cps draft
- 20 cps NLQ
- direct connect for Commodore



$124.95

PRINTERS

SEIKOSHA

SP 180Ai	$124.95
SP 180VC	$124.95
SP 1000VC	$135.95
SP 1200VC	$144.95
SP 1200Ai	$155.95
SP 1200AS RS232	$155.95
SL 80Ai	$295.95
MP1300Ai	$269.95
MP5300Ai	$375.95
BP5420Ai	$879.95
SP Series Ribbon	$7.95
SK3000 Ai	$349
SK3005 Ai	$429

BROTHER

M1109	$195
M1409	$299
M1509	$365
M1709	$475
Twinwriter 6 Dot & Daisy	$899
M1724L	$CALL
HR20	$339
HR40	$569
HR60	$709.95

Star Micronics Inc

NP-10	$134.95
NX-10	$149.95
NX-10C w/interface	$165.95
NL-10	$199.95
NX-15	$289.95
NX-1000	$New!
ND-10	$259.95
ND-15	$375.95
NR-15	$409.95
NB-15 24 Pin	$699.95
NB24-10 24 Pin	$409.95
NB24-15 24 Pin	$549.95

CITIZEN

120 D	$149.95
180 D	$175.95
MSP-10	$259.95
MSP-40	$289.95
MSP-15	$324.95
MSP-50	$359.95
MSP-45	$425.95
MSP-55	$459.95
Premiere 35	$464.95
Tribute 224	$624.95

EPSON

LX800	$175.95
FX86E	$289.95
FX286E	$429.95
EX800	$374.95
EX1000	$509.95
LQ800	$369.95
LQ1000	$549.95
LQ2500	$849.95
GQ3500	$LOW
LQ850	$499.95
LQ1050	$679.95

OKIDATA

Okimate 20	$119
Okimate 20 w/cart	$179.95
120	$189.95
180	$219.95
182	$199.95
192+	$309.95
193+	$449.95
292 w/interface	$449.95
293 w/interface	$585.95
294 w/interface	$819.95
393	$955.95

Panasonic

1080i Model II	$159.95
1091i Model II	$179.95
1092i	$295
1592	$379
1595	$419
3131	$259
3151	$399
KXP 4450 Laser	$CALL
1524 24 Pin	$549.95

SILVER REED

EXP 420P	$199
EXP 600P	$299.95
EXP 800P	$649

Toshiba

321SL	$489
341 SL	$659
P351 Model II	$899

DIABLO

D25	$499.95
635	$779.95

Join the thousands who shop Lyco and Save!

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

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Gradebook

See instructions in article on page 33 before typing in.

```
C000:A9 13 85 AA A9 C8 85 A8 9A
C008:A9 04 85 A9 A9 00 85 A6 A9
C010:85 AF 85 AB 85 AC A9 05 E2
C018:85 A7 85 B0 A9 80 8D 8A F6
C020:02 A9 01 85 8B A9 8E 20 C5
C028:D2 FF A9 08 20 D2 FF 20 35
C030:42 C0 20 EC C1 A2 14 A0 37
C038:00 88 D0 FD CA D0 FA 4C B1
C040:32 C0 A9 00 85 06 85 04 93
C048:85 A6 A9 05 85 A7 A9 0F E8
C050:8D 86 02 8D 21 D0 A9 93 86
C058:20 D2 FF A9 00 8D 20 D0 80
C060:A9 0B 8D 21 D0 A0 13 20 8C
C068:91 C0 C8 C8 C8 C8 C0 28 9B
C070:90 F5 A9 2B A0 CA 20 1E 2E
C078:AB A2 18 A0 00 18 20 F0 17
C080:FF A9 6A A0 CA 20 1E AB 82
C088:20 B8 C0 20 54 C9 4C E8 AD
C090:C6 20 A1 C0 A2 19 A9 42 CC
C098:91 02 20 AA C0 CA D0 F6 DB
C0A0:60 A9 00 85 02 A9 04 85 59
C0A8:03 60 A5 02 18 69 28 85 D4
C0B0:02 A5 03 69 00 85 03 60 10
C0B8:A5 06 AA 85 AD A0 00 20 4C
C0C0:A1 C0 8A 18 69 01 20 83 29
C0C8:C1 E8 C8 C0 05 D0 F3 A0 7E
C0D0:00 20 AA C0 A6 AD BD 80 A3
C0D8:65 20 83 C1 E8 C8 C0 05 92
C0E0:D0 F4 20 AA C0 A0 00 A6 E5
C0E8:AD BD 80 66 20 83 C1 E8 A2
C0F0:C8 C0 05 D0 F4 A9 00 85 88
C0F8:8E A5 04 4A 66 8E 48 A9 F7
C100:00 18 65 8E 85 FB 68 69 75
C108:42 85 FC 48 A5 FB 48 A9 89
C110:C8 85 02 A9 04 85 03 A2 13
C118:13 A0 00 B1 FB D0 02 A9 39
C120:20 91 02 C8 C0 13 D0 F3 CC
C128:A5 FB 18 69 14 85 FB A5 6B
C130:FC 69 00 85 FC A5 06 18 87
C138:65 FB 85 FB A5 FC 69 00 D1
C140:85 FC A0 00 B1 FB 20 83 1B
C148:C1 C8 C0 05 D0 F6 68 18 92
C150:69 80 85 FB A8 68 69 00 D2
C158:85 FC 48 98 48 20 AA C0 49
C160:CA D0 B6 A6 06 A0 00 BD 2F
C168:80 67 20 83 C1 E8 C8 C0 46
C170:05 D0 F4 A0 27 B9 C0 07 FB
C178:09 80 99 C0 07 88 10 F5 50
C180:68 68 60 85 05 8A 48 98 32
C188:48 0A 0A 18 69 14 A8 A2 05
C190:02 A9 2F 95 F7 CA 10 F9 C3
C198:A2 00 A5 05 C9 C9 D0 0C 95
C1A0:A2 02 BD E9 C1 95 F7 CA 6B
C1A8:10 F8 30 11 F6 F7 38 FD 8F
C1B0:E6 C1 10 F8 18 7D E6 C1 EF
C1B8:E8 E0 03 D0 EF A2 00 86 E6
C1C0:FD B5 F7 C9 30 D0 0A A5 CA
C1C8:FD D0 04 A9 20 D0 04 A9 90
C1D0:30 E6 FD 91 02 C8 E8 E0 E4
C1D8:02 90 E6 E6 FD E0 03 90 D6
C1E0:E0 68 A8 68 AA 60 64 0A 34
C1E8:01 20 2A 20 A9 00 85 C6 5B
C1F0:A5 A7 85 B0 A5 A6 85 AF 6F
C1F8:20 E2 C6 48 20 BC C6 A5 C9
C200:8B 85 B5 68 C9 5F D0 08 5F
C208:A5 8B 49 04 85 8B D0 E8 91
C210:C9 13 D0 17 AE 8D 02 D0 4B
C218:12 20 54 C9 A9 00 85 06 34
C220:85 04 85 AF A9 05 85 B0 32
C228:4C 48 C2 C9 0D F0 04 C9 D8
C230:11 D0 2B 20 54 C9 A6 B0 A1
C238:E0 17 D0 12 A5 04 C9 33 33
C240:F0 02 E6 04 C6 B5 D0 EE 79
C248:20 B8 C0 4C E8 C6 E6 B0 C9
C250:A5 B0 C9 03 D0 05 18 69 72
C258:02 85 B0 4C 44 C2 C9 91 6D
C260:D0 30 20 54 C9 A4 B0 A6 8C
C268:AF F0 06 C0 01 D0 06 F0 17
C270:14 C0 05 F0 10 C6 B0 A5 82
C278:B0 C9 04 D0 05 38 E9 02 35
C280:85 B0 4C 8B C2 A5 04 F0 DC
C288:02 C6 04 C6 B5 D0 D6 4C 98
C290:48 C2 C9 9D D0 23 20 54 A5
C298:C9 A6 AF F0 0A A4 B0 C0 B6
C2A0:05 B0 0D E0 01 D0 09 A6 88
C2A8:06 F0 07 C6 06 4C B2 C2 44
C2B0:C6 AF C6 B5 D0 E3 4C 48 B0
C2B8:C2 C9 1D D0 1D 20 54 C9 9E
C2C0:A6 AF E0 05 D0 0B A6 06 F7
C2C8:E0 5F F0 07 E6 06 4C D3 E0
C2D0:C2 E6 AF C6 B5 D0 E9 4C E4
C2D8:48 C2 C9 20 90 07 C9 5B FD
C2E0:B0 03 4C 43 C7 C9 14 D0 9B
C2E8:10 A5 AB F0 0C C6 AB A4 DB
C2F0:AB A9 20 91 A8 C8 91 A8 08
C2F8:60 C9 5C D0 4B A5 06 85 3C
C300:B5 20 54 C9 B0 3F 20 F5 4A
C308:C7 A9 64 85 06 A5 A6 85 5C
C310:B6 A9 01 85 A6 20 C2 C9 DA
C318:E6 A6 20 C2 C9 A5 B6 85 C4
C320:A6 20 F5 C0 A2 00 A0 14 38
C328:18 20 F0 FF A9 7E A0 CA 35
C330:20 1E AB A5 B5 85 06 20 0F
C338:E2 C6 C9 10 D0 07 A9 01 14
C340:85 B6 20 60 C4 4C 48 C2 EC
C348:C9 93 D0 32 20 FE C5 D0 30
C350:2A 20 22 CA A2 27 A0 00 D8
C358:A9 C9 91 02 88 D0 FB E6 DF
C360:03 CA D0 F6 20 22 CA A2 67
C368:47 A0 13 A9 00 91 02 88 8B
C370:10 F9 20 E5 CB CA D0 F1 FD
C378:4C 48 C2 4C F8 C1 C9 13 CA
C380:D0 41 20 16 C6 20 2F C6 02
C388:B0 39 20 02 CC A5 8E C9 BE
C390:54 F0 17 A9 0F A2 08 A8 B7
C398:20 BA FF A5 AD 18 69 03 DC
C3A0:A2 21 A0 CC 20 BD FF 20 BA
C3A8:C0 FF A5 AD A2 24 A0 CC D3
C3B0:20 BD FF 20 22 CA A9 02 4B
C3B8:A2 80 A0 68 38 20 D8 FF 40
C3C0:4C DC C3 C9 0C D0 7F 20 7D
C3C8:16 C6 20 2F C6 B0 77 20 0C
C3D0:02 CC A9 00 A2 00 A0 42 5A
C3D8:18 20 D5 FF A5 8E C9 54 7E
C3E0:F0 45 A2 04 A0 00 18 20 1C
C3E8:F0 FF A9 0F 20 C3 FF A9 C8
C3F0:0F A2 08 A8 20 BA FF A9 C9
C3F8:00 20 BD FF 20 C0 FF A2 E6
C400:0F 20 C6 FF 20 CF FF C9 FB
C408:2C D0 F9 20 CF FF C9 2C 5B
C410:F0 06 20 D2 FF 4C 0B C4 D0
C418:20 CF FF C9 0D D0 F9 A9 8B
C420:0F 20 C3 FF 20 CC FF A9 8F
C428:02 20 C3 FF 20 24 C6 A9 FB
C430:01 85 C0 A5 01 09 20 85 FF
C438:01 A9 29 A0 CB 20 1E AB A2
C440:20 E2 C6 4C 02 CC C9 01 08
C448:D0 0B 20 54 C9 B0 03 20 7D
C450:37 CB 4C 48 C2 C9 10 F0 C4
C458:03 4C 42 C5 A9 00 85 B6 2A
C460:20 CF CB 8A F0 EC A5 06 9C
C468:18 65 A6 85 FD 18 69 13 BA
C470:85 05 A9 00 85 02 A9 42 FC
C478:85 03 A9 04 A2 04 A0 00 61
C480:20 BA FF A9 00 20 BD FF 5F
C488:20 C0 FF A2 04 20 C9 FF B0
C490:A5 A6 05 B6 F0 4B A5 B6 59
C498:F0 0D A9 0E A0 CB 20 1E 86
C4A0:AB A9 78 85 05 D0 34 A9 4F
C4A8:E6 A0 CA 20 1E AB A6 FD 14
C4B0:20 E1 C5 20 10 C6 A9 FA 67
C4B8:A0 CA 20 1E AB A4 FD 88 9F
C4C0:B9 80 65 AA 20 E1 C5 20 D2
C4C8:10 C6 A9 6A A0 CA 20 1E 76
C4D0:AB A4 FD 88 B9 80 67 AA EA
C4D8:20 E1 C5 20 10 C6 20 10 91
C4E0:C6 20 E1 FF F0 53 A9 FE 39
C4E8:85 B5 20 E1 FF F0 4A A0 BD
C4F0:00 B1 02 C9 1B B0 06 09 74
C4F8:40 D0 02 A9 20 20 D2 FF D8
C500:C8 C0 13 D0 EC A5 A6 05 DF
C508:B6 F0 24 A4 05 B1 02 C9 B6
C510:C9 90 08 A9 2A 20 D2 FF B7
C518:4C 1F C5 AA 20 E1 C5 A5 AE
C520:B6 F0 0C A9 20 20 D2 FF 86
C528:A4 05 C8 E6 B5 D0 DE 20 9D
C530:E5 CB 20 10 C6 C6 BE D0 46
C538:A8 A9 04 20 C3 FF 20 CC 30
C540:FF 60 C9 18 D0 06 20 FE 7C
C548:C5 D0 03 00 C9 04 F0 03 8E
C550:4C DB C5 20 AC C6 C9 43 0B
C558:D0 2D 20 CF CB A5 A6 F0 CB
C560:7A 18 65 06 85 8E 69 13 88
C568:A8 20 22 CA A9 C9 91 02 DA
C570:20 E5 CB CA D0 F6 A4 8E E5
C578:88 A9 C9 99 80 67 99 80 DA
C580:65 99 80 66 4C 48 C2 C9 6E
C588:52 D0 50 A5 A7 C9 03 90 D0
C590:4A A5 A6 48 A9 00 85 B5 12
C598:85 A6 20 83 C7 68 85 A6 5E
C5A0:A5 02 18 69 80 85 FB A5 D0
C5A8:03 69 00 85 FC A0 00 B1 84
C5B0:FB C0 00 D0 06 29 DF D0 DC
C5B8:02 E6 B5 91 02 C8 C0 80 04
C5C0:D0 ED 20 E5 CB A5 B5 F0 E3
C5C8:D7 A0 7F A9 C9 91 02 88 14
C5D0:C0 13 D0 F9 A9 00 91 02 AD
C5D8:88 10 FB 20 B8 C0 4C 48 D7
C5E0:C2 8A A0 03 88 30 05 D9 9D
C5E8:E6 C1 B0 F8 98 30 09 C8 5E
C5F0:A9 20 20 D2 FF 88 D0 F8 47
C5F8:A9 00 20 CD BD 60 A9 C8 C5
C600:A0 CA 20 D0 C6 20 E2 C6 E4
C608:C9 59 08 20 BC C6 28 60 85
C610:A9 0D 20 D2 FF 60 20 54 FC
C618:C9 A2 FE B5 01 9D 00 CF BC
C620:CA D0 F8 60 A2 FE BD 00 F8
C628:CF 95 01 CA D0 F8 60 A9 A4
C630:A6 A0 CA 20 D0 C6 20 E2 59
C638:C6 C9 54 85 8E D0 04 A2 E0
C640:01 D0 0B C9 44 F0 05 20 90
C648:BC C6 38 60 A2 08 A9 02 7D
C650:A0 01 20 BA FF 20 BC C6 DE
C658:A9 B9 A0 CA 20 D0 C6 A9 65
C660:00 85 CC 85 AD 20 E2 C6 BB
C668:C9 0D F0 1C C9 14 F0 2C AA
C670:C9 20 90 F1 C9 5B B0 ED 27
C678:A4 AD 99 24 CC 20 D2 FF C5
C680:E6 AD A5 AD C9 0C D0 DD 7A
C688:20 BC C6 A9 01 85 CC A5 26
C690:AD F0 B4 A2 24 A0 CC 20 4F
C698:BD FF 18 60 A5 AD F0 C5 99
C6A0:A9 C2 A0 CA 20 1E AB C6 0C
C6A8:AD 4C 65 C6 A9 92 A0 CA DC
C6B0:20 D0 C6 20 E2 C6 48 20 40
C6B8:BC C6 68 60 A9 20 A0 12 8A
C6C0:99 A0 04 88 10 FA 60 A9 23
C6C8:00 85 02 A9 42 85 03 60 21
C6D0:48 98 48 A2 04 A0 00 18 96
C6D8:20 F0 FF 68 A8 68 20 1E 7E
C6E0:AB 60 20 E4 FF F0 FB 60 CA
C6E8:20 37 C7 20 54 C9 20 37 90
C6F0:C7 A5 AF 85 A6 A5 B0 85 CC
C6F8:A7 20 A1 C0 A6 A7 20 AA 61
C700:C0 CA D0 FA A6 A6 BD 31 E8
C708:C7 18 65 02 85 A8 A5 03 6B
C710:69 00 85 A9 A0 02 A5 A6 9E
C718:D0 02 A0 12 84 AA 20 B8 8D
C720:C0 A4 AA B1 A8 09 80 91 A5
C728:A8 88 10 F7 A9 00 85 AB B3
C730:60 00 14 18 1C 20 24 A4 42
C738:AA B1 A8 29 7F 91 A8 88 4D
C740:10 F7 60 48 A6 A6 D0 0A E1
C748:C9 20 F0 06 C9 41 B0 02 F9
C750:68 60 A5 AB D0 09 A4 AA 3A
C758:A9 20 91 A8 88 10 F9 68 62
C760:C9 41 90 03 38 E9 40 A4 F5
C768:AB 48 20 83 C7 68 09 80 8E
C770:91 A8 A2 12 A5 A6 F0 02 14
```

```
C778:A2 02 E4 AB F0 02 E6 AB 3A
C780:84 AC 60 A9 00 85 8E A5 FC
C788:04 4A 66 8E 48 A9 00 18 63
C790:65 8E 85 02 68 69 42 85 3A
C798:03 A6 A7 CA E0 03 B0 23 8C
C7A0:8A 18 69 65 85 03 A5 06 88
C7A8:18 69 80 85 02 A5 03 69 1D
C7B0:00 85 03 A6 A6 CA 8A 18 F9
C7B8:65 02 85 02 A5 03 69 00 58
C7C0:85 03 60 CA CA CA CA F0 94
C7C8:06 20 E5 CB CA D0 FA A6 13
C7D0:A6 F0 1B A5 06 18 65 02 0B
C7D8:85 02 A5 03 69 00 85 03 E9
C7E0:A5 02 18 7D EF C7 85 02 4A
C7E8:A5 03 69 00 85 03 60 00 32
C7F0:14 15 16 17 18 20 2B C9 65
C7F8:A9 80 85 02 A9 65 85 03 3F
C800:20 F0 C8 84 8C 85 8D 20 F4
C808:CF CB 86 B1 A9 14 85 02 0B
C810:A9 42 85 03 A5 8C 85 AD FF
C818:A5 8D 85 AE 20 F0 C8 20 F1
C820:91 B3 20 0F BC A4 AD A5 D5
C828:AE 20 91 B3 20 F4 CB 90 83
C830:01 60 20 E2 BA 20 E2 BA 63
C838:20 49 B8 20 A1 B7 8A A0 E6
C840:64 91 02 20 E5 CB C6 B1 48
C848:D0 CA 20 2B C9 20 CF CB E5
C850:86 B1 A9 14 85 02 A9 42 D1
C858:85 03 A9 00 85 8E A0 07 51
C860:99 3C 03 88 10 FA 85 B2 E0
C868:A4 B2 B9 80 66 F0 5A C9 AD
C870:C9 B0 4E 85 AD A4 B2 B1 4C
C878:02 C9 C9 B0 44 A5 AD 65 3B
C880:8E 85 8E A4 AD A9 00 20 0B
C888:91 B3 20 0F BC A4 B2 B9 5C
C890:80 65 A8 A9 00 20 91 B3 C2
C898:20 F4 CB 90 01 60 20 0F D2
C8A0:BC A4 B2 B1 02 A8 A9 00 31
C8A8:20 91 B3 A5 61 20 2B BA 1C
C8B0:A9 3C A0 03 20 8C BA 20 33
C8B8:6A B8 A2 3C A0 03 20 D4 EB
C8C0:BB E6 B2 A5 B2 C9 64 D0 F0
C8C8:9F A5 8E C9 64 F0 04 A9 9A
C8D0:C9 D0 0E A9 3C A0 03 20 62
C8D8:A2 BB 20 49 B8 20 A1 B7 84
C8E0:8A A0 65 91 02 20 E5 CB CD
C8E8:C6 B1 F0 03 4C 5A C8 60 56
C8F0:A9 00 85 05 85 FD A0 63 21
C8F8:B1 02 C9 C9 90 18 B9 80 92
C900:65 C9 C9 B0 1E A5 AD 38 18
C908:F9 80 65 85 AD A5 AE E9 09
C910:00 85 AE 4C 23 C9 A5 05 30
C918:18 71 02 85 05 A5 FD 69 D0
C920:00 85 FD 88 10 D2 A4 05 77
C928:A5 FD 60 A2 10 A9 00 95 01
C930:61 CA 10 F9 60 A9 14 85 20
C938:02 A9 43 85 03 A0 6C A9 15
C940:D9 A0 CA 20 D0 C6 38 60 B6
C948:20 83 C7 B1 A8 91 02 88 F8
C950:10 F9 18 60 20 37 C7 A4 85
C958:AA C0 12 F0 EB A9 00 85 4E
C960:05 85 AB A2 02 B1 A8 C9 69
C968:2A D0 0C A5 A7 C9 03 90 1C
C970:CE A9 C9 85 05 D0 3D 38 85
C978:E9 30 F0 14 C9 F0 F0 11 71
C980:85 FD A5 05 18 7D E6 C1 A1
C988:85 05 B0 B3 C6 FD D0 F2 34
C990:CA 88 10 D1 A5 A7 C9 03 2D
C998:90 1A A5 06 18 65 A6 A8 5C
C9A0:88 B9 80 65 C9 C9 D0 02 66
C9A8:A9 00 C5 05 90 91 A5 05 35
C9B0:C9 C9 B0 8B A0 00 20 83 33
C9B8:C7 A5 05 91 02 20 C2 C9 33
C9C0:18 60 20 2B C9 20 CF CB 69
C9C8:8A F0 55 A5 06 18 65 A6 E4
C9D0:85 8E 69 13 A8 20 22 CA FD
C9D8:A9 00 85 05 85 FD B1 02 CB
C9E0:C9 C9 90 05 C6 BE 4C F4 EC
C9E8:C9 18 65 05 85 05 A5 FD ED
C9F0:69 00 85 FD 20 E5 CB CA C4
C9F8:D0 E4 A5 FD A4 05 20 91 CD
CA00:B3 20 0C BC A4 BE F0 18 DE
CA08:A9 00 20 91 B3 20 F4 CB 63
CA10:90 01 60 20 49 B8 20 A1 4B
CA18:B7 8A A4 8E 88 99 80 67 BC
CA20:18 60 A9 00 85 02 A9 42 D8
CA28:85 03 60 13 1D 1D 1D 1D 33
CA30:1D 1D 41 53 53 49 47 4E 95
CA38:4D 45 4E 54 20 23 0D 1D 99
CA40:1D 1D 1D 1D 1D 50 45 52 28
CA48:46 45 43 54 20 4D 41 52 0B
CA50:4B 0D 1D 1D 1D 1D 1D 1D F8
CA58:1D 1D 25 20 4F 46 20 47 85
CA60:52 41 44 45 0D 4E 41 4D BD
CA68:45 00 41 56 45 52 41 47 6B
CA70:45 20 20 20 20 20 20 20 98
CA78:20 20 20 20 20 00 41 56 06
CA80:47 1D 46 49 4E 1D 20 20 A5
CA88:20 1D 20 20 20 1D 20 20 51
CA90:20 00 12 28 52 29 4F 57 28
CA98:20 4F 52 20 28 43 29 4F 4E
CAA0:4C 55 4D 4E 92 00 12 28 21
CAA8:54 29 41 50 45 20 4F 52 7B
CAB0:20 28 44 29 49 53 4B 92 3C
CAB8:00 12 4E 41 4D 45 3F 92 41
CAC0:20 00 20 9D 9D 20 9D 00 EC
CAC8:12 20 41 52 45 20 59 4F 69
CAD0:55 20 53 55 52 45 3F 92 91
CAD8:00 12 44 41 54 41 20 45 BC
CAE0:52 52 4F 52 92 00 41 53 AD
CAE8:53 49 47 4E 4D 45 4E 54 B8
CAF0:20 4E 55 4D 42 45 52 20 95
CAF8:20 00 50 45 52 46 45 43 76
CB00:54 20 4D 41 52 4B 20 20 A7
CB08:20 20 20 20 20 00 4E 41 9C
CB10:4D 45 20 20 20 20 20 20 87
CB18:20 20 20 20 20 20 20 20 AF
CB20:20 41 56 47 20 46 49 4E 52
CB28:00 0D 0D 48 49 54 20 41 46
CB30:4E 59 20 4B 45 59 00 20 AD
CB38:CF CB E0 02 B0 01 60 E6 18
CB40:BE A9 12 85 B8 A9 00 85 2E
CB48:B9 A9 B0 8D A5 CB A5 A6 64
CB50:F0 13 A5 06 18 65 A6 18 F5
CB58:69 13 85 B9 A9 01 85 B8 CA
CB60:A9 90 8D A5 CB 20 22 CA EA
CB68:A9 00 85 F7 A6 BE CA CA 95
CB70:A5 02 18 69 80 85 FB A5 AC
CB78:03 69 00 85 FC 86 FD A2 E4
CB80:FF A4 B9 88 C8 E8 E4 B8 6D
CB88:F0 30 A5 B8 C9 01 D0 07 DF
CB90:B1 FB D1 02 4C A3 CB B1 94
CB98:02 29 DF 85 B5 B1 FB 29 65
CBA0:DF C5 B5 F0 DF B0 13 A0 E7
CBA8:7F B1 02 85 8E B1 FB 91 C9
CBB0:02 A5 8E 91 FB 88 10 F1 B1
CBB8:E6 F7 20 E5 CB A6 FD CA E3
CBC0:D0 AE C6 BE A5 BE C9 02 EE
CBC8:F0 04 A5 F7 D0 97 60 20 D3
CBD0:22 CA A0 00 A2 FF E8 B1 D8
CBD8:02 29 DF D0 03 86 BE 60 D4
CBE0:20 E5 CB B0 F1 A5 02 18 C8
CBE8:69 80 85 02 A5 03 69 00 32
CBF0:85 03 38 60 A5 61 D0 05 72
CBF8:20 3F C9 38 60 20 12 BB 90
CC00:18 60 AD 18 D0 49 20 8D 6E
CC08:18 D0 AD 88 02 49 08 8D F2
CC10:88 02 48 A9 93 20 D2 FF D4
CC18:68 C9 04 D0 03 20 4E C0 DB
CC20:60 53 30 3A 00 00 00 00 68
```

Autoboot Maker

Article on page 60.

```
FF 100 INPUT "NAME OF PROGRAM
       {SPACE}TO LOAD";PN$
RH 110 INPUT "NAME OF BOOT PRO
       GRAM";BN$
BE 120 INPUT "LOCATION TO JUMP
        AFTER LOAD";LOC$
XS 130 IF LOC$="" THEN BASIC=1
       :GOTO 160
CQ 140 IF LEFT$(LOC$,1)="$" TH
       EN GOSUB 760:GOTO 160
MG 150 LOC=VAL(LOC$)
PX 160 INPUT "KILL RUNSTOP/RES
       TORE (Y/N)";RN$
EK 170 RN$=LEFT$(RN$,1)
FE 180 IF NOT (RN$="Y" OR RN$=
       "N") THEN 160
FE 190 OPEN 15,8,15,"I"
BB 200 OPEN 8,8,8,"0:"+BN$+",P
       ,W"
KC 210 PRINT#8,CHR$(167)CHR$(0
       2);
XX 220 PRINT#8,CHR$(LEN(PN$));
XQ 230 PRINT#8,PN$;
KB 240 IF RN$="N" THEN 380
JX 250 FOR C=1 TO 17
BM 260 READ CODE
SK 270 PRINT#8,CHR$(CODE);
EE 280 NEXT C
SJ 290 DATA 120{9 SPACES}: REM
        SEI
KK 300 DATA 169,052{5 SPACES}:
        REM LDA #52
AK 310 DATA 141,020,003 : REM
       {SPACE}STA $0314
SK 320 DATA 169,134{5 SPACES}:
        REM LDA #134
PM 330 DATA 141,024,003 : REM
       {SPACE}STA $0318
QF 340 DATA 169,234{5 SPACES}:
        REM LDA #234
SG 350 DATA 141,025,003 : REM
       {SPACE}STA $0319
XQ 360 DATA 088{9 SPACES}: REM
        CLI
MS 370 GOTO 390
FK 380 FOR C=1 TO 17:READ CODE
       :NEXT C
BP 390 FOR C=1 TO 38
GC 400 READ CODE
PB 410 PRINT#8,CHR$(CODE);
MR 420 NEXT C
AH 430 DATA 169,002{5 SPACES}:
       REM LDA #2
RX 440 DATA 162,008{5 SPACES}:
       REM LDX #8
SR 450 DATA 160,255{5 SPACES}:
       REM LDY #255
BA 460 DATA 032,186,255 :REM J
       SR $FFBA
KD 470 DATA 173,167,002 :REM L
       DA $02A7
JE 480 DATA 162,168{5 SPACES}:
       REM LDX #$A8
EH 490 DATA 160,002{5 SPACES}:
       REM LDY #$02
DG 500 DATA 032,189,255 :REM J
       SR $FFBD
PB 510 DATA 169,000{5 SPACES}:
       REM LDA #$00
AH 520 DATA 032,213,255 :REM J
       SR $FFD5
MH 530 DATA 169,131{5 SPACES}:
       REM LDA #131
MX 540 DATA 141,002,003 :REM S
       TA $0302
XH 550 DATA 169,164{5 SPACES}:
       REM LDA #164
SP 560 DATA 141,003,003 :REM S
       TA $0303
MP 570 DATA 134,045{5 SPACES}:
       REM STX $2D
EG 580 DATA 132,046{5 SPACES}:
       REM STY $2E
AA 590 IF BASIC=1 THEN GOSUB 8
       60:GOTO 620
HD 600 L1=INT(LOC/256):L2=LOC-
       (L1*256)
CM 610 PRINT#8,CHR$(76)CHR$(L2
       )CHR$(L1);
RP 620 IF RN$="Y" THEN BU=58+L
       EN(PN$)+1
CR 630 IF RN$="N" THEN BU=41+L
       EN(PN$)+1
GE 640 IF BASIC=1 THEN BU=BU+1
       1
GR 650 BL=88-BU
```

```
KC 660 FOR C=1 TO BL+1:PRINT#8
       ,CHR$(0);:NEXT C
KP 670 PRINT#8,CHR$(139)CHR$(2
       27);
GH 680 B=679+LEN(PN$)+1
SD 690 L1=INT(B/256):L2=B-(L1*
       256)
CA 700 PRINT#8,CHR$(L2)CHR$(L1
       );
JP 710 CLOSE8
AS 720 GET#15,A$:S=ST
AJ 730 PRINT A$;
QE 740 IF S=0 THEN 720
KF 750 END
KP 760 L=LEN(LOC$)
ED 770 S=L-1
RK 780 FOR C=1 TO L
JR 790 I$=MID$(LOC$,C,1)
MD 800 IF I$<="9" THEN I$=STR$
       (VAL(I$))
MB 810 IF I$=>"A" THEN I$=STR$
       (ASC(I$)-55)
QQ 820 LOC=LOC+VAL(I$)*16↑S
FK 830 S=S-1
AH 840 NEXT C
SM 850 RETURN
JC 860 FOR C=1 TO 14
JA 870 READ CODE
RE 880 PRINT#8,CHR$(CODE);
EP 890 NEXT C
CK 900 DATA 169,000{5 SPACES}:
       REM LDA #$00
EA 910 DATA 133,122{5 SPACES}:
       REM STA $7A
EG 920 DATA 169,008{5 SPACES}:
       REM LDA #$08
RR 930 DATA 133,123{5 SPACES}:
       REM STA $78
KP 940 DATA 032,096,166 :REM J
       SR $A660
QG 950 DATA 076,174,167 :REM J
       MP $A7AE
SD 960 RETURN
```

Snapshot

Article on page 28.

Program 1: Snapshot

```
KQ 100 POKE56,146:CLR:SC=1024:
       CM=54272:XT=0:YT=0:X=0:
       Y=0:R=24:C=39:DIMNA$(31
       )
KQ 110 SNAPSHOT=51000:POKE5328
       0,11:POKE53281,12:GOSUB
       240
AE 120 PRINT TAB(7)"SET UP AND
        COPY SCREENS":FORI=0TO
       1200:NEXT
BK 10 GOSUB240:PRINTTAB(9)"SNA
      PSHOT DEMONSTRATION"
EA 140 SYSSNAPSHOT,D,*:REM ERA
       SE OLD SCREENS
BB 150 FORI= 0 TO 31:X=I:Y=0:G
       OSUB250
SQ 160 X=X-1:Y=Y+1:IF X>=0 THE
       N GOSUB250:GOTO160
ME 170 NA$(I)=STR$(I):NA$(I)=R
       IGHT$(NA$(I),LEN(NA$(I)
       )-1):N$=NA$(I)
EM 180 SYSSNAPSHOT,C,N$:NEXT:R
       EM COPY SCREEN
ME 190 GOSUB240:PRINT TAB(8)"S
       NAPSHOT PRINTS SCREENS"
KQ 200 PRINT:PRINT:PRINT TAB(1
       0)"PRESS (STOP) TO END"
       :FORI=0TO3000:NEXT
KJ 210 FORI=0TO31:N$=NA$(I):SY
       SSNAPSHOT,P,N$:NEXT
QA 220 FORI=31TO0STEP-1:N$=NA$
       (I):SYSSNAPSHOT,P,N$:NE
       XT
CX 230 GOTO210
KD 240 PRINT"{CLR}{WHT}":FORI=
       0TO9:PRINT:NEXT:RETURN
CC 250 IF Y>24 THEN300
JX 260 XT=X:YT=Y:CH=78:GOSUB31
       0
XA 270 XT=C-X:CH=77:GOSUB310
JK 280 YT=R-Y:CH=78:GOSUB310
CP 290 XT=X:CH=77:GOSUB310
HJ 300 RETURN
CM 310 P=40*YT+XT+SC:IFPEEK(P)
       =77ORPEEK(P)=78THENCH=8
       6
GD 320 POKEP,CH:POKEP+CM,0
DM 330 RETURN
```

Program 2: Snapshot Demo

```
C124:78 AD 15 03 C9 C1 F0 20 79
C12C:8D E8 C1 AD 14 03 8D E7 73
C134:C1 A9 64 8D 14 03 A9 C1 2A
C13C:8D 15 03 A0 00 8C 36 C7 9C
C144:AD 63 C1 D0 03 20 4E C1 B3
C14C:58 60 98 99 00 C0 C8 D0 26
C154:FA A9 FF 8D FE C0 A9 BF A6
C15C:8D FF C0 8D 63 C1 60 00 7A
C164:AD 36 C7 D0 07 AD 8E 02 60
C16C:C9 04 F0 03 4C E6 C1 A5 4B
C174:C5 CD 13 C1 F0 6C 8D 13 34
C17C:C1 C9 14 F0 1B C9 29 F0 28
C184:1D C9 0D F0 1F C9 2A F0 1F
C18C:21 C9 12 F0 23 C9 21 F0 D7
C194:25 C9 15 F0 27 4C E6 C1 C8
C19C:20 C6 C1 4C 0D C3 20 C6 5D
C1A4:C1 4C C0 C4 20 C6 C1 4C 6C
C1AC:F9 C7 20 C6 C1 4C 12 C8 BB
C1B4:20 C6 C1 4C DF C5 20 C6 14
C1BC:C1 4C F2 C6 20 C6 C1 4C EA
C1C4:67 C2 BA FE 01 01 D0 03 A4
C1CC:FE 02 01 E8 E8 8E 16 C1 6E
C1D4:A9 20 48 48 48 48 8D 36 77
C1DC:C7 85 CC 20 F1 C2 A9 36 65
C1E4:85 01 4C 31 EA 20 FF C2 A2
C1EC:A0 00 8C 36 C7 BA EC 16 CE
C1F4:C1 F0 04 68 4C F1 C1 AE F8
C1FC:37 C7 D0 06 68 A8 68 AA E9
C204:68 40 8C 37 C7 60 A2 00 D7
C20C:84 0B AD 37 C7 F0 04 AC 76
C214:14 C1 60 85 CC 86 0D 20 33
C21C:E4 FF A6 0D C9 0D F0 1A 38
C224:C9 14 D0 05 CA 10 0C 30 DC
C22C:EE 29 7F C9 20 90 E6 9D AE
C234:00 C1 E8 20 D7 C2 E4 0B E7
C23C:D0 DB A9 0D 20 D7 C2 8A 97
C244:D0 03 4C E9 C1 A8 A9 20 3F
C24C:9D 00 C1 60 AD FE C0 85 4F
C254:04 85 08 AD FF C0 85 05 2C
C25C:85 09 A0 00 20 71 C4 20 6B
C264:71 C4 60 20 50 C2 20 C0 70
C26C:C2 85 02 8D 65 C3 86 03 18
C274:8E 69 C3 AD 86 02 85 0D 43
C27C:A2 02 CA E4 0D F0 FB 8E 2E
C284:86 02 AD 21 D0 29 0F CD AC
C28C:86 02 F0 EE 84 CC 20 E4 5F
C294:FF F0 FB C9 0D F0 06 20 CA
C29C:D7 C2 4C 92 C2 A0 00 85 8F
C2A4:CC 20 C0 C2 18 69 02 90 D7
C2AC:01 E8 8D 63 C4 8E 69 C4 CC
C2B4:98 20 D7 C2 A5 0D 8D 86 B8
C2BC:02 4C 3B C3 18 A5 D1 65 5A
C2C4:D3 08 20 71 C4 28 48 A5 4E
C2CC:D2 69 00 20 71 C4 AA 68 74
C2D4:60 85 01 A4 CF F0 0A A0 4D
C2DC:01 84 CD A4 A2 C4 A2 F0 66
C2E4:FC 20 D2 FF A0 02 84 CD 2F
C2EC:A0 00 84 CF 60 A0 00 B9 8F
C2F4:00 00 99 7C C9 C8 C0 0F 77
C2FC:D0 F5 60 A0 00 B9 7C C9 28
C304:99 00 00 C8 C0 0F D0 F5 BE
C30C:60 20 50 C2 84 02 8C 65 AC
C314:C3 98 20 71 C4 AD 88 02 AE
C31C:85 03 20 71 C4 A2 EA 8E 57
C324:63 C4 CA CA 8A 20 71 C4 11
C32C:AD 88 02 18 69 03 8D 69 4A
C334:C4 20 71 C4 8C 69 C3 84 B6
C33C:0A 84 0C 8C 15 C1 20 95 B9
C344:C4 85 0B C9 00 F0 50 20 12
C34C:95 C4 85 0E A5 0B C5 0E 54
C354:F0 36 20 71 C4 A5 0E 85 5B
C35C:0B F0 3C 4C 4B C3 84 0A 6E
C364:A9 00 85 02 A9 00 29 03 34
C36C:09 D8 85 03 85 0C AD 69 B0
C374:C4 29 03 09 D8 8D 69 C4 2E
C37C:20 CD C5 A5 04 91 08 A5 B6
C384:05 C8 91 08 88 4C 42 C3 31
C38C:EE 15 C1 20 95 C4 C5 0B 61
C394:D0 26 EE 15 C1 F0 2B 20 85
C39C:95 C4 C5 0B F0 F4 85 0E FD
C3A4:A9 00 20 71 C4 A5 0B 20 0F
C3AC:71 C4 AD 15 C1 20 71 C4 5B
C3B4:A9 00 8D 15 C1 4C 59 C3 C9
C3BC:85 0E A5 0B 8D 15 C1 4C 80
C3C4:AB C3 CE 15 C1 4C A2 C3 86
C3CC:68 68 AD 15 C1 F0 08 85 11
C3D4:0E CE 15 C1 4C A4 C3 A5 F7
C3DC:0C D0 03 4C 62 C3 A9 20 59
C3E4:85 02 84 0A A9 D0 85 03 7F
C3EC:20 95 C4 20 71 C4 20 95 F8
C3F4:C4 20 71 C4 20 95 C4 20 62
C3FC:71 C4 A9 00 20 5F C7 A0 52
C404:03 20 0A C2 AC FD C0 C8 2C
C40C:C8 AD FE C0 99 00 C0 C8 68
C414:AD FF C0 99 00 C0 C8 8C 47
C41C:FD C0 A5 04 8D FE C0 A5 59
C424:05 8D FF C0 20 B3 C6 C0 BD
C42C:FF F0 03 20 08 C6 AC FD 07
C434:C0 88 88 88 88 C0 FB B0 C9
C43C:0F AD 00 C1 99 00 C0 AD D0
C444:01 C1 99 01 C0 4C E9 C1 CE
C44C:8C FD C0 A9 3A 20 5F C7 27
C454:4C E9 C1 91 02 A6 02 E8 67
C45C:86 02 D0 02 E6 03 E0 EA D3
C464:D0 0A A6 03 E0 07 90 04 26
C46C:A9 01 85 0A 60 91 04 A6 54
C474:04 D0 02 C6 05 CA 86 04 45
C47C:E4 37 D0 14 A6 05 E4 38 EC
C484:D0 0E E6 04 D0 02 E6 05 78
C48C:A9 2B 20 5F C7 4C E9 C1 B4
C494:60 A5 0C F0 1A B1 02 29 0D
C49C:0F 85 0D A6 0A D0 1A 20 03
C4A4:59 C4 B1 02 0A 0A 0A 0A F8
C4AC:18 65 0D 20 59 C4 60 B1 8F
C4B4:02 20 59 C4 A6 0A D0 01 BE
C4BC:60 4C CC C3 A9 00 20 5F 4C
C4C4:C7 A0 03 20 0A C2 20 B3 0C
C4CC:C6 C0 FF D0 08 A9 48 20 8E
C4D4:5F C7 4C E9 C1 B9 00 C0 DD
C4DC:85 04 B9 01 C0 85 05 A0 38
C4E4:00 20 C0 C5 85 09 20 C0 3C
C4EC:C5 85 08 20 C0 C5 85 02 E7
C4F4:85 06 20 C0 C5 85 03 29 46
C4FC:03 09 D8 85 07 20 C0 C5 BD
C504:8D 63 C4 20 C0 C5 8D 69 6B
C50C:C4 84 0A 84 0E A9 02 85 45
C514:0F 20 C0 C5 C9 00 F0 2B FE
C51C:20 57 C4 A5 0A D0 0A C6 EE
C524:0F D0 EE 20 69 C5 4C 15 5B
C52C:C5 C6 0F D0 06 20 69 C5 84
C534:4C 99 C5 A9 60 8D 7E C5 9B
C53C:20 69 C5 A9 4A 8D 7E C5 D0
C544:4C 99 C5 20 C0 C5 85 0B 4A
C54C:20 C0 C5 85 0D E6 0D A5 EC
C554:0B 20 57 C4 C6 0D F0 C3 B4
C55C:C6 0F D0 F3 20 69 C5 4C E6
C564:53 C5 4C 1F C5 A6 0E F0 5C
C56C:07 A5 0C C6 0E 4C 7B C5 31
C574:20 CB C5 C9 00 F0 0F 20 5A
C57C:D6 C5 4A 4A 4A 4A 20 D6 65
C584:C5 A2 02 86 0F 60 20 CB 4A
C58C:C5 85 0C 20 CB C5 85 0E 6E
C594:A5 0C 4C 7B C5 AC BC C5 57
C59C:D0 1F A9 20 85 02 A9 D0 E7
```

```
C5A4:85 03 20 CB C5 20 57 C4 96
C5AC:A6 02 E0 25 B0 0B 4A 4A 0B
C5B4:4A 4A 20 57 C4 4C A6 C5 DB
C5BC:00 4C E9 C1 B1 04 A6 04 A3
C5C4:D0 02 C6 05 C6 04 60 B1 1B
C5CC:08 A6 08 D0 02 C6 09 C6 18
C5D4:08 60 91 06 E6 06 D0 02 02
C5DC:E6 07 60 A9 00 20 5F C7 4B
C5E4:A0 03 20 0A C2 A9 2A 20 57
C5EC:DD C6 A9 AC 20 DD C6 20 3F
C5F4:B3 C6 C0 FF D0 08 A9 48 66
C5FC:20 5F C7 4C E9 C1 20 08 CC
C604:C6 4C E9 C1 84 0D 98 18 03
C60C:69 02 CD FD C0 D0 0F B9 89
C614:00 C0 8D FE C0 B9 01 C0 23
C61C:8D FF C0 4C 7B C6 B9 00 B7
C624:C0 85 02 B9 01 C0 85 03 68
C62C:B9 04 C0 85 04 B9 05 C0 D9
C634:85 05 20 8E C6 A4 0D 38 CD
C63C:B9 00 C0 F9 04 C0 85 08 94
C644:B9 01 C0 F9 05 C0 85 09 E5
C64C:88 88 A6 0D E8 E8 BD 00 4C
C654:C0 99 00 C0 C8 E8 EC FD 76
C65C:C0 90 F3 A4 0D 18 B9 00 73
C664:C0 65 08 99 00 C0 C8 B9 94
C66C:00 C0 65 09 99 00 C0 C8 7E
C674:F0 05 C8 C8 4C 61 C6 AD 84
C67C:FD C0 38 E9 04 8D FD C0 F1
C684:A8 A9 20 99 00 C0 99 01 A5
C68C:C0 60 A2 00 A1 04 81 02 09
C694:A5 02 D0 02 C6 03 C6 02 81
C69C:A5 04 D0 02 C6 05 C6 04 14
C6A4:AD FE C0 C5 04 D0 E5 AD 1A
C6AC:FF C0 C5 05 D0 DE 60 A0 D6
C6B4:00 CC FD C0 B0 1E AD 00 9A
C6BC:C1 D9 00 C0 F0 07 C8 C8 AB
C6C4:C8 C8 4C B5 C6 C8 AD 01 83
C6CC:C1 D9 00 C0 F0 09 C8 C8 C3
C6D4:C8 4C B5 C6 A9 FE A8 C8 60
C6DC:60 CD 00 C1 F0 06 CD 01 66
C6E4:C1 F0 01 60 68 68 A0 00 DB
C6EC:20 4E C1 4C E9 C1 A9 93 58
C6F4:20 D2 FF A9 20 AC FD C0 52
C6FC:99 00 C0 C8 C0 FB 90 F8 0C
C704:A2 54 20 56 C7 A0 00 B9 DD
C70C:00 C0 20 D2 FF B9 01 C0 A6
C714:20 D2 FF A2 0D 20 56 C7 EF
C71C:B9 54 C0 20 D2 FF B9 55 17
C724:C0 20 D2 FF A2 0D 20 56 56
C72C:C7 B9 A8 C0 20 D2 FF 4C C7
C734:3B C7 00 00 4C 5E C8 B9 7A
C73C:A9 C0 20 D2 FF A2 04 20 B4
C744:51 C7 C8 C8 C8 C8 C0 54 53
C74C:D0 BD 4C E9 C1 A9 0D 20 CA
C754:D2 FF A9 20 20 D2 FF CA 9B
C75C:D0 FA 60 A8 AD 37 C7 F0 74
C764:05 C0 2B B0 01 60 A9 13 07
C76C:20 D2 FF A2 27 20 56 C7 19
C774:84 0B A0 5D 20 82 C7 A4 32
C77C:0B A9 12 20 D2 FF B9 94 DE
C784:C7 F0 07 20 D2 FF C8 4C 8B
C78C:82 C7 A9 92 20 D2 FF 60 5A
C794:53 43 52 45 45 4E 20 4E 2F
C79C:41 4D 45 3F 00 46 49 4C B4
C7A4:45 20 4E 41 4D 45 3F 00 BA
C7AC:54 41 50 45 20 4F 52 20 18
C7B4:44 49 53 4B 20 28 54 2F 51
C7BC:44 29 00 4D 45 4D 4F 52 DD
C7C4:59 20 49 53 20 46 55 4C 78
C7CC:4C 00 49 4E 44 45 58 20 98
C7D4:49 53 20 46 55 4C 4C 00 BA
C7DC:4E 41 4D 45 20 4E 4F 54 0F
C7E4:20 46 4F 55 4E 44 00 45 1E
C7EC:52 52 4F 52 00 14 20 9D 77
C7F4:94 20 1D 91 00 20 1D C8 17
C7FC:AD FE C0 85 04 AD FF C0 2B
C804:85 05 A9 04 A0 C1 A2 00 60
C80C:20 D8 FF 4C E9 C1 20 1D 5C
C814:C8 A9 00 20 D5 FF 4C E9 A7
C81C:C1 A9 0D 20 5F C7 A0 10 08
C824:20 0A C2 98 A2 00 A0 C1 42
C82C:20 BD FF A9 18 20 5F C7 9F
C834:A2 10 A0 11 20 0C C2 AD A4
C83C:10 C1 C9 44 D0 04 A2 08 A7
C844:D0 06 C9 54 D0 0A A2 01 33
C84C:A9 02 A0 01 20 BA FF 60 A3
C854:68 68 A9 57 20 5F C7 4C 39
C85C:E9 C1 A0 00 20 F1 C2 AD 63
C864:63 C1 D0 03 20 4E C1 A9 C9
C86C:36 85 01 8D 37 C7 8D 36 9D
C874:C7 BA 8E 16 C1 A0 00 8C E8
C87C:01 C1 20 73 00 8D 37 C7 A6
C884:C9 49 F0 55 85 0D 20 73 D4
C88C:00 20 73 00 F0 C4 99 00 62
C894:C1 C8 20 73 00 F0 10 C9 22
C89C:2C F0 0C C9 24 F0 63 99 E3
C8A4:00 C1 C0 0F D0 EB C8 A9 21
C8AC:20 99 00 C1 8C 14 C1 A5 AE
C8B4:0D C9 4C F0 2C C9 53 F0 F7
C8BC:28 20 FB C8 A5 0D C9 43 AE
C8C4:F0 0B C9 44 F0 0A C9 50 A2
C8CC:F0 09 4C 56 C8 4C 0D C3 5D
C8D4:4C DF C5 4C C0 C4 4C F9 AD
C8DC:C7 20 73 00 D0 FB 4C F2 CA
C8E4:C6 20 73 00 C9 2C F0 F9 2B
C8EC:8D 10 C1 20 FB C8 A5 0D DE
C8F4:C9 4C D0 E2 4C 12 C8 20 23
C8FC:79 00 F0 05 20 73 00 D0 59
C904:FB 60 A5 2D 85 02 A5 2E E2
C90C:85 03 A0 00 AD 01 C1 09 35
C914:80 8D 01 C1 AE 00 C1 8A 0B
C91C:D1 02 F0 06 20 60 C9 4C F9
C924:1B C9 C8 AD 01 C1 D1 02 60
C92C:F0 07 88 20 60 C9 4C 1B EA
C934:C9 C8 B1 02 C9 10 90 02 E6
C93C:A9 0F 8D 14 C1 AA CA C8 72
C944:B1 02 85 04 C8 B1 02 85 B8
C94C:05 A0 00 B1 04 99 00 C1 ED
C954:C8 CA 10 F7 A9 20 99 00 81
C95C:C1 4C AB C8 18 A5 02 69 AA
C964:07 85 02 A5 03 69 00 85 BA
C96C:03 C5 30 B0 01 60 A5 02 DA
C974:C5 2F B0 01 60 4C 54 C8 82
C97C:00 00 00 00 00 00 00 00 10
```

Memo Diary

Article on page 37.

```
CE 90 F=(1=1)
QM 100 GOSUB 2250
RG 110 DATA JAN,FEB,MAR,APR,MA
      Y,JUN
DR 120 DATA JUL,AUG,SEP,OCT,NO
      V,DEC
CB 130 DATA SUNDAY,MONDAY,TUES
      DAY,WEDNESDAY
ES 140 DATA THURSDAY,FRIDAY,SA
      TURDAY
CF 150 DIM M$(12),W$(6),L$(100
      )
BK 160 FOR J=1 TO 12
AC 170 READ M$(J)
KC 180 NEXT J
GE 190 FOR J=0 TO 6
BJ 200 READ W$(J)
JD 210 NEXT J
DB 220 PRINT "EVENT CALENDAR"
SA 230 IF F=0 THEN 260
JH 240 C=1
PD 250 GOSUB 3010
JC 255 IF E=0 THEN 260
CM 256 F=0
DH 260 PRINT "TODAY'S DATE:"
FF 270 Y8=Y9
PM 280 GOSUB 1670
DC 290 M8=M
GQ 300 D8=D
PJ 310 IF M8>=M9 THEN 330
JX 320 Y8=Y9+1
AJ 330 IF M8<>M9 OR D8>=D9 THE
      N 350
DA 340 Y8=Y9+1
JS 350 IF Y8<=Y9 THEN 370
HP 360 PRINT "HAPPY NEW YEAR"
RM 370 IF F THEN 400
PA 380 PRINT "YEAR";
JK 390 INPUT Y8
HD 400 D9$=RIGHT$(STR$(100+M9)
      ,2)+"/"
JC 410 D9$=D9$+RIGHT$(STR$(100
      +D9),2)
PA 420 IF F THEN 440
FA 430 D9$=D8$
DM 440 F=(1=1)
FG 450 GOSUB 1960
CR 460 PRINT "PAST EVENTS: ";
FJ 470 IF L0>=0 THEN 500
DB 480 PRINT "NONE"
MF 490 GOTO 650
KA 500 PRINT L0+1
EC 510 GOSUB 4010
SK 520 F9=-1
RA 530 FOR J=0 TO L0
BE 540 IF MID$(L$(J),6,1)="/"
      {SPACE}THEN 570
QF 550 L$(L9)=L$(J)
PG 560 L9=L9+1
GM 570 NEXT J
SD 580 L8=L0+1
XX 590 FOR J=L8 TO L9-1
CS 600 L$(J-L8)=L$(J)
PQ 610 NEXT J
AD 620 L9=L9-L8
AM 630 L8=0
XG 640 L=L9
EC 650 F=0
BP 660 F9=0
QA 670 D9$=D8$
FR 680 L=L9-L8
JQ 690 IF L<>0 THEN 710
HC 700 PRINT "NO FUTURE EVENTS
      "
QA 710 IF L=0 THEN 730
PE 720 PRINT L;" FUTURE EVENTS
      "
EF 730 PRINT
FQ 740 PRINT "1. SEE FUTURE EV
      ENTS"
RB 750 PRINT "2. ADD NEW EVENT
      "
PS 760 PRINT "3. CANCEL EVENT"
RH 770 PRINT "4. SEARCH FOR EV
      ENT"
HX 780 PRINT "5. QUIT"
QM 790 PRINT
KF 800 PRINT "...YOUR CHOICE (
      1-5)";
MA 810 INPUT A
XP 820 PRINT
SM 830 ON A GOTO 850,940,1210,
      1450,1570
DP 840 GOTO 730
SM 850 PRINT "AHEAD TO DATE:"
PS 860 GOSUB 1670
PA 870 GOSUB 1960
CG 880 IF L0<>-1 THEN 910
XA 890 PRINT "NO EVENTS"
XQ 900 GOTO 920
BR 910 GOSUB 4010
XK 920 PRINT L9-L0-1;" OTHER F
      UTURE EVENTS"
QS 930 GOTO 730
SR 940 PRINT "ANNUAL OR ONE-TI
      ME (A/O)"
QJ 950 INPUT P$
CE 960 A=0
MA 970 P$=LEFT$(P$,1)
QG 980 IF P$="O" THEN 1010
QH 990 A=1
HK 1000 IF P$<>"A" THEN 730
PJ 1010 GOSUB 1670
DP 1020 Y$="/"+RIGHT$(STR$(101
      +Y8),2)
AA 1030 IF D8$<=D9$ THEN 1050
DP 1040 Y$="/"+RIGHT$(STR$(100
```

```
        +Y8),2)
RM 1050 IF A<>1 THEN 1070
FK 1060 Y$=""
QS 1070 GOSUB 1960
JQ 1080 IF L9-1<L0+1 THEN 1120
PC 1090 FOR J=L9-1 TO L0+1 STE
        P -1
XF 1100 L$(J+1)=L$(J)
DB 1110 NEXT J
MQ 1120 PRINT "DETAIL";
GF 1130 INPUT LL$
GK 1140 D8$=D8$+Y$
RJ 1150 D8$=D8$+" "
SK 1160 L$(L0+1)=D8$+LL$
CC 1170 L9=L9+1
AF 1180 L=L9
MC 1190 F9=-1
JM 1200 GOTO 680
CQ 1210 PRINT "CHANGE WHICH DA
        TE:"
BH 1220 GOSUB 1670
MC 1230 L0=-1
QP 1240 FOR J=L8 TO L9-1
ME 1250 IF D8$<>LEFT$(L$(J),5)
         THEN 1300
RJ 1260 L1=J
BA 1270 IF L0<>-1 THEN 1290
KK 1280 L0=J
KQ 1290 PRINT J;": ";L$(J)
AR 1300 NEXT J
ED 1310 IF L0<>-1 THEN 1340
BB 120 PRINT "NO EVENTS"
FD 1330 GOTO 730
QB 1340 PRINT
CM 1350 PRINT " DELETE WHICH E
        VENT ABOVE";
EX 1360 INPUT A
HF 1370 IF A<L0 OR A>L1 THEN 7
        30
CX 1380 FOR J=A TO L9-1
BH 1390 L$(J)=L$(J+1)
FD 1400 NEXT J
RD 1410 L9=L9-1
FX 1420 F9=-1
SC 1430 PRINT " ... DELETED"
KM 1440 GOTO 680
DE 1450 PRINT "SEARCH FOR";
XH 1460 INPUT P$
DG 1470 P=LEN(P$)
GP 1480 FOR J=0 TO L9-1
GA 1490 A=7
JS 1500 IF MID$(L$(J),6,1)<>"/
        " THEN 1520
DF 1510 A=10
PH 1520 IF A+P-1>LEN(L$(J)) OR
         P$<>MID$(L$(J),A,P) T
        HEN 1540
SC 1530 PRINT L$(J)
DR 1540 NEXT J
KQ 1550 PRINT "{4 SPACES}END O
        F SEARCH"
FA 1560 GOTO 730
HM 1570 IF F9<>0 THEN 1590
HB 1575 IFOTHENCLOSE15
FD 1580 END
XK 1590 PRINT "READY TO WRITE
        {SPACE}NEW EVENTS FILE
         (Y/N)"
BX 1600 INPUT P$
DH 1610 IF LEFT$(P$,1)="Y" THE
        N 1630
HK 1620 STOP
GE 1630 D9$=D9$+"/"
QK 1640 D9$=D9$+RIGHT$(STR$(Y8
        +100),2)
FJ 1650 C=2
MK 1660 GOTO 3010
PQ 1670 M=0
CP 1680 PRINT "MONTH";
PM 1690 INPUT MM$
CA 1700 M=VAL(MM$)
FG 1710 MM$=LEFT$(MM$+"XX",3)
BF 1720 IF M=0 THEN 1760
PD 1730 IF M<1 OR M>12 THEN 16
        70
PB 1740 PRINT M$(M)
AA 1750 GOTO 1810
SB 1760 FOR J=1 TO 12
MQ 1770 IF MM$<>M$(J) THEN 179
        0
RX 1780 M=J
RR 1790 NEXT J
MX 1800 IF M<1 OR M>12 THEN 16
        70
CC 1810 PRINT "DAY";
SR 1820 INPUT D
DH 1830 IF D<1 OR D>31 THEN 16
        70
FG 1840 D8$=RIGHT$(STR$(100+M)
        ,2)+"/"
BJ 1850 D8$=D8$+RIGHT$(STR$(10
        0+D),2)
BP 1860 Y=Y8
EF 1870 IF D8$>D9$ THEN 1890
QD 1880 Y=Y8+1
KX 1890 GOSUB 2150
HH 1900 IF LEN(LL$)<=0 THEN 19
        20
QH 1910 PRINT "(";W$(W);")"
EJ 1920 RETURN
HK 1930 C=1
FQ 1940 GOSUB 3010
RM 1950 RETURN
GD 1960 LL$=CHR$(255)
DD 1970 L0=-1
KC 1980 IF L<>0 THEN 2000
KX 1990 RETURN
SP 2000 V$=D8$+LL$
QK 2010 WW$=D9$
JA 2020 IF F<>0 THEN 2040
MS 2030 WW$=D9$+LL$
BK 2040 F1=(WW$>=V$)
XS 2050 FOR J=L8 TO L9-1
XC 2060 F2=(L$(J)>WW$)
EX 2070 F3=(V$>L$(J))
AJ 2080 F0=F2 AND F3
XD 2090 IF F1=0 THEN 2110
AP 2100 F0=F2 OR F3
HB 2110 IF F0=0 THEN 2130
KR 2120 L0=J
AB 2130 NEXT J
SH 2140 RETURN
CB 2150 IF Y>=85 THEN 2170
GB 2160 Y=Y+100
SJ 2170 M1=M+1
JH 2180 M2=INT(1/M1+.7)
JG 2190 M3=Y-M2
JE 2200 M4=M1+12*M2
DH 2210 N=INT(M4*30.6001)+INT(
        M3*365.25)+D
GM 2220 M6=INT(N/7)
AM 2230 W=N-7*M6
ER 2240 RETURN
EJ 2250 PRINT CHR$(147)CHR$(5)
SX 2260 RETURN
AC 3000 REM INPUT/OUTPUT ROUTI
        NE
ES 3010 F$="EVENTS"
PM 3020 PRINT"DISK OR CASSETTE
         (D/C)?"
EG 3030 GETA$:IF((A$<>"C")AND(
        A$<>"D"))ORA$=""THEN30
        30
DF 3040 IFA$="D"THEN3060
FF 3050 D1=0:G$="":GOTO3070
HB 3060 F$="@0:"+F$:D1=1
BG 3070 IFC=2THEN3160
DR 3080 IFD1=1THENG$=",S,R"
XJ 3090 OPEN1,1+7*D1,8*D1,F$+G
        $:GOSUB3220:IFETHENCLO
        SE1:GOTO3150
FC 3100 INPUT#1,LL$:IF LEN(LL$
        )<>8 THEN PRINT LL$;"?
        ":GOTO 3140
XD 3110 M=VAL(LEFT$(LL$,2)):D=
        VAL(MID$(LL$,4,2)):Y0=
        VAL(MID$(LL$,7,2))
QE 3120 M9=M:D9=D:Y9=Y0:L=0:PR
        INT "LAST ACCESS: ";LL
        $
ES 3130 INPUT#1,L$(L):L=L+1:IF
         ST=0 THEN 3130
PH 3140 CLOSE1:GOSUB3220
MC 3150 L8=0:L9=L:RETURN
SB 3160 IFD1=1THENG$=",S,W"
RD 3170 OPEN1,1+7*D1,1+7*D1,F$
        +G$:GOSUB3220:IFETHENC
        LOSE1:CLOSE15:END
FB 3180 PRINT#1,D9$;CHR$(13);
GG 3190 FORJ=0TOL9-1:PRINT#1,L
        $(J);CHR$(13);:NEXTJ
XA 3200 GOSUB3220:CLOSE1:GOSUB
        3220:IFOTHENCLOSE15
SG 3210 END
SR 3220 IFD1=0THENRETURN
AB 3230 IFO=0THENOPEN15,8,15:O
        =1
RM 3240 INPUT#15,E,B$:IFETHENP
        RINTB$:CLOSE15:O=0
SR 3250 RETURN
AQ 4000 REM PRINT ROUTINE
DX 4010 D$="":P=3
GP 4020 INPUT "WANT EVENTS ON
        {SPACE}PRINTER (Y/N)";
        P$
FX 4030 IF LEFT$(P$,1)<>"Y" TH
        EN 4050
ED 4040 P=4
GF 4050 OPEN 3,P
JD 4060 FOR J=L8 TO L0
RP 4070 IF D$=LEFT$(L$(J),5) T
        HEN 4150
MQ 4080 D$=LEFT$(L$(J),5)
GM 4090 M=VAL(LEFT$(D$,2))
JG 4100 D=VAL(MID$(D$,4,2))
JM 4110 Y=Y8:IF D$<=D9$ THEN Y
        =Y8+1
SF 4120 GOSUB 2150
XD 4130 PRINT#3,W$(W);" ";
PM 4140 PRINT#3,M$(M);D
QC 4150 PRINT#3,"{3 SPACES}";M
        ID$(L$(J),6)
CS 4160 NEXT J
GK 4170 CLOSE 3
QE 4180 RETURN
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Solarpix

Article on page 21.

```
XA 10 POKE56,132:CLR
FG 20 POKE792,193:POKE648,132
RC 30 POKE56576,(PEEK(56576)AN
      D252)OR1
XR 40 POKE53280,3:POKE53281,3:
      DIMS(255),C(255),M$(9,17
      )
PA 50 BP=34808:V=53248:LC=4915
      2:EY=1984.82
CB 60 PP=2*↑:P$(0)="AC":P$(1)=
      "DE":SA=1.1
RA 70 PRINT"{CLR}{8 DOWN}{BLU}
      "SPC(14)"PLEASE WAIT"
CB 80 SP$=CHR$(32):X$=SP$+"Z=E
      XIT"
```

```
KC 90 RE$=CHR$(18):RX$=CHR$(14
      6)
XQ 100 T$(1)=RE$+"SUN & PLANET
      S"+RX$
FF 110 T$(2)=RE$+"SUN,EARTH &
      {SPACE}MOON"+RX$
RD 120 T$(3)=RE$+"COMETS"+RX$
PQ 130 T$(4)=RE$+"FACTS"+RX$
SH 140 FORJ=49152TO49240:READK
      :POKEJ,K:NEXT
AF 150 FORJ=34816TO35199:READK
      :POKEJ,K:NEXT
KK 160 FORJ=35520TO35775:POKEJ
      ,0:NEXT
PX 170 FORL=1TO10:READJ,K:POKE
      35584+J,K:NEXT
SJ 180 FORJ=0TO7:READK:POKEBP+
      J,K:NEXT
BH 190 FORJ=0TO7:READH(J):NEXT
BC 200 FORJ=0TO10:READF$(J),U$
      (J)
HQ 210 L=0:H=10:IFJ>4THENL=1:H
      =9
EB 220 FORK=LTOH:READZ(K,J):NE
      XT:NEXT
DQ 230 FORJ=1TO9:D(J)=Z(J,5)
BB 240 P(J)=Z(J,6):NEXT
XE 250 FORJ=0TO10:READN$(J)
DG 260 IFJ<4ORJ=10THEN280
HF 270 FORK=1TOZ(J,10):READM$(
      J,K):NEXT
CE 280 NEXT:FORJ=1TO9:READR:R(
      J)=R/360:NEXT
SJ 290 FORN=0TO255:M=N*2*↑/256
RJ 300 S(N)=SIN(M):C(N)=COS(M)
      :NEXT
RR 310 FORJ=32TO36:K=74-J
BA 320 POKE251,J:POKE252,2:POK
      E253,K
AK 330 POKE254,2:SYS49175:NEXT
MF 340 FORJ=0TO7:POKEV+39+J,H(
      J):NEXT
BH 350 POKEV+23,0:POKEV+29,0:P
      OKEV+16,128
QQ 360 GOSUB550:POKEV+39,7:POK
      EV+40,3
HK 370 POKEV+32,3:POKEV+33,3:P
      OKE646,6
EG 380 PRINT"{CLR}":SYSLC,4,14
KQ 390 PRINT"SOLARPIX":PRINT:P
      RINT
KC 400 FORJ=1TO4:PRINTTAB(8)J;
      T$(J):PRINT
GD 410 NEXT:PRINTTAB(9)"Q QUIT
      "
DR 420 GOSUB470:ONKGOTO630,101
      0,1280,1460
MB 430 IFK$<>"Q"THEN420
MM 440 POKE56576,PEEK(56576)OR
      3
MG 450 POKE648,4:POKE792,71:GO
      SUB550
KE 460 PRINT"{CLR}":POKE56,160
      :CLR:END
GE 470 POKE198,0
RR 480 GETK$:IFK$=""THEN480
HF 490 K=VAL(K$):RETURN
BM 500 SYSLC,18,31:PRINT"G=GO"
BC 510 PRINTTAB(31)"S=STOP"
PH 520 PRINTTAB(31)"+=FASTER"
ME 530 PRINTTAB(31)"-=SLOWER"
RS 540 PRINTTAB(30)X$:RETURN
PX 550 POKEV+21,0:FORJ=0TO15
JK 560 POKEV+J,0:NEXT:RETURN
DH 570 SYSLC,23,30:PRINTX$
GA 580 SYSLC,2,0:RETURN
FR 590 GETK$:IFK$="S"THENFL=1
PH 600 IFK$="G"THENFL=0
QJ 610 IFK$="-"THENM=M/2
XR 620 RETURN
JA 630 POKEV+32,11:POKEV+33,0:
      POKE646,7
KR 640 GOSUB550:PRINT"{CLR}"T$
      (1):GOSUB570
HG 650 PRINT"{7}PLEASE SELECT
      {SPACE}PLANETS (MAX5)"
FF 660 PRINT:FORJ=1TO9:PRINTJ;
      N$(J):NEXT
DG 670 PRINT:PRINT"INNER PLANE
      T?";
CX 680 GOSUB470:IFK$="Z"THEN36
      0
QF 690 L=K:IFL<1THEN680
EG 700 PRINTN$(L):PRINT"OUTER
      {SPACE}PLANET?";
RK 710 GOSUB470:IFK$="Z"THEN36
      0
DE 720 H=K:IFH<LORH>9ORH-L>4TH
      EN710
FR 730 PRINTN$(H):PRINT:X=1
MH 740 FORJ=1TOH-L+1:X=X+2↑J:N
      EXT
MK 750 FORJ=LTOH:F(J)=D(J)*90/
      D(H)
RR 760 E(J)=F(J)*SA:NEXT:M=P(L
      )/50:ML=M*9
SR 770 INPUT"STARTING YEAR (1-
      2000)";K$
FC 780 SY=VAL(K$):IFSY<1ORSY>2
      000THEN630
DJ 790 PRINT"{CLR}{YEL}"T$(1):
      K=1
AF 800 FORJ=LTOH:POKE646,H(K)
RF 810 PRINTTAB(31)N$(J):K=K+1
      :NEXT
DJ 820 POKE646,14:GOSUB500:SYS
      LC,7,31
AX 830 PRINT"EARTH":PRINTTAB(3
      1)"YEAR"
PE 840 E=.25:T=SY-EY:FL=1
RS 850 POKEV,130:POKEV+1,148:P
      OKEV+21,X
BH 860 FORJ=LTOH:A=T/P(J)+R(J)
BD 870 A=INT((A-INT(A))*256):I
      FJ=9THEN890
AD 880 X(J)=C(A):Y(J)=-S(A):GO
      TO910
MB 890 Z=1+E*C(A):X(J)=E+(E+C(
      A))/Z
FK 900 Y(J)=(E*E-1)*S(A)/Z
XE 910 X(J)=INT(131+X(J)*E(J))
MQ 920 Y(J)=INT(149+Y(J)*F(J))
      :NEXT
JC 930 K=1:FORJ=LTOH:POKEV+2*K
      ,X(J)
ER 940 POKEV+2*K+1,Y(J):K=K+1:
      NEXT
JB 950 T$=STR$(INT((T+EY)*10)/
      10)
BA 960 SYSLC,9,30:PRINTT$SP$SP
      $
CP 970 GOSUB590:IFK$="Z"THEN63
      0
QJ 980 IFFLTHEN970
HK 990 IFK$="+"ANDM<MLTHENM=M*
      2
RJ 1000 T=T+M:GOTO860
BQ 1010 POKEV+32,14:POKEV+33,0
      :POKE646,14
SM 1020 PRINT"{CLR}"T$(2):PRIN
      TTAB(31)"MOON"
AQ 1030 PRINTTAB(31)"PHASE":GO
      SUB500
BR 1040 SYSLC,24,1:PRINT"(EART
      H-MOON DISTANCE MAGNIF
      IED BY 30)";
CP 1050 SYSLC,7,31:PRINT"EARTH
      "
EQ 1060 PRINTTAB(31)"DAY"
GP 1070 E=84:D=E*SA:F=D*30/390
      :G=F/SA
BF 1080 M=0.3/365.25:T=0:H=13.
      3685
EB 1090 POKEV+39,14:POKEV+40,1
      2
SF 1100 POKEV+12,141:POKEV+13,
      147
HS 1110 POKEV+14,24:POKEV+15,8
      0
PS 1120 POKEV+21,195:FL=1
DG 1130 A=T-INT(T):B=T*H:B=B-I
      NT(B)
CQ 1140 C=INT(12*(B-A)+.5-7)
RX 1150 IFC<0THENC=C+12:GOTO11
      50
DG 1160 A=A*PP:B=B*PP
XB 1170 X=INT(142.5+COS(A)*D)
RH 1180 Y=INT(148.5-SIN(A)*E)
KJ 1190 W=INT(X+1+COS(B)*F)
GA 1200 Z=INT(Y+1-SIN(B)*G)
MD 1210 POKEV,X:POKEV+1,Y:POKE
      V+2,W
EP 1220 POKEV+3,Z:POKEBP+7,32+
      C
JR 1230 SYSLC,9,30:PRINTINT(T*
      365.25)
GR 1240 GOSUB590:IFK$="Z"THEN3
      60
QX 1250 IFFLTHEN1240
PB 1260 IFK$="+"ANDM<.006THENM
      =M*2
GJ 1270 T=T+M:GOTO1130
GD 1280 POKEV+32,5:POKEV+33,0:
      POKE646,5
CR 1290 GOSUB550:PRINT"{CLR}"R
      E$T$(3):GOSUB570
SB 1300 INPUT"ECCENTRICITY(0-0
      .96)";K$
HJ 1310 E=VAL(K$):IFK$="Z"THEN
      360
KB 1320 IFE<0ORE>.96ORLEN(K$)=
      0THEN1280
BE 1330 FL=1:IFRIGHT$(K$,1)="R
      "THENFL=-1
BS 1340 PRINT"{CLR}"RE$T$(3):G
      OSUB570
FF 1350 PRINT"ECCENTRICITY="E:
      SYSLC,2,27
JS 1360 IFETHENPRINT"ACCELERAT
      ING"
HC 1370 M=0.01:T=0:F=65*SA:G=6
      5*FL
QK 1380 POKEV,106:POKEV+1,148:
      POKEV+21,3
MG 1390 A=(T-INT(T))*PP:Z=1+E*
      COS(A)
RC 1400 X=E+(E+COS(A))/Z:Y=(E*
      E-1)*SIN(A)/Z
AG 1410 X=INT(107+X*F):Y=INT(1
      49+Y*G)
FD 1420 J=INT(A/↑):POKEV+2,X:P
      OKEV+3,Y
JQ 1430 IFE>0.1THENSYSLC,2,27:
      PRINTP$(J)
FX 1440 GETK$:IFK$="Z"THEN1280
XQ 1450 T=T+M:GOTO1390
XA 1460 POKEV+32,3:POKEV+33,6:
      POKE646,3
CD 1470 PRINT"{CLR}"RE$T$(4)"
      {SPACE}MENU":SYSLC,4,0
MS 1480 GOSUB570:SYSLC,4,0
JK 1490 PRINT" S SUN":PRINT" M
       MOON"
SJ 1500 FORJ=1TO9:PRINTJ;N$(J)
      :NEXT
FS 1510 SYSLC,4,0:FORJ=0TO10
BC 1520 PRINTTAB(20)CHR$(J+65)
      " "F$(J):NEXT
CR 1530 GOSUB470:IFK$="Z"THEN3
      60
QD 1540 IFK$="S"THENK=0:GOTO15
      90
GE 1550 IFK$="M"THENK=10:GOTO1
      590
EM 1560 IFK>0ANDK<10THEN1590
```

```
RS 1570 K=ASC(K$)-65:IFK<0ORK>
        10THEN1530
RG 1580 GOTO1720
FM 1590 POKEV+32,3:POKEV+33,11
        :POKE646,3
QG 1600 PRINT"{CLR}"RE$N$(K)RX
        $;
AG 1610 H=4:IFK=0ORK=10THENPRI
        NT:GOTO1630
SB 1620 H=10:PRINT" PLANET #"K
        "FROM SUN"
PE 1630 PRINT:FORJ=0TOH
HJ 1640 PRINTF$(J)TAB(18)Z(K,J
        )TAB(28)U$(J)
GR 1650 NEXT:IFK<10THEN1680
FS 1660 PRINT"DIST.FROM EARTH"
        TAB(19)"243000 MILES"
DH 1670 PRINTF$(9)TAB(19)"6 MO
        NTHS"
SE 1680 PRINT:IFK<4ORK=10THENJ
        770
CQ 1690 FORJ=1TOZ(K,10):C=INT(
        (J-1)/7)
PE 1700 L=13+J-C*7:SYSLC,L,C*1
        3
BD 1710 PRINTM$(K,J):NEXT:GOTO
        1770
MR 1720 POKEV+32,3:POKEV+33,11
        :POKE646,3
BK 1730 PRINT"{CLR}"RE$F$(K)RX
        $TAB(19)U$(K)
CP 1740 L=0:H=10:IFK>4THENL=1:
        H=9
AR 1750 SYSLC,5,0:FORJ=LTOH
KD 1760 PRINTN$(J)TAB(18)Z(J,K
        ):NEXT
RQ 1770 GOSUB570:GOSUB470:GOTO
        1460
XJ 1780 DATA 32,13,192,132,251
        ,32,13
HG 1790 DATA 192,166,251,76,24
        0,255,32
EM 1800 DATA 253,174,32,158,17
        3,32,170
AH 1810 DATA 177,96,162,6,6,25
        1,38
BA 1820 DATA 252,6,253,38,254,
        202,208
MH 1830 DATA 245,169,21,133,2,
        160,2
CC 1840 DATA 177,251,162,8,106
        ,38,255
ME 1850 DATA 202,208,250,192,1
        ,240,4
QX 1860 DATA 152,73,2,168,165,
        255,145
PK 1870 DATA 253,192,1,240,4,1
        52,73
PH 1880 DATA 2,168,136,16,223,
        162,3
QR 1890 DATA 230,251,230,253,2
        02,208,249
KS 1900 DATA 198,2,208,208,96
KS 1910 DATA 0,15,0,0,0,192,0,
        0,48
JG 1920 DATA 0,0,56,0,0,28,0,0
        ,28
FP 1930 DATA 0,0,14,0,0,14,0,0
        ,7
GC 1940 DATA 0,0,7,0,0,7,0,0,7
GF 1950 DATA 0,0,7,0,0,14,0,0,
        14
GR 1960 DATA 0,0,28,0,0,28,0,0
        ,56
JX 1970 DATA 0,0,48,0,0,192,0,
        15,0,0
QM 1980 DATA 0,15,0,0,7,192,0,
        3,240
GR 1990 DATA 0,1,248,0,1,252,0
        ,1,252
ER 2000 DATA 0,0,254,0,0,254,0
        ,0,255
BJ 2010 DATA 0,0,255,0,0,255,0
        ,0,255
MR 2020 DATA 0,0,255,0,0,254,0
        ,0,254
SG 2030 DATA 0,1,252,0,1,252,0
        ,1,248
KH 2040 DATA 0,3,240,0,7,192,0
        ,15,0,0
HH 2050 DATA 0,15,0,0,15,192,0
        ,15,240
RQ 2060 DATA 0,15,248,0,15,252
        ,0,15,252
GF 2070 DATA 0,15,254,0,15,254
        ,0,15,255
QB 2080 DATA 0,15,255,0,15,255
        ,0,15,255
SE 2090 DATA 0,15,255,0,15,254
        ,0,15,254
EF 2100 DATA 0,15,252,0,15,252
        ,0,15,248
SJ 2110 DATA 0,15,240,0,15,192
        ,0,15,0,0
BB 2120 DATA 0,15,0,0,31,192,0
        ,63,240
SX 2130 DATA 0,127,248,0,127,2
        52,0,127,252
MX 2140 DATA 0,255,254,0,255,2
        54,0,255,255
DD 2150 DATA 0,255,255,0,255,2
        55,0,255,255
QX 2160 DATA 0,255,255,0,255,2
        54,0,255,254
FK 2170 DATA 0,127,252,0,127,2
        52,0,127,248
XD 2180 DATA 0,63,240,0,31,192
        ,0,15,0,0
FR 2190 DATA 0,31,0,0,255,192,
        3,255,240
HA 2200 DATA 3,255,248,7,255,2
        52,7,255,252
HP 2210 DATA 15,255,254,15,255
        ,254,31,255,255
RP 2220 DATA 31,255,255,31,255
        ,255,31,255,255
HS 2230 DATA 31,255,255,15,255
        ,254,15,255,254
XE 2240 DATA 7,255,252,7,255,2
        52,3,255,248
CA 2250 DATA 3,255,240,0,255,1
        92,0,31,0,0
KQ 2260 DATA 0,255,0,3,255,192
        ,15,255,240
PB 2270 DATA 31,255,248,63,255
        ,252,63,255,252
XR 2280 DATA 127,255,254,127,2
        55,254,255,255,255
AF 2290 DATA 255,255,255,255,2
        55,255,255,255,255
CB 2300 DATA 255,255,255,127,2
        55,254,127,255,254
DG 2310 DATA 63,255,252,63,255
        ,252,31,255,248
RR 2320 DATA 15,255,240,3,255,
        192,0,255,0,0
CK 2330 DATA 0,112,3,248,6,248
        ,9,248,12,112
BR 2340 DATA 64,192,67,192
BR 2350 DATA 128,224,131,224,1
        34,224
SE 2360 DATA 46,45,45,45,45,45
        ,44,37
FR 2370 DATA 7,3,4,14,8,6,7,12
AK 2380 DATA DIAMETER,MILES
FH 2390 DATA 864000,3100,7700,
        7926
RC 2400 DATA 4200,88000,71000,
        32000
GA 2410 DATA 31000,1500,2160
GJ 2420 DATA MASS,(EARTH=1)
MK 2430 DATA 332830,0.055,0.81
        5,1
HF 2440 DATA 0.107,318,95.2,14
        .6
CM 2450 DATA 17.2,0.01,0.012
SA 2460 DATA DENSITY,(WATER=1)
SD 2470 DATA 1.4,5.4,5.2,5.5,3
        .9,1.3
BD 2480 DATA 0.7,1.2,1.7,1.4,3
        .4
EE 2490 DATA ROTATION PERIOD,E
        ARTHDAYS
QS 2500 DATA 25,58.9,243,1,1.0
        3,0.404
MJ 2510 DATA 0.444,0.72,0.77,6
        .39,27.3
KK 2520 DATA TEMPERATURE,FAHRE
        NHEIT
AJ 2530 DATA 10000,620,900,72,
        -10,-240
BM 2540 DATA -300,-340,-370,-4
        00,-10
FE 2550 DATA DISTANCE FROM SUN
        ,MILL.MILES
SP 2560 DATA 36,67.2,92.9,142,
        483
BK 2570 DATA 887,1783,2795,367
        0
GK 2580 DATA LENGTH OF YEAR,EA
        RTHYEARS
EX 2590 DATA 0.241,0.615,1,1.8
        8,11.86
PR 2600 DATA 29.46,84,164,248
BK 2610 DATA ORBIT SPEED,MPH
XJ 2620 DATA 107000,78000,6600
        0,54000
QD 2630 DATA 29000,22000,15000
        ,12000,10000
XG 2640 DATA AXIS TILT,DEGREES
AJ 2650 DATA 0,179,23.5,25,3.1
FD 2660 DATA 26.7,97.9,28.8,0
XQ 2670 DATA DRIVETIME(55MPH),
        YEARS
JA 2680 DATA 75,139,193,295,10
        02
MM 2690 DATA 1840,3698,5797,76
        12
RS 2700 DATA MOONS,-
CE 2710 DATA 0,0,1,2,16,17,15,
        3,1
XC 2720 DATA SUN,MERCURY,VENUS
        ,EARTH,MARS,PHOBOS,DEI
        MOS
DH 2730 DATA JUPITER,METIS,ADR
        ASTREA,AMALTHEA,THEBE,
        IO,EUROPA,GANYMEDE,CAL
        LISTO
JB 2740 DATA LEDA,HIMALIA,LYSI
        THEA,ELARA,ANANKE,CARM
        E,PASIPHAE,SINOPE
FR 2750 DATA SATURN,ATLAS,X,X,
        JANUS,EPIMETHEUS,MIMAS
        ,ENCELADUS,TETHYS,TELE
        STO
MA 2760 DATA CALYPSO,DIONE,X,R
        HEA,TITAN,HYPERION,IAP
        ETUS,PHOEBE
SF 2770 DATA URANUS,MIRANDA,AR
        IEL,UMBRIEL,TITANIA,OB
        ERON,X,X,X,X,X,X,X,X,X
        ,X
JK 2780 DATA NEPTUNE,TRITON,NE
        REID,X,PLUTO,CHARON,MO
        ON
EH 2790 DATA 198,255,351,285,2
        49,180,204,227,173
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Math Dungeon

Article on page 22.

```
RK 100 DIM CM$(22),CM(22),IN$(
       5),IN(5),OB$(13),OB(13)
       ,DS$(40),FL(10,10)
KM 120 SYS65517:CC=PEEK(781)
EQ 130 IFCC=40THENPOKE53281,1
FC 150 GOSUB 700:GOSUB1620
SK 155 PRINT"{CLR}"
KS 160 GOSUB790:CM(2)=1:CM(5)=
       1:FORI=14TO19:CM(I)=1:N
       EXT:OB(1)=1
XK 170 PR$=DS$(1):GOSUB2570:IF
       GTTHENPR$=DS$(2):GOSUB2
       570
GR 180 IFGT=0THENPR$=DS$(3):GO
       SUB2570
KA 190 GOSUB2310:IFPL=20RPL=5T
       HENGOSUB800:GOTO190
XS 200 IFPL=16ANDGT=0THENPR$=D
       S$(3):GOSUB2570:GOTO190
SM 210 IFPL=16THEN280
SD 220 IFPL>16THENGOTO250
DC 230 GOTO170
QC 240 PRINT"{CLR}";:PR$=DS$(4)
       :GOSUB2570
PS 250 GOSUB790:FORI=14TO19:CM
       (I)=1:NEXT:GOSUB2310:IF
       PL=16THEN280
EE 260 IFPL=15THENPR$=DS$(4):G
       OSUB2570
RF 270 GOTO250
KA 280 GOSUB790:PRINT"{CLR}":C
       M(1)=1:CM(3)=1:CM(6)=1:
       FORI=14TO19:CM(I)=1:NEX
       T
MK 290 CM(4)=1
RC 300 PR$=DS$(5):GOSUB2570:IF
       MB=0THENPR$=DS$(6):GOSU
       B2570
CE 310 IFMBTHENPR$=DS$(7):CM(7
       )=1:GOSUB2570:IFNT=1THE
       NPR$=DS$(11):GOSUB 2570
FF 320 IFFD=0THENPR$=DS$(8):NT
       =1:GOSUB2570
FG 330 IFFDTHENPR$=DS$(9):NT=0
       :GOSUB2570
BG 340 GOSUB2310:IFPL=15THEN30
       0
KE 350 IF(PL=18)OR(PL=19)THENG
       OTO300
GM 360 IFPL=17THENGOTO160
MS 370 IF(PL=16)AND(FD=0)THEN3
       20
XE 380 IFPL=16ANDFD=0THEN320
QB 390 IFPL=16THEN450
PK 400 IFPL=1THENPR$=DS$(10):G
       OSUB2570
SX 410 IF(PL=3)OR(PL=6)THENGOS
       UB900:GOTO340
BX 415 IFPL=7ANDNT=0THENPRINT"
       I SEE NO NOTE HERE!":GO
       TO340
DJ 420 IFPL<>7THEN340
SH 430 PR$="THERE IS A QUESTIO
       N ON THE NOTE.":GOSUB25
       70
HS 435 GOSUB1320:IFWR=1THENGOT
       O160
GA 440 PR$=DS$(12):GOSUB2570:P
       R$=DS$(13):GOSUB2570:FD
       =1:MB=0:GOTO340
CG 450 GOSUB790:PRINT"{CLR}
       {7 DOWN}";
EB 470 PR$="A TRAP DOOR OPENS
       {SPACE}BENEATH YOUR FEE
       T AND YOU BEGIN SLIDING
        DOWN A "
BG 480 PR$=PR$+"CHUTE. YOU PAS
       S A NEON SIGN THAT SAYS
        :":GOSUB2570:PRINT
BC 490 IFCC=40THEN505
JA 500 PRINT"{4 SPACES}'MATH D
       UNGEON'":PRINT:PRINT"
       {6 SPACES}'LEVEL "+STR$
       (LV)+"'"
MH 501 PRINT:PRINT:GOTO510
FF 505 PRINT"{11 SPACES}'MATH
       {SPACE}DUNGEON'"
JH 506 PRINT:PRINT"{13 SPACES}
       'LEVEL "+STR$(LV)+"'"
KC 510 PR$="PRESS ANY KEY TO C
       ONTINUE":GOSUB2570:PRIN
       T:PRINT
SQ 520 GETA$:IFA$=""THEN520
CA 530 LT=1:PX=INT(RND(1)*10)+
       1:PY=INT(RND(1)*10)+1:F
       L(PY,PX)=2
XR 540 GOSUB790:CM(9)=1:FORI=1
       2TO19:CM(I)=1:NEXT:CM(9
       )=LT:CM(22)=1
KX 550 IFFL(PY,PX)=2THENPRINT"
       {CLR}":PR$=DS$(14):GOSU
       B2570
MK 560 IFFL(PY,PX)=1THENPRINT"
       {CLR}":PR$=DS$(15):GOSU
       B2570
XX 570 IFLT=1THENPR$=DS$(16):G
       OSUB2570
EJ 580 GOSUB2310
EK 590 IFPL=9THENGOSUB950:GOTO
       580
JG 600 IFPL=12THENGOSUB970:GOT
       O580
GR 610 IFPL=13THENGOSUB1000:GO
       TO580
EA 620 IFPL=15ANDLO=0THEN550
JB 630 IFPL=15ANDLO=1THENGOSUB
       1020:GOTO570
FQ 60 IFPL=14THEN580
HA 650 IFPL=22THENGOSUB2470:GO
       TO670
MD 660 IFPL>15THENGOSUB1560:IF
       (LO=0)OR(WA=1)THEN580
HG 670 ONZGOSUB1860,1860,2000
FD 680 GOTO670
EB 690 END
HX 700 PRINT"{CLR}{9 DOWN}":IF
       CC=40THENPRINTTAB(10);:
       GOTO720
MP 710 PRINTTAB(1);
MM 720 PRINT"{4 SPACES}MATH DU
       NGEON{3 SPACES}":PRINT"
       {DOWN}":IFCC=40THENPRIN
       TTAB(12);:GOTO740
JQ 730 PRINTTAB(3);
PF 740 PRINT"A MATH ADVENTURE"
MM 750 PRINT"{HOME}{21 DOWN}":
       IFCC=40THEN755
GC 751 PRINT"{3 SPACES}ENTER D
       ESIRED{10 SPACES}LEVEL(
       1-5)";:GOTO760
MS 755 PRINT"{3 SPACES}ENTER D
       ESIRED LEVEL(1-5)";
PQ 760 INPUTLV:IFLV<1ORLV>5THE
       NGOTO750
EJ 770 FORI=1TO22:READCM$(I):N
       EXT:FORI=1TO5:READIN$(I
       ):NEXT
AA 780 FORI=1TO29:READDS$(I):N
       EXT:RETURN
DA 790 FORI=1TO22:CM(I)=0:NEXT
       I:IFIN(5)=1THENCM(20)=1
       :RETURN
ER 800 IFPL=2ANDGT=1THENPR$="T
       HE GATE IS ALREADY OPEN
       .":GOSUB2570
MG 810 IFPL=2ANDGT=0THENGT=1:P
       RINT"OK."
DA 820 IFPL=5ANDGT=0THENPR$="T
       HE GATE IS ALREADY CLOS
       ED.":GOSUB2570
BF 830 IFPL=5ANDGT=1THENGT=0:P
       RINT"OK."
CM 840 RETURN
CA 850 IFPL=1ANDFD=1THENPR$="T
       HE DOOR IS ALREADY OPEN
       .":GOSUB2570
KD 860 IFPL=1ANDFD=0THENFD=1:P
       RINT"OK."
XM 870 IFPL=4ANDFD=0THENPR$="T
       HE DOOR IS ALREADY CLOS
       ED.":GOSUB2570
QG 880 IFPL=4ANDFD=1THENFD=0:P
       RINT"OK."
SX 890 RETURN
KG 900 IFPL=3ANDMB=1THENPR$="T
       HE MILKBOX IS ALREADY O
       PEN.":GOSUB2570
PM 910 IFPL=3ANDMB=0THENMB=1:P
       RINT"OK.":CM(7)=1
KS 920 IFPL=6ANDMB=0THENPR$="T
       HE MILKBOX IS ALREADY C
       LOSED.":GOSUB2570
PM 930 IFPL=6ANDMB=1THENMB=0:P
       RINT"OK.":CM(7)=0
GA 940 RETURN
KH 950 IFIN(1)=1THENPR$="YOU A
       LREADY HAVE THE LIGHT."
       :GOSUB2570:RETURN
AB 960 IFLT=1THENLT=0:IN(1)=1:
       PRINT"OK.":CM(9)=0:RETU
       RN
GG 970 IFIN(1)=0THENPR$="YOU D
       O NOT HAVE A LIGHT.":GO
       SUB2570:RETURN
RD 980 IFLO=0THENPRINT"OK.":LO
       =1:RETURN
DJ 990 IFLO=1THENPR$="THE LIGH
       T IS ALREADY ON.":GOSUB
       2570:RETURN
AG 1000 IFLO=0THENPR$="THE LIG
       HT IS ALREADY OFF.":GO
       SUB2570:RETURN
RP 1010 IFLO=1THENPRINT"OK.":L
       O=0:RETURN
AB 1020 PRINT"{CLR}"DS$(16-FL(
       PY,PX))
KD 1030 EC=PX+1:DN$="EAST":IFE
       C=11THENGOTO1050
RF 1040 IFFL(PY,EC)<>0THENGOSU
       B1120
SS 1050 WC=PX-1:DN$="WEST":IFW
       C=0THENGOTO1070
AG 1060 IFFL(PY,WC)<>0THENGOSU
       B1120
FF 1070 NC=PY-1:DN$="NORTH":IF
       NC=0THENGOTO1090
BE 1080 IFFL(NC,PX)<>0THENGOSU
       B1120
HS 1090 SC=PY+1:DN$="SOUTH":IF
       SC=11THENGOTO1110
BH 1100 IFFL(SC,PX)<>0THENGOSU
       B1120
CE 1110 RETURN
FR 1120 PRINTDN$" IS A DOORWAY
       .":RETURN
KG 1130 IFGK=0ANDBK=0ANDRK=0TH
       ENPRINT"WHAT KEY?":RET
       URN
QM 1140 PRINT"OK."
FP 1150 IFGKTHENGK=0:IN(2)=1:R
       ETURN
GH 1160 IFBKTHENBK=0:IN(3)=1:R
       ETURN
GA 1170 IFRKTHENRK=0:IN(4)=1:B
       D=2:RETURN
FM 1180 RETURN
SX 1190 IFGD<>1THEN1220
GS 1200 IFIN(2)=0THENPR$=DS$(1
       0):GOSUB2570:RETURN
DA 1210 IFIN(2)=1THENPR$=DS$(2
       7):GOSUB2570:MA=1:CM(1
       1)=1:CM(1)=0:RETURN
QR 1220 IFBD<>1THEN1250
```

```
CB 1230 IFIN(3)=0THENPR$=DS$(1
        0):GOSUB2570:RETURN
QK 1240 IFIN(3)=1THENPR$=DS$(2
        8):GOSUB2570:CM(8)=1:R
        K=1:CM(1)=0:RETURN
JJ 1250 IFRD<>1THEN1280
DE 1260 IFIN(4)=0THENPR$=DS$(1
        0):GOSUB2570:RETURN
MG 1270 IFIN(4)=1THENPRINT"
        {CLR}":PR$=DS$(29):GOS
        UB2570:GOSUB2520
CA 1280 RETURN
JJ 1290 G=INT(RND(1)*LV*10)+1:
        PR$="YOU HAVE FOUND "+
        STR$(G)+" GOLD PIECES.
        "
MB 1300 GOSUB2570:GP=GP+G:PR$=
        "YOU NOW HAVE "+STR$(G
        P)+" GOLD PIECES.":GOS
        UB2570
QS 1310 CM(10)=0:RETURN
CM 1320 OP=INT(RND(1)*4)+1
SX 1340 IF(LV<=2)AND(OP>=3)THE
        N1320
SD 1350 PRINT"WHAT IS...";
SX 1360 ONOPGOTO1370,1400,1430
        ,1460
ED 1370 X=INT(RND(1)*LV*10):Y=
        INT(RND(1)*LV*10)
SD 1380 IFLV>4THENX=X-INT(RND(
        1)*LV*5):Y=Y-INT(RND(1
        )*LV*5)
HR 1390 Z1=X+Y:PRINTX" + "Y"?"
        :INPUTZ$:GOSUB1510:IFW
        R=2THEN1390
GP 1395 RETURN
KC 1400 X=INT(RND(1)*LV*10):Y=
        INT(RND(1)*LV*10)
SB 1410 IFLV<=3THENIFY>XTHENT=
        X:X=Y:Y=T
RD 1420 Z1=X-Y:PRINTX" - "Y"?"
        :INPUTZ$:GOSUB1510:IFW
        R=2THEN1420
PP 1425 RETURN
GX 1430 X=INT(RND(1)*LV*5):Y=I
        NT(RND(1)*LV*5)
RP 1440 IFLV>=4THENX=X-INT(RND
        (1)*LV*2):Y=Y-INT(RND(
        1)*LV*2)
JP 1450 Z1=X*Y:PRINTX" * "Y"?"
        :INPUTZ$:GOSUB1510:IFW
        R=2THEN1450
AQ 1455 RETURN
RK 1460 X=INT(RND(1)*LV*5):Y=I
        NT(RND(1)*LV*5)
RD 1470 IFLV>=4THENX=X-INT(RND
        (1)*LV*3):Y=Y-INT(RND(
        1)*LV*3)
RR 1480 IFY=0THENY=1
FC 1490 Z1=X/Y:IFZ1<>INT(Z1)TH
        EN1460
PB 1500 PRINTX" / "Y"?":INPUTZ
        $
MG 1505 GOSUB1510:IFWR=2THEN14
        90
MS 1506 RETURN
AA 1510 IFVAL(Z$)<>0THENZ2=VAL
        (Z$):GOTO1540
CP 1520 IFZ$<>"0"THENPR$="ANSW
        ER THE QUESTION FIRST!
        ":GOSUB2570:WR=2:RETUR
        N
EG 1530 Z2=VAL(Z$)
CE 1540 IFZ1=Z2THENPRINT"{CLR}
        CORRECT!!":WR=0:RETURN
CR 1550 PRINT"{CLR}INCORRECT!!
        ":WR=1:RETURN
EF 1560 WA=0:IFLO=0THENPR$="I'
        D TURN ON MY LIGHT FIR
        ST.":GOSUB2570:GOTO161
        0
FF 1570 C=ABS(PL=18)-ABS(PL=19
        ):D=ABS(PL=17)-ABS(PL=
        16):TX=PX+C:TY=PY+D
DB 1580 IFTX=11ORTX=0ORTY=11OR
        TY=0THENPRINT"THERE IS
         A WALL THERE.":WA=1:G
        OTO1610
CQ 1590 IFFL(TY,TX)=0THENPRINT
        "THERE IS A WALL THERE
        .":WA=1:GOTO1610
BR 1600 PX=TX:PY=TY
RS 1610 Z=FL(PY,PX):RETURN
QS 1620 FORI=1TO10:FORJ=1TO10:
        FL(I,J)=0:NEXTJ:NEXTI
BF 1625 NM=INT(RND(1)*10)+LV:I
        =1
JK 1630 X=INT(RND(1)*10)+1:Y=I
        NT(RND(1)*10)+1:IFFL(Y
        ,X)=3THEN1630
AQ 1640 FL(Y,X)=3:I=I+1:IFI<=N
        MTHEN1630
EG 1650 FORI=1TONM*2
DX 1660 X=INT(RND(1)*10)+1:Y=I
        NT(RND(1)*10)+1
MF 1670 IFFL(Y,X)=3THEN1660
HS 1680 FL(Y,X)=0:NEXTI
QR 1690 FORI=1TO10:FORJ=1TO10
SE 1700 IFFL(I,J)=3THENGOSUB17
        80
SP 1710 NEXTJ:NEXTI
MJ 1720 GOSUB1840:GX=X:GY=Y:IF
        (BX=XANDBY=Y)OR(RX=XAN
        DRY=Y)ORFL(Y,X)=0THEN1
        720
SS 1730 GOSUB1840:BX=X:BY=Y:IF
        (GX=XANDGY=Y)OR(RX=XAN
        DRY=Y)ORFL(Y,X)=0THEN1
        730
HF 1740 GOSUB1840:X1=X:Y1=Y:IF
        (X2=XANDY2=Y)OR(X3=XAN
        DY3=Y)ORFL(Y,X)=0THEN1
        740
RK 1750 GOSUB1840:X2=X:Y2=Y:IF
        (X1=XANDY1=Y)OR(X3=XAN
        DY3=Y)ORFL(Y,X)=0THEN1
        750
CJ 1760 GOSUB1840:X3=X:Y3=Y:IF
        (X1=XANDY1=Y)OR(X2=XAN
        DY2=Y)ORFL(Y,X)=0THEN1
        760
XB 1770 RETURN
KH 1780 FORK=JTO10:GOSUB1800:N
        EXTK:FORK=JTO1STEP-1:G
        OSUB1800:NEXTK
JH 1790 FORK=ITO10:GOSUB1820:N
        EXTK:FORK=ITO1STEP-1:G
        OSUB1820:NEXTK:RETURN
AC 1800 IFFL(I,K)=0THENFL(I,K)
        =INT(RND(1)*2)+1
HD 1810 RETURN
SX 1820 IFFL(K,J)=0THENFL(K,J)
        =INT(RND(1)*2)+1
RC 1830 RETURN
DR 1840 X=INT(RND(1)*10)+1:Y=I
        NT(RND(1)*10)+1:IFFL(Y
        ,X)=0THEN1840
DE 1850 RETURN
KA 1860 GOSUB790:FORI=12TO19:C
        M(I)=1:NEXT:CM(22)=1
EP 1870 PRINT"{CLR}":PR$=DS$(1
        6-Z):GOSUB2570:IFLO=1T
        HENGOSUB1030
XF 1880 GOSUB2270:GOSUB2200
SB 1890 GOSUB2310:IFPL=15ANDLO
        THENGOSUB1020:GOTO1880
FS 1900 IFPL=20THENGOSUB2850:G
        OTO1890
CF 1910 IFPL=15ANDLO=0THEN1870
DR 1920 IFPL=1THENGOSUB1190:GO
        TO1890
KJ 1930 IFPL=14THEN1890
QD 1940 IFPL=11THENGOSUB2490:G
        OTO1890
MQ 1950 IFPL=8THENGOSUB1130:GO
        TO1890
XC 1960 IFPL=22THENGOSUB2470:R
        ETURN
KF 1970 IFPL>15THENGOSUB1560:I
        F(LO=0)OR(WA=1)THEN189
        0
GX 1980 RETURN
AB 2000 GOSUB790:FORI=10TO19:C
        M(I)=1:NEXTI:CM(22)=1
JD 2010 PR$=DS$(23):GOSUB2570:
        PR$=DS$(24):GOSUB2570:
        GOSUB1320:IFWR=0THENGO
        TO2050
XD 2020 GP=0:PR$=DS$(25):GOSUB
        2570:FORW=1TO2000:NEXT
        W
FG 2030 X=INT(RND(1)*10+1):Y=I
        NT(RND(1)*10)+1:IFFL(Y
        ,X)=0THEN2030
GB 2040 PX=X:PY=Y:Z=FL(PY,PX):
        RETURN
EQ 2050 PR$=DS$(26):GOSUB2570:
        FL(PY,PX)=2
BA 2060 PR$=DS$(14):GOSUB2570:
        IFLOTHENGOSUB1030
PJ 2070 GOSUB2270:GOSUB2200
MG 2080 GOSUB2310
PM 2090 IFPL=20THENGOSUB2850:G
        OTO2080
CH 2100 IFPL=15ANDLOTHENGOSUB1
        020:GOTO2070
PF 2110 IFPL=15ANDLO=0THEN2060
PS 2120 IFPL=14THEN2080
JE 2130 IFPL=11THEN2490:GOTO20
        80
QJ 2140 IFPL=8THENGOSUB1130:GO
        TO2080
BQ 2150 IFPL=1THENGOSUB1190:GO
        TO2080
DM 2160 IFPL=10THENGOSUB1290:G
        OTO2080
GX 2170 IFPL=22THENGOSUB2470:R
        ETURN
QJ 2180 IFPL>15THENGOSUB1560:I
        F(LO=0)OR(WA=1)THEN208
        0
XP 2190 RETURN
BM 2200 IF(X1=PXANDY1=PY)ANDGD
        <>2THENPR$=DS$(17):GOS
        UB2570:GD=1:CM(1)=1
JF 2210 IF(X2=PXANDY2=PY)ANDBD
        <>2THENPR$=DS$(19):GOS
        UB2570:BD=1:CM(1)=1
BF 2220 IF(X3=PXANDY3=PY)ANDRD
        <>2THENPR$=DS$(18):GOS
        UB2570:RD=1:CM(1)=1
CF 2230 IF(X1<>PXORY1<>PY)ANDG
        D<>2THENGD=0
FE 2240 IF(X2<>PXORY2<>PY)ANDB
        D<>2THENBD=0
JE 2250 IF(X3<>PXORY3<>PY)ANDR
        D<>2THENRD=0
SX 2260 RETURN
JC 2270 IFBX=PXANDBY=PYANDIN(3
        )=0THENPR$=DS$(21):GOS
        UB2570:BK=1:CM(8)=1
XH 2280 IFGX=PXANDGY=PYANDIN(2
        )=0THENPR$=DS$(20):GOS
        UB2570:GK=1:CM(8)=1
CH 2290 IFRX=PXANDRY=PYANDIN(4
        )=0THENPR$=DS$(22):GOS
        UB2570:RK=1:CM(8)=1
GS 2300 RETURN
KA 2310 INPUTC$:PL=0:FORI=1TO2
        2:IFCM$(I)=C$THENPL=I:
        I=23
KD 2320 NEXT
BK 2330 IFPL=0THENPRINT"YOU CA
        N'T DO THAT!":GOTO2310
XD 2340 N$="":I=1
FH 2350 T$=MID$(C$,I,1)
```

```
JD 2360 IFASC(T$)=32THENL=I:GO
        TO2390
AA 2370 IFI=LEN(C$)THENGOTO241
        0
ES 2380 I=I+1:GOTO2350
KP 2390 N$=MID$(C$,L+1,LEN(C$)
        )
MX 2400 IFCM(PL)=0THENPRINT"I
        {SPACE}SEE NO "N$" HER
        E!":GOTO2310
BR 2410 IFC$="INV"THENGOTO2430
HG 2420 RETURN
JG 2430 PRINT"YOU ARE CARRYING
        :"
AM 2440 FORI=1TO5:IFIN(I)=1THE
        NPRINT"A "IN$(I)
PB 2450 NEXTI:IFGP<>0THENPRINT
        GP" GOLD PIECES."
PM 2460 RETURN
JX 2470 PX=INT(RND(1)*10)+1:PY
        =INT(RND(1)*10)+1:IFFL
        (PY,PX)=0THEN2470
KE 2480 Z=FL(PY,PX):RETURN
QK 2490 IFIN(5)=1THENPR$="YOU
        {SPACE}ALREADY HAVE TH
        E MAP!":GOSUB2570:RETU
        RN
CS 2500 IFMA=0THENPRINT"WHAT M
        AP?":RETURN
HR 2510 PRINT"OK.":IN(5)=1:MA=
        0:CM(20)=1:GD=2:RETURN
JB 2520 PR$="{2 DOWN}YOU HAVE
        {SPACE}FOUND YOUR WAY
        {SPACE}OUT OF THE MATH
         DUNGEON.":GOSUB2570
XK 2530 PR$="{DOWN}{2 SPACES}Y
        OU HAVE WON "+STR$(GP)
        +" GOLD PIECES IN YOUR
         JOURNEY.":GOSUB2570
BP 2540 PRINT"{DOWN} PLAY AGAI
        N";:INPUT A$
XB 2550 IFA$="Y"THENRUN
GB 2560 END
BF 2570 IFLEN(PR$)<=CCTHENPRIN
        TPR$:RETURN
SH 2580 T$=MID$(PR$,CC,1):IFAS
        C(T$)=32THENI=CC:GOTO2
        620
MH 2590 I=CC-1
MS 2600 T$=MID$(PR$,I,1):IFASC
        (T$)=32THENGOTO2620
FM 2610 I=I-1:GOTO2600
BS 2620 A$=MID$(PR$,I+1,LEN(PR
        $)):B$=MID$(PR$,1,I-1)
        :PRINTB$
PD 2630 IFLEN(A$)<=CCTHENPRINT
        A$:RETURN
PE 2640 PR$=A$:GOTO2570
XH 2650 DATAOPEN DOOR,OPEN GAT
        E,OPEN MILKBOX,CLOSE D
        OOR,CLOSE GATE,CLOSE M
        ILKBOX
SP 2660 DATAGET NOTE,GET KEY,G
        ET LIGHT,GET GOLD,GET
        {SPACE}MAP,LIGHT ON,LI
        GHT OFF,INV
RC 2670 DATALOOK,N,S,E,W,MAP,M
        AP,ALGEBRA,LIGHT,GREEN
         KEY,BLUE KEY,RED KEY,
        MAP
KF 2680 DATAYOU ARE FACING NOR
        TH. A GATE IS IN FRONT
         OF YOU.,THE GATE IS O
        PEN.
CM 2690 DATATHE GATE IS CLOSED
        .,YOU ARE IN A DENSE F
        OREST.
FQ 2700 DATAYOU ARE IN FRONT O
        F AN OLD HOUSE. THERE
        {SPACE}IS A MILKBOX BY
         THE DOOR.
DS 2710 DATATHE MILKBOX IS CLO
        SED.,THE MILKBOX IS OP
        EN.,THE DOOR IS CLOSED
        .
EC 2720 DATATHE DOOR IS OPEN.,
        THE DOOR IS LOCKED. YO
        U DO NOT HAVE THE RIGH
        T KEY.
HM 2730 DATATHERE IS A NOTE IN
         THE MILKBOX.,THE DOOR
         FLIES OPEN WITH A BAN
        G!
KD 2740 DATATHE NOTE DISAPPEAR
        S AND THE MILKBOX CLOS
        ES.
BF 2750 DATAYOU ARE IN A COLD
        {SPACE}AND DAMP ROOM.,
        YOU ARE IN A LONG AND
        {SPACE}NARROW HALLWAY.
MG 2760 DATATHERE IS AN UNLIT
        {SPACE}LIGHT HERE.,THE
        RE IS A GREEN DOOR HER
        E.
XR 2770 DATATHERE IS A RED DOO
        R HERE.,THERE IS A BLU
        E DOOR HERE.
QA 2780 DATATHERE IS A GREEN K
        EY HERE.,THERE IS A BL
        UE KEY HERE.
AF 2790 DATATHERE IS A RED KEY
         HERE.,BEFORE YOU CAN
        {SPACE}GET A GOOD LOOK
         AT THE AREA..
FS 2800 DATAA MONSTER JUMPS OU
        T AND SAYS 'YOU MUST A
        NSWER THIS QUESTION!
QD 2810 DATA'NOW I'LL TAKE ALL
         YOUR GOLD AND TRANSPO
        RT YOU AWAY!
CF 2820 DATATHE MONSTER DISAPP
        EARS.{17 SPACES}YOU SE
        E SOME GOLD HERE!
MS 2830 DATATHERE IS A MAP HER
        E!,THERE IS A RED KEY
        {SPACE}HERE!
SF 2840 DATATHERE IS A SUDDEN
        {SPACE}FLASH AND YOU F
        IND YOURSELF AMIDST SO
        ME TREES.
HH 2850 IFIN(5)=0THENPRINT"YOU
         DON'T HAVE A MAP!":RE
        TURN
XC 2860 PRINT"{CLR}":PRINT
HS 2870 FORI=1TO10:TB=1:FORJ=1
        TO10
BF 2880 IFI=PYANDJ=PXTHENPRINT
        TAB(TB)"{GRN}*";:GOTO2
        930
DJ 2890 IFFL(I,J)=3THENPRINTTA
        B(TB)"{BLK}M";:GOTO293
        0
GJ 2900 IFFL(I,J)=2THENPRINTTA
        B(TB)"{BLU}R";:GOTO293
        0
SE 2910 IFFL(I,J)=1THENPRINTTA
        B(TB)"{PUR}H";:GOTO293
        0
KD 2920 PRINTTAB(TB)"{RVS}
        {YEL} {OFF}";
KS 2930 TB=TB+2:NEXTJ:PRINT:PR
        INT:NEXTI
GP 2940 PRINT"{BLU}":RETURN
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Laser Chess

Version by Bill Chin, Editorial Programmer

See instructions in article on page 6 before typing in.

```
0801:0B 08 0A 00 9E 32 30 36 2E
0809:31 00 00 00 A9 00 8D 21 3B
0811:D0 8D 20 D0 A9 0F 8D 86 29
0819:02 20 C0 0A A9 78 A0 18 73
0821:20 1E AB A5 C6 F0 FC A9 36
0829:FF 8D 21 C8 20 7B 0B A9 FB
0831:09 8D 04 C8 A9 07 8D 21 DB
0839:D0 A9 00 8D 16 C8 A9 00 1C
0841:85 FD A9 C0 85 FE A9 CE 1F
0849:85 FB A9 18 85 FC A0 00 33
0851:B1 FB F0 04 C9 19 90 0C 77
0859:91 FD 20 FD 08 20 EC 08 38
0861:D0 EE F0 17 AA A9 FF 91 B2
0869:FD CA F0 07 20 EC 08 D0 4F
0871:F4 F0 08 20 FD 08 20 EC 78
0879:08 D0 D5 20 04 09 20 4C 4F
0881:0B 20 AA 0E AD 16 C8 49 F5
0889:01 8D 16 C8 20 06 0B F0 EC
0891:F0 A9 07 20 91 0A A9 03 72
0899:8D 24 C8 A9 09 8D 23 C8 BA
08A1:A9 0A 8D 31 C9 8D 32 C9 80
08A9:A2 01 A9 05 20 2A 10 20 BA
08B1:15 17 20 38 0F 20 8B 0F B8
08B9:AD 0A C8 F0 08 DE 31 C9 33
08C1:F0 10 4C CB 08 A9 0A 9D 2D
08C9:31 C9 CA 10 DD A2 01 4C 07
08D1:AB 08 AD 23 C8 C9 09 D0 F1
08D9:B8 AD 24 C8 C9 03 F0 05 04
08E1:C9 02 D0 AD 00 20 04 09 DD
08E9:4C 82 08 E6 FD D0 02 E6 4E
08F1:FE A5 FD C9 E0 D0 04 A5 3F
08F9:FE C9 C4 60 E6 FB D0 02 65
0901:E6 FC 60 AD C6 17 A2 00 84
0909:9D E0 C8 E8 E0 12 D0 F8 B3
0911:A9 FF A2 12 9D E0 C8 E8 58
0919:E0 3F D0 F8 AD C7 17 A2 72
0921:3F 9D E0 C8 E8 E0 51 D0 21
0929:F8 A2 00 8E 25 C8 A0 08 DE
0931:BD C8 17 9D 3E C8 99 86 DF
0939:C8 BD D1 17 9D 47 C8 99 FF
0941:7D C8 BD F0 17 9D 8F C8 22
0949:BD 0B 18 9D D7 C8 BD F9 31
0951:17 9D 98 C8 BD 02 18 9D B9
0959:CE C8 E8 88 10 D2 A9 03 CC
0961:8D E5 CA A9 4D 8D E6 CA E0
0969:A9 FF 8D 21 C8 20 B3 0B 4D
0971:A9 08 8D 23 C8 8D 24 C8 CB
0979:20 40 0D AD 0E C8 C9 FF 4F
0981:F0 06 20 A3 0C 4C 8C 09 7F
0989:20 FC 0B CE 23 C8 10 E8 7E
0991:A9 08 8D 23 C8 CE 24 C8 F0
0999:10 DE A9 04 8D 23 C8 8D F8
09A1:24 C8 20 40 0D A9 00 8D 9C
09A9:21 C8 20 5F 0C A9 01 8D 0F
09B1:66 C8 A9 03 8D 20 C8 8D 9A
09B9:24 C8 A9 16 8D 31 C9 A9 15
09C1:01 8D 23 C8 8D 1F C8 20 43
09C9:DE 09 EE 31 C9 EE 24 C8 99
09D1:AD 24 C8 C9 06 D0 E8 EE AD
09D9:24 C8 EE 24 C8 AD 31 C9 79
09E1:8D 02 C8 8D 00 C8 AD 24 CF
09E9:C8 8D 20 C8 20 58 0D 20 F0
09F1:A3 0C A9 00 8D 21 C8 20 B0
09F9:5F 0C A9 E1 8D 22 C8 20 B8
0A01:81 0C 60 8A 48 A9 08 8D 14
0A09:23 C8 8D 24 C8 20 40 0D 29
0A11:AD 0E C8 CD 1A C8 D0 14 1F
0A19:20 E6 0B AD 21 C8 8D 22 9C
0A21:C8 20 81 0C A9 FF AC 12 4B
0A29:C8 99 E0 C8 CE 23 C8 10 55
0A31:DC A9 08 8D 23 C8 CE 24 F5
0A39:C8 10 D2 68 AA 60 A9 4E 0F
0A41:8D F9 57 A9 DC 8D 03 D0 14
0A49:A9 A8 8D 02 D0 A9 00 8D E8
0A51:10 D0 A9 06 8D 28 D0 A9 8F
0A59:02 8D 1D D0 A9 03 8D 15 0C
```

```
0A61:D0 20 E4 FF C9 00 D0 D6 49
0A69:A9 07 20 91 0A EE 28 D0 5E
0A71:CE 27 D0 20 E4 FF C9 00 8D
0A79:F0 F9 C9 4E F0 0A C9 59 3F
0A81:D0 E6 A9 01 8D 15 D0 60 BF
0A89:A9 01 8D 15 D0 A9 00 60 43
0A91:8D FA 0A 8A 48 98 48 AE F8
0A99:FA 0A BD DA 0A A8 BD E2 64
0AA1:0A 99 05 D4 A9 00 99 06 95
0AA9:D4 BD EA 0A 99 01 D4 BD CD
0AB1:F2 0A 99 04 D4 49 01 99 9C
0AB9:04 D4 68 A8 68 AA 60 A2 ED
0AC1:18 A9 00 9D 00 D4 CA 10 1F
0AC9:FA A9 0F 8D 18 D4 A9 FF E7
0AD1:8D 0F D4 A9 80 8D 12 D4 D8
0AD9:60 00 07 00 00 07 00 00 1B
0AE1:00 01 19 04 19 8A 92 84 36
0AE9:19 17 10 50 0A 01 64 03 77
0AF1:1E 10 10 80 20 80 80 80 A7
0AF9:20 00 A0 A5 A2 8D EA 0A B7
0B01:A9 00 4C 91 0A AD 25 C8 A8
0B09:F0 3A C9 02 F0 22 C9 03 26
0B11:F0 33 A0 51 B9 3E C8 D0 BE
0B19:14 B9 E0 C8 CD C6 17 F0 F8
0B21:18 AD C6 17 8D 1A C8 20 7F
0B29:04 0A A9 01 60 88 10 E4 33
0B31:AD C6 17 8D 1A C8 20 04 C3
0B39:0A AD C7 17 8D 1A C8 20 B0
0B41:04 0A A9 01 60 20 04 09 B5
0B49:A9 00 60 AD 11 D0 09 20 19
0B51:8D 11 D0 AD 18 D0 09 08 85
0B59:29 0F 09 50 8D 18 D0 AD 0A
0B61:16 D0 09 10 8D 16 D0 AD EC
0B69:02 DD 09 03 8D 02 DD AD 27
0B71:00 DD 29 FC 09 02 8D 00 5F
0B79:DD 60 A9 00 A2 3F 9D C0 D9
0B81:53 CA 10 FA A9 FF A2 00 38
0B89:9D C0 53 9D F6 53 E8 E0 9A
0B91:06 D0 F5 A2 06 A9 80 9D 3D
0B99:C0 53 A9 01 9D C2 53 E8 B1
0BA1:E8 E8 E0 36 90 EF A2 3F AE
0BA9:BD 79 1B 9D 80 53 CA 10 31
0BB1:F7 60 A9 00 8D C3 0B A9 4C
0BB9:60 8D C4 0B AD 21 C8 A0 D0
0BC1:00 99 41 C9 C8 D0 FA EE 71
0BC9:C4 0B AE C4 0B E0 80 D0 D4
0BD1:F0 A0 00 A9 00 99 00 D8 62
0BD9:99 00 D9 99 00 DA 99 E8 19
0BE1:DA C8 D0 F1 60 A9 0B 8D 1E
0BE9:21 C8 AD 23 C8 18 6D 24 50
0BF1:C8 29 01 D0 05 A9 0F 8D 5E
0BF9:21 C8 60 20 0B 0D A9 FF C0
0C01:8D 22 C8 AC 12 C8 99 3E 71
0C09:C8 99 8F C8 99 E0 C8 A9 F5
0C11:00 8D 02 C8 20 46 0D 20 AD
0C19:25 0C 20 B3 0C 20 2F 0C 51
0C21:20 5C 0C 60 AD 2A C8 85 15
0C29:FD AD 2B C8 85 FE A0 00 07
0C31:AD 22 C8 91 FD C8 C0 18 87
0C39:D0 F9 60 AD 2A C8 85 FD 9C
0C41:AD 2B C8 85 FE AD 2C C8 3C
0C49:85 FB AD 2D C8 85 FC A0 A2
0C51:00 B1 FB 91 FD C8 C0 18 1B
0C59:D0 F7 60 20 E6 0B A0 00 8A
0C61:AD 2E C8 85 FD AD 2F C8 1B
0C69:85 FE AD 21 C8 91 FD C8 1D
0C71:91 FD C8 91 FD A0 28 91 58
0C79:FD C8 91 FD C8 91 FD 60 BD
0C81:A0 00 AD 30 C8 85 FD AD A8
0C89:31 C8 85 FE AD 22 C8 91 26
0C91:FD C8 91 FD C8 91 FD A0 16
0C99:28 91 FD C8 91 FD C8 91 1E
0CA1:FD 60 20 3C 0C 20 B3 0C EC
0CA9:20 50 0C 20 81 0C 20 5C 42
0CB1:0C 60 A5 FD 18 69 40 85 E8
0CB9:FD A5 FE 69 01 85 FE A5 72
0CC1:FB 18 69 18 85 FB A5 FC F0
0CC9:69 00 85 FC 60 AD 1B D4 DB
0CD1:29 0F C9 09 B0 F7 F0 F5 49
0CD9:60 AD 1B D4 6D 41 C9 29 6B
0CE1:7F C9 51 B0 F4 A8 60 49 B5
0CE9:FF 85 02 E6 02 A5 02 60 1D
0CF1:8C 23 C8 A9 00 8D 24 C8 14
0CF9:AD 23 C8 38 E9 09 30 09 2B
0D01:8D 23 C8 EE 24 C8 4C F9 89
0D09:0C 60 AD 24 C8 8D 35 C9 E9
0D11:0A 0A 0A 18 6D 35 C9 6D B6
0D19:23 C8 8D 35 C9 8D 12 C8 6D
0D21:A8 60 B9 3E C8 8D 00 C8 08
0D29:B9 8F C8 8D 10 C8 18 6D 37
0D31:00 C8 8D 02 C8 B9 E0 C8 07
0D39:8D 0E C8 8D 22 C8 60 20 A4
0D41:0B 0D 20 23 0D AD 24 C8 8A
0D49:18 69 01 8D 20 C8 AD 23 65
0D51:C8 18 69 02 8D 1F C8 AD 4B
0D59:20 C8 0A 8D 2B C8 A9 00 9F
0D61:8D 2A C8 AD 20 C8 4A 6E E7
0D69:2A C8 18 6D 2B C8 8D 2B 67
0D71:C8 AD 1F C8 0A 0A 0A 8D E5
0D79:35 C9 0A 18 6D 35 C9 18 4F
0D81:6D 2A C8 8D 2A C8 AD 2B C9
0D89:C8 69 00 8D 2B C8 AD 2A 3D
0D91:C8 18 69 48 8D 2A C8 AD 1C
0D99:2B C8 69 61 8D 2B C8 A9 13
0DA1:00 8D 36 C9 AD 02 C8 0A 93
0DA9:2E 36 C9 0A 2E 36 C9 0A 2A
0DB1:2E 36 C9 0A 2E 36 C9 8D B5
0DB9:35 C9 8D 37 C9 AD 36 C9 41
0DC1:8D 38 C9 0E 35 C9 2E 36 2E
0DC9:C9 AD 35 C9 18 6D 37 C9 26
0DD1:8D 35 C9 AD 36 C9 6D 38 FF
0DD9:C9 8D 36 C9 AD 35 C9 69 DE
0DE1:00 8D 2C C8 AD 36 C9 69 B4
0DE9:C0 8D 2D C8 A9 00 8D 35 97
0DF1:C9 8D 36 C9 AD 20 C8 0A 41
0DF9:2E 36 C9 0A 2E 36 C9 0A 7A
0E01:2E 36 C9 0A 2E 36 C9 8D 07
0E09:35 C9 8D 37 C9 AD 36 C9 92
0E11:8D 38 C9 0E 37 C9 2E 38 91
0E19:C9 0E 37 C9 2E 38 C9 AD B4
0E21:37 C9 18 6D 35 C9 8D 2E 3F
0E29:C8 AD 38 C9 6D 36 C9 8D 1E
0E31:2F C8 AD 1F C8 8D 35 C9 6F
0E39:0A 18 6D 35 C9 8D 35 C9 1A
0E41:18 6D 2E C8 8D 35 C9 AD 99
0E49:2F C8 69 00 8D 36 C9 AD E2
0E51:35 C9 18 69 29 8D 2E C8 B8
0E59:AD 36 C9 69 D8 8D 2F C8 CD
0E61:AD 35 C9 18 69 29 8D 30 97
0E69:C8 AD 36 C9 69 54 8D 31 A1
0E71:C8 AD 02 C8 0A 0A 8D 32 EF
0E79:C8 A9 20 18 6D 32 C8 8D 3D
0E81:32 C8 A9 18 69 00 8D 33 39
0E89:C8 AD 1F C8 0A 0A 8D 38 B1
0E91:C8 0A 18 6D 38 C8 69 04 2A
0E99:8D 38 C8 AD 20 C8 0A 0A C0
0EA1:0A 0A 18 69 0C 8D 39 C8 B0
0EA9:60 A9 00 8D 03 C8 8D 05 94
0EB1:C8 8D 14 C8 8D 26 C8 8D C8
0EB9:E3 CA 8D E4 CA A9 02 8D 09
0EC1:0C C8 A9 04 8D 23 C8 8D A3
0EC9:24 C8 AE 16 C8 BD C6 17 43
0ED1:8D 0F C8 20 B8 12 AD 25 22
0ED9:C8 D0 0B 20 E8 0E 20 15 C6
0EE1:17 AD 0C C8 D0 F0 60 A9 B7
0EE9:05 20 2A 10 20 E2 10 AE 32
0EF1:16 C8 20 38 0F AD 05 C8 D4
0EF9:F0 03 4C 01 0F 4C 2A 0F F5
0F01:20 D0 0F A9 14 8D 04 C8 87
0F09:20 0B 0D CC 13 C8 F0 03 09
0F11:4E 04 C8 AD 0A C8 F0 0B AB
0F19:AD 0B C8 F0 03 4C E2 0F 17
0F21:4C 39 10 8D 03 C8 4C 8B ED
0F29:0F AD 0A C8 F0 03 4C 39 6D
0F31:10 8D 03 C8 4C 8B 0F A9 FF
0F39:00 8D 0A C8 8D 0B C8 8D 40
0F41:17 C8 8D 1B C8 BD 00 DC 9A
0F49:8D 41 C9 4E 41 C9 B0 08 37
0F51:A9 FF 8D 1B C8 8D 0B C8 03
0F59:4E 41 C9 B0 08 A9 01 8D A9
0F61:1B C8 8D 0B C8 4E 41 C9 6D
0F69:B0 08 A9 FF 8D 17 C8 8D FE
0F71:0B C8 4E 41 C9 B0 08 A9 EF
0F79:01 8D 17 C8 8D 0B C8 4E 63
0F81:41 C9 B0 05 A9 01 8D 0A 8F
0F89:C8 60 AD 23 C8 18 6D 17 A4
0F91:C8 8D 23 C8 AD 24 C8 18 10
0F99:6D 1B C8 8D 24 C8 AD 24 EA
0FA1:C8 30 0E C9 09 F0 02 90 2F
0FA9:0D A9 00 8D 24 C8 4C B7 26
0FB1:0F A9 08 8D 24 C8 AD 23 5E
0FB9:C8 30 0E C9 0A F0 02 90 4F
0FC1:0D A9 00 8D 23 C8 4C CF 4E
0FC9:0F A9 09 8D 23 C8 60 AD 7E
0FD1:36 C8 85 FD AD 37 C8 85 2F
0FD9:FE A9 04 8D 21 C8 4C 5F 5F
0FE1:0C 20 0B 0D 20 23 0D AD 95
0FE9:12 C8 CD 13 C8 D0 35 AD CF
0FF1:10 C8 18 6D 0B C8 29 03 F4
0FF9:8D 34 C9 AD 00 C8 AA BD 36
1001:DA 17 2D 34 C9 8D 34 C9 F3
1009:AC 12 C8 AD 34 C9 99 8F 83
1011:C8 20 23 0D 20 46 0D 20 27
1019:A3 0C A9 1E 20 2A 10 A9 98
1021:02 4C 91 0A A9 01 4C 91 A3
1029:0A 18 65 A2 8D 38 10 A5 3E
1031:A2 CD 38 10 D0 F9 60 00 4D
1039:AE 03 C8 EC 04 C8 F0 05 83
1041:E8 8E 03 C8 60 A9 00 8D 9D
1049:03 C8 AD 23 C8 C9 09 D0 55
1051:03 4C C0 12 AD 05 C8 49 9B
1059:01 8D 05 C8 F0 4E 20 40 CB
1061:0D AD 0E C8 CD 0F C8 D0 CE
1069:39 AD 12 C8 8D 13 C8 A9 54
1071:02 20 91 0A AD 2E C8 8D B2
1079:36 C8 AD 2F C8 8D 37 C8 43
1081:AD 0C C8 8D 0D C8 AD 10 64
1089:C8 8D 11 C8 8D 34 C9 AD 9E
1091:23 C8 8D 06 C8 AD 24 C8 95
1099:8D 07 C8 AD 00 C8 8D 01 75
10A1:C8 60 A9 00 8D 05 C8 A9 2F
10A9:01 4C 91 0A A9 0A 8D 04 C4
10B1:C8 AD 23 C8 CD 06 C8 D0 7B
10B9:16 AD 24 C8 CD 07 C8 D0 4E
10C1:0E A9 00 8D 15 C8 20 89 C1
10C9:12 20 86 17 4C DF 10 20 5F
10D1:25 11 F0 42 20 89 12 F0 47
10D9:3D 20 CF 11 F0 38 4C B8 65
10E1:12 AE 16 C8 BD E5 CA 10 31
10E9:05 A9 01 8D 26 C8 A0 02 87
10F1:AD 0F C8 99 BC 55 AE 0C 04
10F9:C8 E0 02 F0 02 A9 00 99 56
1101:E4 55 AD 0F C8 AE 26 C8 A7
1109:F0 02 A9 00 99 AC 56 99 1F
1111:D4 56 88 10 DB 60 AD 0D 0E
1119:C8 8D 0C C8 A9 01 8D 05 82
1121:C8 4C 91 0A AD 23 C8 38 51
1129:ED 06 C8 8D 18 C8 10 03 BC
1131:20 E8 0C 8D 35 C9 AD 24 48
1139:C8 38 ED 07 C8 8D 1C C8 79
1141:10 03 20 E8 0C 18 6D 35 8F
1149:C9 8D 15 C8 F0 08 C9 01 1F
1151:F0 04 C9 02 F0 03 A9 01 2E
1159:60 20 0B 0D AD 18 C8 C9 0F
1161:FE D0 06 B9 3F C8 4C C6 10
1169:11 AD 18 C8 C9 02 D0 06 0D
1171:B9 3D C8 4C C6 11 AD 1C 8F
1179:C8 C9 02 D0 06 B9 35 C8 0A
1181:4C C6 11 AD 1C C8 C9 FE 0F
1189:D0 06 B9 47 C8 4C C6 11 57
1191:AD 18 C8 C9 01 D0 0A B9 5F
1199:3D C8 C9 FF F0 2D 4C A9 44
11A1:11 B9 3F C8 C9 FF F0 23 82
11A9:AD 1C C8 C9 01 D0 0C A0 63
11B1:00 B9 35 C8 C9 FF F0 13 B8
11B9:4C C3 11 B9 47 C8 C9 FF A1
11C1:F0 09 A9 00 60 C9 FF F0 EE
11C9:02 D0 F7 A9 01 60 20 0B 8F
11D1:0D C0 28 D0 2D AD E4 CA 71
11D9:D0 3E A9 05 20 91 0A A9 7E
11E1:00 8D 41 C9 EE E4 CA 20 EC
11E9:DA 0C C0 28 F0 0D B9 3E 84
11F1:C8 C9 FF D0 06 20 F1 0C 98
11F9:4C 2F 12 EE 41 C9 D0 E7 F9
1201:F0 16 B9 3E C8 C9 FF F0 9C
1209:25 AD 01 C8 C9 00 F0 0B 13
1211:C9 01 F0 20 C9 06 F0 03 C5
1219:A9 00 60 AD 14 C8 D0 F8 57
1221:EE 14 C8 B9 3E C8 20 EE BA
1229:15 A9 06 20 91 0A 20 95 8F
1231:17 A9 01 60 AD E3 CA F0 F4
1239:03 4C 19 12 20 DA 0C B9 74
1241:3E C8 C9 FF D0 07 C0 28 3C
1249:F0 03 4C 56 12 EE 41 C9 2E
```

```
1251:D0 EA 4C 19 12 8C 42 C9 C4
1259:AE 12 C8 20 73 17 EE E3 2E
1261:CA AD 23 C8 48 AD 24 C8 51
1269:48 AC 42 C9 20 F1 0C 20 C2
1271:40 0D A9 05 20 91 0A 20 F9
1279:A3 0C 68 8D 24 C8 68 8D FA
1281:23 C8 20 95 17 A9 01 60 88
1289:AD 0C C8 8D 35 C9 AD 34 D9
1291:C9 CD 11 C8 F0 05 CE 35 2B
1299:C9 30 10 AD 15 C8 F0 0E 47
12A1:CE 15 C8 CE 35 C9 30 03 AC
12A9:4C 9C 12 A9 00 60 AD 35 0A
12B1:C9 8D 0C C8 A9 01 60 A9 E7
12B9:01 8D 15 D0 4C 91 0A AD DB
12C1:24 C8 C9 02 D0 06 20 3F A1
12C9:0A F0 EC 00 C9 03 D0 10 D8
12D1:20 3F 0A F0 E2 A9 00 8D 71
12D9:0C C8 A9 03 8D 25 C8 60 8E
12E1:AD 05 C8 D0 D2 AD 24 C8 A2
12E9:C9 04 D0 06 A9 00 8D 0C E2
12F1:C8 60 C9 07 F0 01 60 AD 36
12F9:26 C8 D0 BB AE 16 C8 A9 42
1301:02 8D 28 C8 8D 29 C8 A2 62
1309:02 A9 04 9D 51 C9 A9 FF F9
1311:9D 45 C9 9D 55 C9 9D 57 CE
1319:C9 E8 E0 06 D0 EB A9 00 64
1321:AA 9D 63 C9 E8 E0 60 D0 69
1329:F8 EE 26 C8 CE 0C C8 A2 B3
1331:51 BD 3E C8 C9 12 F0 06 42
1339:CA 10 F6 4C B8 12 BD E0 D6
1341:C8 CD 0F C8 D0 F2 8A AE C3
1349:16 C8 9D E5 CA A8 20 F1 E9
1351:0C AD 23 C8 8D 47 C9 AD A4
1359:24 C8 8D 4D C9 8D 24 C8 DF
1361:20 0B 0D B9 8F C8 8D 53 A5
1369:C9 A9 00 8D 27 C8 A0 02 57
1371:8C 28 C8 B9 51 C9 C9 04 E5
1379:F0 1D B9 45 C9 8D 23 C8 7E
1381:B9 4B C9 8D 24 C8 20 40 2E
1389:0D 20 69 15 AC 28 C8 20 74
1391:74 16 F0 03 20 0E 15 EE 18
1399:28 C8 AC 28 C8 CC 29 C8 B2
13A1:F0 D1 90 CF AD 27 C8 D0 30
13A9:C0 A9 0F 8D 33 C9 A9 0A 73
13B1:8D 1E C8 AD 1E C8 20 2A 98
13B9:10 CE 1E C8 10 03 EE 1E 74
13C1:C8 A2 00 8E 27 C8 20 38 B2
13C9:14 BD 63 C9 F0 08 20 64 BE
13D1:14 A9 01 8D 27 C8 AD 33 50
13D9:C9 D0 35 BD 63 C9 48 BD 2C
13E1:C3 C9 48 BD 34 03 9D 63 8D
13E9:C9 BD 94 03 9D C3 C9 20 D6
13F1:38 14 BD C3 C9 9D 94 03 1E
13F9:BD 63 C9 9D 34 03 F0 08 82
1401:A9 01 8D 27 C8 20 64 14 06
1409:68 9D C3 C9 68 9D 63 C9 2C
1411:E8 E0 60 90 BJ AD 33 C9 6F
1419:F0 12 CE 33 C9 D0 0D A9 A0
1421:77 8D 1A C8 20 04 0A A9 07
1429:08 8D 1E C8 AD 27 C8 F0 95
1431:03 4C B4 13 4C 69 09 BD 8D
1439:63 C9 F0 26 18 7D 23 CA CD
1441:C9 9F 90 08 A9 00 9D 63 B4
1449:C9 4C 63 14 9D 63 C9 BD E2
1451:C3 C9 18 7D 83 CA 9D C3 EE
1459:C9 C9 C7 90 05 A9 00 9D 47
1461:63 C9 60 8A 48 98 48 BD 55
1469:63 C9 4A 4A 8D 0D 15 BD 2C
1471:C3 C9 4A 4A 4A 8D 0C 15 91
1479:A9 00 8D 36 C9 AD 0C 15 BD
1481:0A 2E 36 C9 0A 2E 36 C9 DC
1489:0A 2E 36 C9 0A 2E 36 C9 E4
1491:0A 2E 36 C9 0A 2E 36 C9 EC
1499:8D 35 C9 AD 36 C9 18 6D 60
14A1:0C 15 8D 36 C9 A9 00 8D AC
14A9:38 C9 AD 0D 15 0A 2E 38 4C
14B1:C9 0A 2E 38 C9 0A 2E 38 95
14B9:C9 8D 37 C9 18 6D 35 C9 58
14C1:8D 35 C9 AD 36 C9 6D 38 FD
14C9:C9 8D 38 C9 8D 36 C9 BD 74
14D1:C3 C9 29 07 18 6D 35 C9 8E
14D9:8D 35 C9 AD 36 C9 69 60 36
14E1:8D 36 C9 8E 41 C9 BD 63 90
14E9:C9 29 03 AA AD 35 C9 85 A7
14F1:FD AD 36 C9 85 FE A0 00 51
14F9:B1 FD 5D 08 15 91 FD AE 40
1501:41 C9 68 A8 68 AA 60 C0 45
1509:30 0C 03 00 00 20 40 0D BC
1511:AC 28 C8 AD 12 C8 C9 28 FE
1519:D0 06 A9 04 99 51 C9 60 A8
1521:AD 0E C8 C9 FF D0 04 EE 95
1529:27 C8 60 AD 32 C8 85 FD BD
1531:AD 33 C8 85 FE AC 28 C8 34
1539:B9 51 C9 8D 32 C9 A8 B1 62
1541:FD 8D 31 C9 10 0C EE 27 46
1549:C8 AD 32 C9 8D 31 C9 4C 37
1551:D0 15 C9 04 D0 09 AC 28 CE
1559:C8 99 51 C9 4C 0D 16 EE C6
1561:27 C8 AC 28 C8 4C BD 16 72
1569:AE 28 C8 BD 63 C9 48 BD 7A
1571:C3 C9 48 BD 23 CA 48 BD 67
1579:83 CA 48 BD 51 C9 A8 B9 B9
1581:14 18 9D 23 CA 6D 39 C8 E8
1589:69 03 9D C3 C9 B9 18 18 96
1591:9D 83 CA 6D 38 C8 69 05 58
1599:9D 63 C9 B9 1C 18 8D 19 B5
15A1:C8 20 38 14 BD 63 C9 F0 80
15A9:10 20 64 14 20 FC 0A AD 68
15B1:1B D4 30 05 A9 01 20 2A B0
15B9:10 CE 19 C8 D0 E3 68 9D D3
15C1:83 CA 68 9D 23 CA 68 9D F9
15C9:C3 C9 68 9D 63 C9 60 AC DE
15D1:28 C8 EE 31 C9 20 BD 16 93
15D9:AC 29 C8 C8 C0 06 90 01 8A
15E1:60 8C 29 C8 CE 31 C9 CE AE
15E9:31 C9 4C BD 16 D0 04 EE 6F
15F1:25 C8 60 C9 12 D0 14 8A 10
15F9:48 A2 00 AD C6 17 D9 E0 F2
1601:C8 F0 01 E8 A9 FF 9D E5 EA
1609:CA 68 AA 60 20 40 0D AD D9
1611:00 C8 20 EE 15 A9 77 8D 2E
1619:22 C8 20 81 0C A9 77 AC 47
1621:12 C8 99 E0 C8 A2 00 BD 58
1629:63 C9 F0 06 E8 E0 60 D0 54
1631:F6 60 8E 41 C9 A9 04 20 F3
1639:91 0A A0 00 AE 41 C9 20 F2
1641:CE 0C 6D 38 C8 9D 63 C9 56
1649:9D 34 03 20 CE 0C 6D 39 6E
1651:C8 9D C3 C9 9D 94 03 20 C3
1659:AD 16 9D 23 CA 20 AD 16 10
1661:9D 83 CA 1D 23 CA F0 EF 7E
1669:E8 C8 E0 60 F0 04 C0 20 97
1671:D0 CD 60 B9 51 C9 AA BD E5
1679:14 18 18 79 45 C9 30 24 26
1681:C9 09 F0 20 B0 1E 99 45 6B
1689:C9 8D 23 C8 BD 18 18 18 85
1691:79 4B C9 30 0F C9 09 F0 2C
1699:0B B0 09 99 4B C9 8D 24 F2
16A1:C8 A9 01 60 A9 04 99 51 A4
16A9:C9 A9 00 60 AD 1B D4 30 DE
16B1:04 20 CE 0C 60 20 CE 0C AF
16B9:20 E8 0C 60 AD 31 C9 29 A6
16C1:03 8D 31 C9 AD 00 C8 C9 5E
16C9:0E D0 36 A2 00 BD 57 C9 91
16D1:CD 12 C8 D0 0E BD 5D C9 7B
16D9:CD 31 C9 D0 06 A9 04 99 F7
16E1:51 C9 60 E8 E0 06 D0 E5 6A
16E9:A2 00 BD 57 C9 30 06 E8 98
16F1:E0 06 D0 F6 60 AD 12 C8 40
16F9:9D 57 C9 AD 31 C9 9D 5D 28
1701:C9 AD 31 C9 99 51 C9 AD 95
1709:23 C8 99 45 C9 AD 24 C8 98
1711:99 4B C9 60 A9 07 8D 27 C9
1719:D0 AD 24 C8 0A 0A 0A 0A C2
1721:69 48 8D 42 C9 AD 23 C8 FF
1729:C9 09 F0 35 0A 0A 0A 8D 0A
1731:41 C9 0A 6D 41 C9 69 50 DE
1739:8D 41 C9 90 08 A9 01 8D 37
1741:10 D0 4C 4B 17 A9 00 8D D6
1749:10 D0 AD 41 C9 8D 00 D0 D2
1751:AD 42 C9 8D 01 D0 A9 01 98
1759:8D 15 D0 A9 4F 8D F8 57 42
1761:60 A9 00 8D 10 D0 A9 38 52
1769:8D 00 D0 AD 42 C9 8D 01 A8
1771:D0 60 BD 3E C8 99 3E C8 AD
1779:BD 8F C8 99 8F C8 BD E0 19
1781:C8 99 E0 C8 60 AD 36 C8 12
1789:8D 2E C8 AD 37 C8 8D 2F 25
1791:C8 4C 5C 0C 20 86 17 20 EC
1799:0B 0D AE 13 C8 20 73 17 5C
17A1:AD 23 C8 48 AD 24 C8 48 E4
17A9:AD 06 C8 8D 23 C8 AD 07 C0
17B1:C8 8D 24 C8 20 FC 0B 68 2C
17B9:8D 24 C8 68 8D 23 C8 20 02
17C1:40 0D 4C A3 0C D5 A2 0A 1E
17C9:0A 04 12 00 01 04 0A 0A 76
17D1:0A 06 06 0E 02 02 06 06 52
17D9:0A 00 00 01 01 01 01 03 2E
17E1:03 03 03 03 03 03 03 03 10
17E9:03 03 03 03 03 03 03 03 18
17F1:03 00 02 00 00 01 02 02 EB
17F9:02 02 02 00 01 00 02 02 F7
1801:03 00 00 00 00 01 02 00 BA
1809:00 01 01 01 01 00 00 00 B1
1811:00 00 00 00 01 00 FF FF 49
1819:00 01 00 10 0C 10 0C 04 47
1821:04 04 04 00 01 02 03 00 EA
1829:03 02 01 02 01 00 03 03 AC
1831:02 01 00 01 00 03 02 04 C6
1839:04 00 04 04 04 04 01 02 60
1841:04 04 04 04 03 04 04 04 69
1849:00 03 04 04 04 01 00 03 22
1851:02 04 04 01 04 04 02 04 4C
1859:00 FF 00 01 04 01 FF FF BD
1861:02 04 02 03 FF 03 04 04 1C
1869:04 04 04 04 04 04 04 04 99
1871:04 04 04 04 04 04 04 93 31
1879:11 43 4F 50 59 52 49 47 DF
1881:48 54 20 31 39 38 37 20 3B
1889:43 4F 4D 50 55 54 45 21 85
1891:20 50 55 42 4C 49 43 41 04
1899:54 49 4F 4E 53 0D 41 4C B2
18A1:4C 20 52 49 47 48 54 53 36
18A9:20 52 45 53 45 52 56 45 C1
18B1:44 0D 11 11 12 41 4E 59 06
18B9:20 4B 45 59 20 54 4F 20 1C
18C1:43 4F 4E 54 49 4E 55 45 E9
18C9:00 00 81 4C 0F 03 FE FE 70
18D1:FE FE FE 03 FB FB FB EA 04
18D9:A6 03 EF EF EF EF AF FE B8
18E1:FE FE 01 FE 03 95 A6 AA C7
18E9:01 AA 03 AF AF AF 01 AF 8E
18F1:05 FA FB FB FB FB FB 02 6C
18F9:AA 02 55 7D 7D 02 AF EF C5
1901:EF EF EF EF FB FB FB FB E7
1909:FB FA 02 7D 7D 55 02 AA FF
1911:02 EF EF EF EF EF AF 0C E7
1919:E7 E7 E7 E7 E7 E7 10 E7 9B
1921:E7 E7 E7 E7 E7 11 FA 07 3D
1929:AA 07 AF F5 07 55 07 5F C2
1931:0A FA FD 07 7F AF D7 FA BC
1939:10 FD 07 7F AF DF 13 FD EC
1941:FA 04 DF AF 7F 03 FD FA E7
1949:04 D7 AF 7F 10 F5 01 FA B6
1951:FA FA 03 55 EB AA AA AA 7F
1959:03 5F 01 AF AF AF FA FA 2D
1961:FA FA FA 03 AA AA AA AA 5F
1969:AA 03 AF AF AF AF AF 06 44
1971:FA FA FA FA FA 03 AA AA D2
1979:AA AA AA 03 DF DF DF 9F 0F
1981:9F FA FA FA FA FA 03 AA C5
1989:AA AA AA AA 03 9F 9F DF 71
1991:DF DF 06 FA FA FA FA FA D0
1999:03 AA AA AA AA AA 03 AF AD
19A1:AF AF AF AF FA FA FA 01 43
19A9:F5 03 AA AA AA EB 55 03 4A
19B1:AF AF AF 01 5F 06 F7 F7 A8
19B9:F7 F6 F6 03 AA AA AA AA B4
19C1:AA 03 AF AF AF AF AF F6 8D
19C9:F6 F7 F7 F7 03 AA AA AA B6
19D1:AA AA 03 AF AF AF AF AF AA
19D9:10 FD F6 DA 03 DF 6F AF 46
19E1:AF AF 01 FD F6 DA 04 6A 6D
19E9:AA AA AA AA 04 AF AF AF FC
19F1:07 F7 F9 FA FA FA 05 7F E1
19F9:9F A7 08 FA FA FA FA 04 54
1A01:A9 AA AA AA 05 7F 9F A7 C1
1A09:07 DA F6 FD 05 AA AA AA 09
1A11:6A DA 03 AF AF AF AF AF D7
1A19:08 F6 FD 06 AF AF 6F DF 2A
1A21:07 FA FA FA FA FA 03 AA 1B
1A29:AA AA A9 A7 03 A7 9F 7F 82
1A31:02 FA FA F9 F7 04 9F 7F B2
1A39:14 FD F5 03 F7 D5 55 55 FC
1A41:55 05 7F 5F 57 FA FA 06 E9
```

```
1A49:AA AA 06 AB AB 0A FE FE 7B
1A51:FE FE 04 7F 5F 57 55 08 48
1A59:FE FE FE FE FE 03 55 55 A0
1A61:57 5F 7F 03 7F 0D FA FA 5A
1A69:06 AA AA 06 AB AB F5 FD F6
1A71:06 55 55 55 D5 F7 03 57 E9
1A79:5F 7F 11 FD F5 D5 55 04 F4
1A81:BF BF BF BF FD 07 55 55 85
1A89:D5 F5 FD 03 BF BF BF BF 52
1A91:BF 0E FB F7 FB F7 FB 10 EF
1A99:EA E6 A6 66 6A AA 04 BF FD
1AA1:7F 7F BF 07 F9 FA FA 05 94
1AA9:BF BF AB 08 F9 FA FA F9 4F
1AB1:04 66 AB BF BF 04 6F 12 F1
1AB9:AA 6A 66 A6 E6 03 BF 7F 57
1AC1:7F BF 09 EA FB F7 FB F7 25
1AC9:FB 17 FE FE EA 05 6F AF 8B
1AD1:AF F9 07 99 EA FE FE 04 2C
1AD9:6F AF AF 6F 05 D5 DF DF BD
1AE1:DF DE DE DE 01 55 01 EB D2
1AE9:BE 04 57 F7 F7 F7 B7 B7 AF
1AF1:B7 DE DE DF DF DF DF D5 A7
1AF9:03 BE EB 02 55 01 B7 B7 D2
1B01:F7 B7 E7 F7 57 02 D5 DF EB
1B09:DE DE DE DE DE 01 55 01 D6
1B11:AA 02 FE AB 01 57 F7 F7 05
1B19:B7 B7 F7 F7 DE DE DE DE A6
1B21:DE DF D5 01 FB FE FE 03 66
1B29:55 01 F7 F7 F7 B7 B7 F7 CE
1B31:57 03 FE FE FE FE FE 03 98
1B39:AA AA AA AA AA 03 BF BF 10
1B41:BF BF BF 02 FE FE FE FE 50
1B49:FE 03 AA AA AA AA AA 03 18
1B51:BF BF BF BF BF 05 FE FA 56
1B59:EA EA EA 02 AA AA 00 2A 67
1B61:2A 00 03 BF AF AB AB AB 38
1B69:EA EA FA FE 04 2A 2A 2A 66
1B71:AA AA 03 AB AB AF BF 04 62
1B79:00 00 00 00 00 00 0C E7 AF
1B81:80 12 94 00 1E E7 00 12 B1
1B89:94 00 12 97 80 00 00 00 C9
1B91:12 64 80 12 94 80 0C 94 6E
1B99:80 04 94 80 04 63 00 00 59
1BA1:00 06 74 B9 E9 84 A5 02 32
1BA9:64 B9 C4 14 A5 04 E3 25 84
1BB1:E0 00 00 04 00 00 00 00 98
```

Basketball Sam & Ed

See instructions in article on page 12 before typing in.

Program 1: Basketball Sam & Ed

```
0801:0C 08 0A 00 9E 20 32 30 64
0809:36 32 00 00 00 4C 25 09 45
0811:A9 7F 8D 0D DC A9 FF 8D 73
0819:00 DC AD 11 D0 29 80 D0 24
0821:F9 AD 12 D0 D0 FB A9 81 34
0829:8D 0D DC 60 2E 2E 2E 0E 79
0831:0F 20 13 0D 0F 0B 09 0E C8
0839:07 20 09 0E 20 14 08 05 3D
0841:20 01 15 04 09 14 0F 12 4D
0849:09 15 0D 2E 2E 2E 00 2E FF
0851:2E 2E 10 0C 05 01 13 05 1E
0859:20 15 13 05 20 14 12 01 E7
0861:13 08 20 12 05 03 05 10 70
0869:14 01 03 0C 05 13 2E 2E E3
0871:2E 00 2E 2E 2E 0F 16 05 20
0879:12 14 09 0D 05 21 2E 2E C0
0881:2E 00 2E 2E 2E 15 10 03 3A
0889:0F 0D 09 0E 07 20 03 0F 34
0891:0C 09 13 05 15 0D 20 05 BE
0899:16 05 0E 14 13 3A 00 2E A8
08A1:2E 2E 12 05 07 09 0F 0E 6F
08A9:01 0C 20 14 09 04 04 0C EE
08B1:19 17 09 0E 0B 20 03 0F 04
08B9:0E 14 05 13 14 20 37 2F 66
08C1:31 2E 2E 2E 00 2E 2E 2E E1
08C9:0E 01 14 09 0F 0E 01 0C F2
08D1:20 0D 09 03 12 0F 03 0F 68
08D9:0D 10 15 14 05 12 20 13 1C
08E1:08 0F 17 20 37 2F 32 37 B0
08E9:2E 2E 2E 00 2E 2E 2E 08 F0
08F1:0F 07 20 03 01 0C 0C 09 D8
08F9:0E 07 20 03 0F 0E 14 05 E4
0901:13 14 20 38 2F 35 2E 2E 02
0909:2E 00 2E 2E 2E 02 01 13 69
0911:0B 05 14 02 01 0C 0C 20 FC
0919:13 01 0D 20 26 20 05 04 58
0921:2E 2E 2E 00 A9 FF 8D 15 19
0929:D0 A9 80 8D 91 02 A9 01 DF
0931:8D B0 0C 8D BB 0C A9 93 85
0939:20 D2 FF A9 8E 20 D2 FF 45
0941:A9 B0 A2 05 9D C8 13 CA F9
0949:10 FA A9 00 8D CE 13 8D B2
0951:CF 13 A9 05 8D E8 12 20 E9
0959:B8 12 20 A4 15 20 DD 13 92
0961:A9 81 8D 0D DC 20 3B 18 21
0969:8D 0D DC A9 FF 8D 00 DC CE
0971:20 51 16 A9 B0 8D 3B 04 7B
0979:A9 01 8D 69 16 A9 0C 8D E5
0981:20 D0 A9 0A 8D 18 D4 A9 CD
0989:0F 8D 21 D0 A9 E4 8D F8 AC
0991:07 A9 E0 8D F9 07 A9 00 C5
0999:8D 27 D0 A9 00 8D 28 D0 48
09A1:A9 08 8D 29 D0 A9 00 8D 89
09A9:2A D0 A9 0E 8D 2B D0 A9 7F
09B1:00 8D 2C D0 8D 2D D0 8D 0A
09B9:2E D0 A9 01 8D AB 0C 8D 1D
09C1:AA 0C 8D B6 0C 8D B5 0C 57
09C9:A9 00 8D A6 0C 8D B1 0C D2
09D1:A9 00 8D A8 0C 8D B3 0C FE
09D9:A9 32 8D A7 0C A9 1E 8D 4A
09E1:B2 0C A9 32 8D A9 0C 8D 61
09E9:B4 0C 20 75 0A A9 F0 8D 1B
09F1:FD 07 8D FE 07 8D FF 07 DB
09F9:A9 80 85 61 85 63 A9 49 1E
0A01:85 64 A9 03 85 62 A0 05 52
0A09:20 50 0A 20 2A 0A A9 47 98
0A11:85 62 A0 06 20 50 0A 20 6B
0A19:2A 0A A9 8F 85 62 A0 07 F0
0A21:20 50 0A 20 2A 0A 4C BC 6B
0A29:0C 98 AA 0A A8 AD F5 0A 51
0A31:99 96 0C BD FE 0A 2D A5 35
0A39:0C 8D A5 0C A5 6D F0 09 F9
0A41:BD F6 0A 0D A5 0C 8D A5 22
0A49:0C A5 6C 99 95 0C 60 A5 37
0A51:64 18 69 32 8D F5 0A A9 EF
0A59:00 85 6D A5 62 85 6C A5 7E
0A61:61 0A 26 6C 26 6D A5 6C D2
0A69:18 69 18 85 6C A5 6D 69 7D
0A71:00 85 6D 60 A2 3F BD 2A 52
0A79:0E 9D 00 38 9D 40 38 9D 7B
0A81:80 38 9D C0 38 BD 2A 0F BF
0A89:9D 00 3B 9D 40 3B 9D 80 58
0A91:3B 9D C0 3B BD 6A 0F 9D C9
0A99:00 3C CA 10 D9 A2 BF BD AD
0AA1:6A 0E 9D 00 39 9D C0 39 1D
0AA9:CA E0 FF D0 F2 A2 0B BD 5E
0AB1:C3 0A BC CF 0A 8E F4 0A 3D
0AB9:20 DB 0A AE F4 0A CA 10 76
0AC1:EE 60 E1 E3 E7 E8 E9 ED 84
0AC9:EE E2 E3 EB EE EF 01 01 83
0AD1:01 01 01 01 01 00 00 00 DE
0AD9:00 00 A2 00 86 FC 85 FB 71
0AE1:06 FB 26 FC E8 E0 06 D0 34
0AE9:F7 C0 00 D0 03 4C CF 0D 2D
0AF1:4C FB 0D 00 00 01 02 04 D8
0AF9:08 10 20 40 80 FE FD FB 16
0B01:F7 EF DF BF 7F A5 6B F0 61
0B09:14 20 F7 12 A5 6A F0 0E 18
0B11:A5 67 18 69 03 85 67 A5 10
0B19:68 69 00 85 68 60 A5 67 8D
0B21:38 E9 01 85 67 A5 68 E9 D2
0B29:00 85 68 90 01 60 A9 00 93
0B31:85 67 85 68 A9 01 85 6A E1
0B39:60 A5 6B F0 1E A5 69 D0 90
0B41:03 4C F7 0B A5 61 18 65 E3
0B49:65 85 61 A5 62 65 66 85 F4
0B51:62 A5 62 C9 95 90 03 4C 2C
0B59:E3 0B 60 A9 00 8D E2 0B D1
0B61:AD 69 16 F0 01 60 EA BD 97
0B69:00 DC 8D E1 0B 20 63 17 3D
0B71:AD E1 0B 49 FF 29 04 F0 6A
0B79:16 A5 62 F0 12 A5 61 38 81
0B81:E9 64 85 61 A5 62 E9 00 F6
0B89:85 62 A9 01 8D E2 0B AD FB
0B91:E1 0B 49 FF 29 08 F0 18 E7
0B99:A5 62 C9 94 F0 12 A5 61 1A
0BA1:18 69 64 85 61 A5 62 69 D2
0BA9:00 85 62 A9 02 8D E2 0B 1F
0BB1:AD E1 0B 49 FF 29 10 F0 C2
0BB9:1F E6 6B A9 96 85 67 A9 64
0BC1:00 85 6A 85 68 85 66 AE B3
0BC9:E2 0B BD DB 0B 85 65 BD 80
0BD1:DE 0B 85 69 46 66 66 65 5F
0BD9:EA 60 00 32 32 00 00 01 33
0BE1:00 00 A9 94 85 62 A9 00 7F
0BE9:85 61 A9 00 85 69 46 66 15
0BF1:66 65 20 07 17 60 A5 61 EF
0BF9:38 E5 65 85 61 A5 62 E5 F6
0C01:66 85 62 A5 62 C9 FA B0 35
0C09:01 60 A9 00 85 61 85 62 0E
0C11:A9 01 85 69 46 66 66 65 83
0C19:20 07 17 60 A5 6B F0 17 BF
0C21:A5 6A F0 3D A5 63 18 65 E8
0C29:67 85 63 A5 64 65 68 85 2C
0C31:64 A5 64 C9 88 B0 01 60 77
0C39:A9 87 85 64 A9 00 85 63 BA
0C41:A9 00 85 6A 46 68 66 67 8D
0C49:A5 67 4A 4A 4A 4A 4A 4A 56
0C51:4A 05 68 85 6B 20 07 17 36
0C59:A5 65 05 66 F0 02 C6 65 27
0C61:60 A5 63 38 E5 67 85 63 3E
0C69:A5 64 E5 68 85 64 A5 64 1E
0C71:C9 FA B0 01 60 A9 00 85 82
0C79:63 85 64 A9 01 85 6A 46 05
0C81:68 66 67 20 07 17 60 A0 4C
0C89:10 B9 95 0C 99 00 D0 88 82
0C91:10 F7 60 00 00 00 00 00 BB
0C99:00 00 00 00 00 00 00 00 B1
0CA1:00 00 00 00 00 00 00 00 B9
0CA9:00 00 00 00 00 00 00 00 C1
0CB1:00 00 00 00 00 00 00 00 C9
0CB9:00 00 00 A2 0A BD A6 0C 9C
0CC1:95 61 CA 10 F8 20 E1 FF 63
0CC9:D0 06 A9 81 8D 0D DC 60 D3
0CD1:AD 8D 02 29 02 C9 02 D0 03
0CD9:1B AD 8D 02 29 02 C9 02 A3
0CE1:F0 F7 AD 8D 02 29 02 C9 81
0CE9:02 D0 F7 AD 8D 02 29 02 D9
0CF1:C9 02 F0 F7 EA AD 61 17 F4
0CF9:F0 0D CE 61 17 AD 61 17 07
0D01:D0 05 A9 80 8D 04 D4 A2 CA
0D09:00 20 06 0B 20 3A 0B 20 BC
0D11:1D 0C A2 00 20 06 0B 20 60
0D19:3A 0B 20 1D 0C A2 00 20 F3
0D21:06 0B 20 3A 0B 20 1D 0C C7
0D29:A2 00 20 31 11 A0 01 20 D8
0D31:50 0A 20 2A 0A A0 02 20 93
0D39:50 0A 20 2A 0A A2 0A B5 49
0D41:61 9D A6 0C CA 10 F8 A2 34
0D49:0A BD B1 0C 95 61 CA 10 A6
0D51:F8 A2 01 20 06 0B 20 3A 89
0D59:0B 20 1D 0C A2 01 20 06 C4
0D61:0B 20 3A 0B 20 1D 0C A2 31
0D69:01 20 06 0B 20 3A 0B 20 9D
0D71:1D 0C A2 01 20 31 11 A0 0A
0D79:03 20 50 0A 20 2A 0A A0 26
0D81:04 20 50 0A 20 2A 0A A2 B0
0D89:0A B5 61 9D B1 0C CA 10 7F
0D91:F8 20 11 08 20 88 0C 20 2E
0D99:2F 10 20 99 11 AD 69 16 15
0DA1:F0 18 AD BB 0C 0D B0 0C AD
0DA9:D0 10 8D 69 16 EE 3B 04 5F
0DB1:AD 3B 04 C9 B5 D0 03 4C D1
0DB9:6A 16 AD 2E 10 18 69 10 EA
0DC1:8D 2E 10 D0 06 20 AA 0F 52
0DC9:20 0C 16 4C BC 0C 20 F0 C5
0DD1:0D A0 00 A2 3C 20 E8 0D 06
0DD9:20 E8 0D 20 E8 0D CA CA BD
0DE1:CA CA CA CA 10 EF 60 BD D8
0DE9:00 CF 91 FB C8 E8 60 A0 35
0DF1:3F B1 FB 99 00 CF 88 10 91
0DF9:F8 60 20 F0 0D A0 00 B9 60
0E01:00 CF 48 B9 02 CF 99 00 38
0E09:CF 68 99 02 CF C8 C8 C8 76
0E11:C0 42 D0 EB A0 3F B9 00 6C
0E19:CF A2 07 4A 26 FD CA 10 1A
0E21:FA A5 FD 91 FB 88 10 EE 0E
0E29:60 00 00 00 00 00 00 00 75
```

```
0E31:64 00 03 CF 80 07 9E 00 3A
0E39:17 3C F0 17 39 F8 37 39 D4
0E41:F8 33 33 F0 13 33 C4 20 2B
0E49:00 1C 32 33 FC 13 39 F8 85
0E51:13 3C F0 07 3E 00 03 9F 2C
0E59:C0 00 CE 00 00 00 00 00 AF
0E61:00 00 00 00 00 00 00 00 7D
0E69:00 00 00 00 00 00 00 00 85
0E71:00 00 00 00 00 00 04 40 D5
0E79:00 08 80 00 02 20 00 03 3B
0E81:30 00 00 00 01 00 00 03 C0
0E89:00 F0 03 81 10 01 F9 E0 B3
0E91:00 F0 00 00 30 00 00 00 6B
0E99:00 00 00 00 00 FC 00 07 B0
0EA1:EE 00 06 0C 00 07 1F 80 91
0EA9:00 00 00 00 00 00 00 00 C5
0EB1:00 00 00 00 00 00 04 40 16
0EB9:00 08 80 00 02 20 00 03 7B
0EC1:30 00 00 00 01 00 00 03 01
0EC9:00 F0 03 81 10 03 E1 E0 CB
0ED1:01 F0 00 00 60 00 00 00 AD
0ED9:00 01 86 00 03 CE 00 07 61
0EE1:8E 00 03 07 E0 03 C7 80 39
0EE9:00 00 00 00 00 00 00 00 06
0EF1:00 00 00 00 00 00 04 40 56
0EF9:00 08 80 00 02 20 00 03 BB
0F01:30 00 00 00 01 00 00 03 42
0F09:00 F0 03 81 10 01 F9 E0 35
0F11:00 F0 00 00 30 00 00 00 EC
0F19:00 01 9E 00 00 E7 00 00 EA
0F21:E3 B8 00 C1 E0 01 F8 80 F8
0F29:00 00 00 00 00 00 00 00 47
0F31:00 00 00 3E 00 00 0E 04 53
0F39:00 00 0E 00 00 1D 00 1C A9
0F41:08 00 33 00 00 1B 00 00 36
0F49:06 00 00 00 00 21 80 00 EF
0F51:45 80 00 84 0C 00 C3 0C 6E
0F59:00 E0 20 04 C0 1C 0E 00 86
0F61:00 7C 00 00 00 00 00 00 9E
0F69:00 3F FF FF 13 14 AA 1C B1
0F71:C6 4A 0A A9 34 0B 14 C4 1C
0F79:0C AB 5C 05 44 64 06 AA CE
0F81:88 03 11 58 02 AA A8 02 5A
0F89:44 48 01 AA B0 01 AA B0 36
0F91:01 55 50 01 AA B0 01 65 1F
0F99:10 00 ED B0 00 92 40 00 53
0FA1:00 00 00 00 00 00 00 00 BF
0FA9:FF EA AE 94 0C A0 00 BD 42
0FB1:D0 0F 99 07 04 C8 E8 C0 75
0FB9:19 D0 F4 EE 94 0C AD 94 EA
0FC1:0C C9 3C F0 01 60 A9 00 CB
0FC9:8D 94 0C 20 B8 12 60 20 46
0FD1:20 20 20 20 20 20 20 20 EF
0FD9:20 20 20 20 20 20 20 20 F7
0FE1:20 20 20 20 20 20 20 20 FF
0FE9:20 20 20 20 20 04 12 09 64
0FF1:0E 0B 20 02 0C 15 05 20 DC
0FF9:10 0F 10 20 20 20 20 20 C9
1001:20 20 20 20 20 20 20 20 21
1009:20 20 20 20 20 20 20 20 29
1011:20 20 20 20 20 20 20 20 31
1019:20 20 20 20 20 20 20 20 39
1021:20 20 20 20 20 20 20 20 41
1029:20 20 20 20 20 00 AD 07 CA
1031:11 D0 3A EE 08 11 AE 08 2E
1039:11 E0 3C D0 0A AD 69 16 9E
1041:F0 3B A9 FF 8D 08 11 BD 4A
1049:CB 10 8D 07 11 BD 8F 10 24
1051:AA BD 1D 11 8D 07 D4 BD DA
1059:09 11 8D 08 D4 A9 69 8D 22
1061:0C D4 A9 56 8D 0D D4 A9 4B
1069:11 8D 0B D4 60 CE 07 11 81
1071:AD 07 11 C9 03 D0 F5 A9 D9
1079:10 8D 0B D4 60 A9 3B 8D 61
1081:08 11 A9 00 8D 07 D4 8D DE
1089:08 D4 8D 07 11 60 06 01 1C
1091:13 00 01 02 00 03 02 04 8F
1099:03 05 03 02 00 00 01 06 05
10A1:00 03 06 04 03 05 04 03 BA
10A9:07 03 03 03 03 07 01 03 D7
10B1:03 03 03 03 07 06 03 11 0C
10B9:00 01 06 03 08 03 08 09 70
10C1:05 03 07 03 12 04 08 07 ED
10C9:03 06 10 10 10 10 08 18 D8
10D1:10 10 08 18 10 20 10 08 A9
10D9:40 10 08 18 10 10 08 18 89
10E1:10 20 10 08 40 10 10 10 07
10E9:10 10 08 28 10 10 10 10 8A
10F1:10 08 28 08 08 08 08 08 1A
10F9:10 08 10 08 18 10 08 08 BF
1101:08 08 08 08 08 18 00 FF 4B
1109:10 12 15 19 1C 21 13 16 8D
1111:1D 1F 23 38 3B 2C 32 27 87
1119:25 0E 1A 11 C3 D1 1F 1E 67
1121:31 87 EF 60 DF A5 86 63 C7
1129:BE C1 3C DF A2 EF 9C C3 72
1131:8A 0A AA A5 6B D0 37 AD 85
1139:E2 0B 29 01 49 01 F0 17 0C
1141:A5 62 4A 4A 29 03 D0 03 B6
1149:18 69 01 18 69 E3 9D F9 83
1151:07 A9 E0 9D FA 07 60 A5 B1
1159:62 4A 4A 29 03 D0 03 18 94
1161:69 01 18 69 E6 9D F9 07 BA
1169:A9 E1 9D FA 07 60 AD E2 34
1171:0B 29 01 49 01 F0 1B A5 BF
1179:62 49 FF 4A 4A 29 03 18 D8
1181:69 EC 9D F9 07 A5 62 4A C4
1189:4A 29 03 18 69 E0 9D FA 02
1191:07 60 A5 62 4C 7C 11 60 01
1199:AD A9 0C 18 69 10 38 ED E9
11A1:B4 0C 90 F3 29 E0 D0 EF D0
11A9:AD A7 0C 18 69 08 38 ED 59
11B1:B2 0C 90 E3 29 F0 D0 DF 0F
11B9:20 B3 16 AD A9 0C CD B4 44
11C1:0C B0 03 4C 29 12 A9 00 20
11C9:8D BA 0C 8D B8 0C 20 F6 E8
11D1:16 8D B7 0C AD A7 0C 18 56
11D9:69 08 38 ED B2 0C 29 0F BF
11E1:4A AA BD 8E 12 85 FB BD D0
11E9:96 12 85 FC AD B9 0C D0 99
11F1:03 20 17 12 AD B5 0C 18 16
11F9:65 FB 8D B5 0C AD B6 0C 6B
1201:65 FC 8D B6 0C 10 09 20 07
1209:17 12 A9 00 8D B9 0C 60 3E
1211:A9 01 8D B9 0C 60 A9 00 CC
1219:38 ED B5 0C 8D B5 0C A9 51
1221:00 ED B6 0C 8D B6 0C 60 18
1229:A9 00 8D AF 0C 8D AD 0C CC
1231:8D AF 0C 20 F6 16 8D AC 63
1239:0C AD B2 0C 18 69 07 38 92
1241:ED A7 0C 29 0F 4A AA BD 0F
1249:8E 12 85 FB BD 96 12 85 9B
1251:FC AD AE 0C D0 03 20 7C 45
1259:12 AD AA 0C 18 65 FB 8D E3
1261:AA 0C AD AB 0C 65 FC 8D CB
1269:AB 0C 10 09 20 7C 12 A9 B9
1271:00 8D AE 0C 60 A9 01 8D C8
1279:AE 0C 60 A9 00 38 ED AA 06
1281:0C 8D AA 0C A9 00 ED AB F9
1289:0C 8D AB 0C 60 1B 09 03 D1
1291:01 FE FC F6 E4 00 00 00 2C
1299:00 FF FF FF FF 2D 08 2D AF
12A1:08 50 08 73 08 60 18 7D 85
12A9:18 A8 18 2D 08 83 08 A0 D8
12B1:08 C6 08 ED 08 0B 09 A2 8C
12B9:53 A9 20 9D D0 0F CA 10 38
12C1:FA AD E8 12 0A AA BD 9E 22
12C9:12 85 FB BD 9F 12 85 FC 01
12D1:A0 00 B1 FB F0 07 99 EE 02
12D9:0F C8 4C D3 12 AE E8 12 AD
12E1:BD E9 12 8D E8 12 60 00 CA
12E9:09 02 08 03 01 06 01 0C 72
12F1:09 0A 0B 0C 01 60 8E D0 B6
12F9:13 A5 6A F0 F8 A5 64 4A DE
1301:C9 1C D0 F1 A2 02 8E D1 58
1309:13 AE D1 13 BD DA 13 18 67
1311:69 08 38 E5 62 85 FB 90 05
1319:39 29 F0 D0 35 20 07 17 A0
1321:A5 FB 4A A8 B9 D2 13 85 B1
1329:69 E6 65 A5 65 4A 4A 05 B2
1331:66 F0 0D A5 65 38 E9 01 A3
1339:85 65 A5 66 E9 00 85 66 57
1341:46 68 66 67 46 68 66 67 EF
1349:A5 67 05 68 F0 11 A9 00 62
1351:85 6A AE D1 13 CE D1 13 52
1359:CA 10 AE AE D0 13 60 20 5D
1361:42 17 A9 00 85 65 85 66 D6
1369:E6 64 E6 64 AE D0 13 FE 1D
1371:CE 13 FE CE 13 FE CC 13 D1
1379:FE CC 13 BD CC 13 C9 BA 91
1381:D0 1C A9 B0 9D CC 13 FE 9C
1389:CA 13 BD CA 13 C9 BA D0 44
1391:0D A9 B0 9D CA 13 FE C8 02
1399:13 A9 04 8D E8 12 AD C8 C0
13A1:13 8D 82 04 AD CA 13 8D 91
13A9:83 04 AD CC 13 8D 84 04 F0
13B1:AD C9 13 8D 92 04 AD CB 28
13B9:13 8D 93 04 AD CD 13 8D D7
13C1:94 04 A9 01 4C 51 13 B0 F6
13C9:B0 B0 B0 B0 B0 00 00 00 1B
13D1:00 01 00 00 00 01 01 01 3F
13D9:00 03 47 8F A9 43 A2 25 67
13E1:9D 09 06 9D C1 07 9D 81 9A
13E9:06 9D 49 07 CA 10 F1 A9 38
13F1:20 A2 14 9D 89 06 9D 51 1E
13F9:07 CA 10 F7 A9 A0 A2 18 05
1401:9D 2F 04 9D 57 04 9D 7F A3
1409:04 CA 10 F4 A2 0A A9 79 41
1411:9D AE 04 A9 E2 9D 9E 05 9E
1419:CA 10 F3 A9 40 A2 04 9D F5
1421:01 05 9D 79 05 CA 10 F7 C1
1429:A2 02 9D BA 06 9D 32 07 94
1431:CA 10 F7 20 26 15 A2 00 8E
1439:BD 57 14 A8 BD 58 14 85 20
1441:FB BD 59 14 85 FC 05 FB 69
1449:F0 0B 98 A0 00 91 FB E8 F0
1451:E8 E8 4C 39 14 60 5F 05 2B
1459:04 DF 06 04 5F 2E 04 5F 97
1461:57 04 20 7F 04 5F 80 04 D4
1469:69 21 04 69 48 04 69 6F 3A
1471:04 69 96 04 6C A0 04 6C 63
1479:AD 04 7B B9 04 7B C7 04 26
1481:70 00 05 6E 06 05 6D 78 01
1489:05 7D 7E 05 5D 28 05 5D A6
1491:50 05 5D 2E 05 5D 56 05 01
1499:63 7B 05 7C 9D 05 7C A4 59
14A1:05 20 A3 05 40 E3 06 40 F6
14A9:0B 07 7E A2 05 7C A4 05 7B
14B1:7E A9 05 70 09 06 72 1B 8B
14B9:06 6E 2E 06 93 5A 04 81 36
14C1:5B 04 8D 5C 04 85 6A 04 1F
14C9:84 6B 04 71 BB 06 55 B9 01
14D1:06 49 BD 06 55 E2 06 49 F2
14D9:E4 06 4A 0A 07 4B 0C 07 64
14E1:4A 31 07 72 33 07 4B 35 05
14E9:07 6B 49 07 7D 51 07 6D B6
14F1:65 07 73 6E 07 6D C1 07 5C
14F9:5D E1 06 5D 09 07 5D E5 E4
1501:06 5D 0D 07 71 D3 07 7D FD
1509:E6 07 6B 81 06 6E 89 06 F0
1511:70 9D 06 20 97 04 E9 20 5E
1519:04 73 A6 06 20 7A 05 20 6C
1521:7C 05 00 00 00 A2 00 A9 FE
1529:A0 9D C8 04 9D EF 04 8A A3
1531:18 69 28 AA E0 F0 D0 EF CD
1539:A2 00 A9 76 9D D5 04 8A 28
1541:18 69 28 AA E0 C8 D0 F2 40
1549:A2 00 A9 75 9D E1 04 A9 77
1551:E1 9D 7F 06 9D DF 05 A9 44
1559:61 9D B8 05 9D 58 06 8A E7
1561:18 69 28 AA E0 C8 D0 E2 50
1569:A2 00 A9 5D 9D 09 06 9D AA
1571:F9 06 9D 89 06 9D 1B 06 49
1579:9D 0B 07 9D 9D 06 9D 2E 5E
1581:06 9D 1E 07 8A 18 69 28 F9
1589:AA E0 F0 D0 DD A9 00 8D 8F
1591:20 D0 8D 21 D0 85 C6 A2 90
1599:0D BD 52 18 9D 0D 04 CA 79
15A1:10 F7 60 A9 06 A2 9F 9D 10
15A9:00 D8 CA E0 FF D0 F8 A9 50
15B1:00 A2 C7 9D A0 D8 9D 68 63
15B9:D9 CA E0 FF D0 F5 A2 EF 33
15C1:A9 0C 9D 08 DA 9D F8 DA 12
15C9:CA E0 FF D0 F5 A2 00 A9 82
15D1:00 9D 08 DA 9D A8 DA 9D F4
15D9:2F DA 9D CF DA 8A 18 69 9D
15E1:28 AA E0 C8 D0 E9 A9 02 F6
15E9:A2 18 9D 07 D8 CA 10 FA 9C
15F1:A9 00 8D 3B D8 A2 04 9D 4D
15F9:59 D8 9D 69 D8 CA 10 F7 5B
1601:A2 04 A9 02 9D 89 D8 CA 64
1609:10 FA 60 AD 69 16 D0 3F 67
1611:CE 8D 04 AD 8D 04 C9 AF 23
1619:D0 35 A9 B9 8D 8D 04 CE 45
1621:8C 04 AD 8C 04 C9 AF D0 8A
```

```
1629:26 A9 B5 8D 8C 04 CE 8A FE
1631:04 AD 8A 04 C9 AF D0 17 22
1639:A9 B9 8D 8A 04 CE 89 04 75
1641:AD 89 04 C9 AF D0 08 20 B4
1649:51 16 A9 01 8D 69 16 60 87
1651:A9 B1 8D 89 04 A9 B0 8D BE
1659:8A 04 A9 B0 8D 8C 04 8D 40
1661:8D 04 A9 BA 8D 8B 04 60 39
1669:00 A9 00 8D 00 D4 8D 01 48
1671:D4 8D 07 D4 8D 08 D4 8D 5D
1679:0E D4 8D 0F D4 A9 B4 8D C8
1681:3B 04 AD CE 13 CD CF 13 71
1689:D0 08 A9 03 8D E8 12 4C 06
1691:BB 0D A0 07 A9 00 85 C6 82
1699:A2 FF 20 11 08 A5 C6 D0 61
16A1:0E CA D0 F6 88 D0 F1 A9 26
16A9:00 8D 00 D4 8D 01 D4 4C EC
16B1:25 09 AD B0 0C 0D BB 0C 8B
16B9:F0 42 AD 61 17 D0 35 A9 CA
16C1:8F 8D 18 D4 A9 10 8D 05 17
16C9:D4 8D 13 D4 A9 64 8D 06 73
16D1:D4 A9 5A 8D 14 D4 A9 10 4E
16D9:8D 12 D4 A9 15 8D 04 D4 42
16E1:A9 18 8D 01 D4 A9 0D 8D 9F
16E9:0F D4 A9 14 8D 04 D4 A9 19
16F1:10 8D 12 D4 60 A9 C8 AE 03
16F9:69 16 D0 01 60 EE AA 0C AA
1701:EE B5 0C A9 A0 60 AD 61 73
1709:17 D0 E9 A9 8F 8D 18 D4 86
1711:A9 10 8D 05 D4 8D 13 D4 F1
1719:A9 64 8D 06 D4 A9 5A 8D D6
1721:14 D4 A9 10 8D 12 D4 A9 CC
1729:15 8D 04 D4 A9 10 8D 01 BC
1731:D4 A9 05 8D 0F D4 A9 10 DC
1739:8D 04 D4 A9 10 8D 12 D4 14
1741:60 A9 0F 8D 18 D4 A9 77 A3
1749:8D 05 D4 A9 77 8D 06 D4 87
1751:A9 28 8D 01 D4 A9 81 8D FD
1759:04 D4 A9 05 8D 61 17 60 C4
1761:00 60 AD 00 18 F0 FA AD 85
1769:B0 0C F0 05 AD AF 0C F0 96
1771:F0 E0 00 F0 EC AD 01 18 97
1779:F0 0F CE 01 18 AD 03 18 63
1781:8D E1 0B 20 05 18 4C 62 D5
1789:17 8E 02 18 A2 02 E0 01 88
1791:F0 5E BD DA 13 18 69 08 09
1799:38 E5 62 85 FB 90 10 29 6D
17A1:F0 D0 0C 20 C6 17 AE 02 F1
17A9:18 A9 41 8D 01 18 60 CA 43
17B1:10 DC AE 02 18 AD A7 0C E7
17B9:CD B2 0C 90 04 20 D5 17 69
17C1:60 20 E3 17 60 BD D2 17 CC
17C9:8D 03 18 A9 14 8D 01 18 0E
17D1:60 17 00 1B A9 2D 8D 01 C5
17D9:18 A9 17 8D 03 18 20 05 F7
17E1:18 60 A9 2D 8D 01 18 A9 86
17E9:1B 8D 03 18 20 05 18 60 90
17F1:AD A7 0C CD DB 13 D0 B7 C3
17F9:A9 64 8D 01 18 D0 B0 01 3E
1801:00 00 00 00 AD A7 0C 38 8D
1809:ED B2 0C C9 14 F0 05 C9 33
1811:EB F0 01 60 AD AF 0C F0 CE
1819:20 AD A9 0C C9 32 90 19 0C
1821:AD A7 0C CD B2 0C 90 07 5E
1829:A9 07 8D E1 0B D0 05 A9 0F
1831:0B 8D E1 0B A9 00 8D 01 A0
1839:18 60 20 E4 FF C9 31 F0 5A
1841:0A C9 32 D0 F5 A9 00 8D 20
1849:00 18 60 A9 01 8D 00 18 7C
1851:60 31 20 10 0C 01 19 05 9E
1859:12 20 0F 12 20 32 3F 12 F7
1861:2E 20 01 0E 04 05 12 13 1D
1869:0F 0E 20 26 20 04 2E 20 98
1871:08 05 0E 13 0C 05 19 20 A0
1879:0A 12 2E 00 03 0F 10 19 86
1881:12 09 07 08 14 20 31 39 1B
1889:38 37 20 03 0F 0D 10 15 B9
1891:14 05 21 20 10 15 02 0C 18
1899:09 03 01 14 09 0F 0E 13 24
18A1:2C 20 09 0E 03 2E 00 01 C3
18A9:0C 0C 20 12 09 07 08 14 90
18B1:13 20 12 05 13 05 12 16 EC
18B9:05 04 00 00 00 00 00 00 6D
```

Program 2: Basketball Sam & Ed Customizer Loader

```
SF 10 POKE 53280,0:POKE 53281,
      0
JC 20 PRINT"{CLR}{2 DOWN}{BLK}
      LOAD"CHR$(34)"SAM & ED"C
      HR$(34)",8"
AA 30 PRINT"{4 DOWN}POKE 44,10
      0:POKE 25600,0"
DK 40 PRINT"{2 DOWN}LOAD"CHR$(
      34)"S & E CUSTOMIZER"CHR
      $(34)",8"
DX 50 PRINT"{4 DOWN}RUN"
GB 60 PRINTSPC(2)"{8}JUST A MO
      MENT ... LOADING CUSTOMI
      ZER{BLK}{HOME}";
DX 70 FORI=1TO4:POKE630+I,13:N
      EXT:POKE198,I
```

Program 3: Basketball Sam & Ed Customizer

```
FG 10 REM SAM & ED CUSTOMIZER
MX 20 REM COPYRIGHT 1987 - COM
      PUTE! PUBLICATIONS, INC.
       ALL RIGHTS RESERVED
RC 30 POKE 53280,2:POKE 53281,
      12:BK=1
AK 40 GOSUB 550
MA 50 PRINT"{DOWN}CHANGE COLOR
      S (Y/N)";:INPUT A$:IF A$
      <>"Y" THEN 230
CK 60 PRINT"{DOWN}{3 RIGHT}PRE
      SS SPACEBAR TO CHOOSE CO
      LORS"
KG 70 PRINT"{3 RIGHT}AND THEN
      {SPACE}PRESS RETURN TO S
      ELECT"
HQ 80 PRINT"{DOWN}BORDER COLOR
      {5 SPACES}";:GOSUB 600
GR 90 BD=X:POKE 53280,X
RM 100 PRINT"BACKGROUND COLOR
      {SPACE}";:GOSUB 600
HA 110 BK=X:POKE 53281,X:IF BK
      =0 THEN POKE 646,1
FC 120 PRINT"{DOWN}SAM'S BODY
      {SPACE}COLOR ";:GOSUB 6
      00
RB 130 SB=X
EM 140 PRINT"SAM'S LEG COLOR
      {2 SPACES}";:GOSUB 600
HF 150 SL=X
DX 160 PRINT"{DOWN}ED'S BODY C
      OLOR{2 SPACES}";:GOSUB
      {SPACE}600
FA 170 EB=X
JJ 180 PRINT"ED'S LEG COLOR
      {3 SPACES}";:GOSUB 600
XF 190 EL=X
MG 200 POKE 2431,BD:POKE 2441,
      BK
EE 210 POKE 2461,SL:POKE 2466,
      SB
SH 220 POKE 2471,EL:POKE 2476,
      EB
JP 230 POKE 53280,2:POKE 53281
      ,12:POKE 646,0
KG 240 GOSUB 550:PRINT"{DOWN}C
      HANGE TIMER (Y/N)";:INP
      UT D$:IF D$="Y" THEN GO
      SUB 690
SS 250 GOSUB 550:PRINT"{DOWN}C
      HANGE MESSAGES (Y/N)";:
      INPUT A$:IF A$<>"Y" THE
      N 450
QD 260 GOSUB550
DH 270 PRINT"{DOWN}UPCOMING EV
      ENT #1 (37 CHARS MAX.)"
      :INPUT A$
XS 280 IF LEN(A$)<37 THEN A$=A
      $+" ":GOTO 280
AF 290 A$=LEFT$(A$,37)
SA 300 FOR Y=1 TO 37:NC=ASC(MI
      D$(A$,Y,1)):IF NC>63 TH
      EN NC=NC-64
XC 310 POKE 2207+Y,NC
AB 320 NEXT Y
DC 330 PRINT"{DOWN}UPCOMING EV
      ENT #2 (37 CHARS MAX.)"
      :INPUT A$
FQ 340 IF LEN(A$)<38 THEN A$=A
      $+" ":GOTO 340
SK 350 A$=LEFT$(A$,38)
BR 360 FOR Y=1 TO 38:NC=ASC(MI
      D$(A$,Y,1)):IF NC>63 TH
      EN NC=NC-64
FE 370 POKE 2245+Y,NC
JG 380 NEXT Y
EM 390 PRINT"{DOWN}UPCOMING EV
      ENT #3 (29 CHARS MAX.)"
      :INPUT A$
KJ 400 IF LEN(A$)<29 THEN A$=A
      $+" ":GOTO 400
ER 410 A$=LEFT$(A$,29)
PK 420 FOR Y=1 TO 29:NC=ASC(MI
      D$(A$,Y,1)):IF NC>63 TH
      EN NC=NC-64
PK 430 POKE 2284+Y,NC
FK 440 NEXT Y
XH 450 PRINT"{2 DOWN}{6 RIGHT}
      INSERT A NEW FORMATTED
      {SPACE}DISK":FOR D=1 TO
       1000:NEXT
QP 460 PRINT"{DOWN}{RVS}
      {2 SPACES}PRESS ANY KEY
       TO SAVE CUSTOM SAM & E
      D {OFF}"
KR 470 GET DH$
FE 480 IF DH$="" THEN 470
XJ 490 POKE 44,8:POKE 46,25
JQ 500 PRINT"{DOWN}JUST A MOME
      NT..."
MC 510 PRINT"SAVING CUSTOM SAM
       & ED"
MG 520 OPEN 1,8,15,"S0:CUSTOM
      {SPACE}SAM & ED":CLOSE
      {SPACE}1
HH 530 SAVE"CUSTOM SAM & ED",8
JM 540 PRINT"{CLR}{DOWN}{RVS}
      {6 SPACES}PRESS RETURN
      {SPACE}TO PLAY SAM & ED
      {5 SPACES}{OFF}":PRINT"
      {3 DOWN}RUN{3 UP}":END
DX 550 PRINT"{CLR}{BLK}";:PRIN
      TSPC(9)"{RVS}{2 SPACES}
      SAM & ED CUSTOMIZER
      {2 SPACES}{OFF}"
JQ 560 FOR M=1 TO 10:PRINT"
      {DOWN}":NEXT
SD 570 PRINTSPC(5)"UDI COPYRIG
      HT 1987"
QD 580 PRINTSPC(5)"GCH COMPUTE
      ! PUBLICATIONS INC."
FC 590 PRINTSPC(5)"JFK ALL RIG
      HTS RESERVED{HOME}":RET
      URN
SM 600 FOR X=0 TO 15
BC 610 POKE 646,X:PRINT"{RVS}
      {SPACE}{OFF}";"{LEFT}";
      :GET A$:IF A$=CHR$(13)
      {SPACE}THEN POKE 646,0:
      GOTO 670
GA 620 IF A$=CHR$(32) THEN 640
QP 630 GOTO 610
FJ 640 IF X=15 THEN 600
RM 650 NEXT
JR 660 GOTO 610
MM 670 PRINT:IF BK=0 THEN POKE
       646,1
JA 680 RETURN
JQ 690 PRINT"{DOWN}ENTER A NUM
      BER LESS THAN 10";:INPU
```

```
      T H$
QJ 700 IF H$="" THEN 700
SJ 710 H=VAL(H$)+176
SJ 720 POKE 5714,176:POKE 5719
      ,H:RETURN
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Ringside Boxing

See instructions in article on page 15 before typing in.

```
0801:0C 08 0A 00 9E 20 32 34 68
0809:38 35 00 00 00 AD E8 03 0E
0811:D0 03 6C 82 03 CE 47 03 E4
0819:CE 49 03 CE 48 03 AD 4E 28
0821:03 D0 0B 20 0C 09 A9 01 23
0829:8D 4E 03 4C 37 08 20 61 34
0831:09 A9 00 8D 4E 03 AD 48 2B
0839:03 D0 08 20 62 08 A9 0F 97
0841:8D 48 03 AD 47 03 D0 08 55
0849:20 9B 08 A9 13 8D 47 03 4C
0851:AD 49 03 D0 08 20 F7 08 B0
0859:A9 6D 8D 49 03 6C 82 03 B1
0861:60 AD 4D 03 D0 1A AD 4F 80
0869:03 D0 05 A9 C2 8D FA 07 B3
0871:AD 50 03 D0 05 A9 CF 8D D5
0879:FF 07 A9 01 8D 4D 03 60 98
0881:AD 50 03 D0 05 A9 CE 8D E3
0889:FF 07 AD 4F 03 D0 05 A9 15
0891:C3 8D FA 07 A9 00 8D 4D 6C
0899:03 60 EE 4A 03 AD 4A 03 2C
08A1:C9 0A 90 31 A9 00 8D 4A F0
08A9:03 A9 00 8D 4A 03 EE 4B 06
08B1:03 AD 4B 03 C9 06 90 1D EC
08B9:A9 00 8D 4B 03 EE 4C 03 74
08C1:AD 4C 03 C9 03 90 0E A9 D8
08C9:00 8D 4A 03 8D 4B 03 8D E3
08D1:4C 03 8D E8 03 AD 4C 03 73
08D9:18 69 30 8D 36 04 A9 3A 7E
08E1:8D 37 04 AD 4B 03 18 69 E1
08E9:30 8D 38 04 AD 4A 03 18 71
08F1:69 30 8D 39 04 60 AD 40 45
08F9:03 C9 C5 B0 03 EE 40 03 19
0901:AD 41 03 C9 C5 B0 03 EE 1D
0909:41 03 60 AD 4F 03 F0 03 CE
0911:4C A0 09 AD 50 03 F0 01 DE
0919:60 AD 01 DC 29 0F C9 0F DC
0921:D0 01 60 38 C9 08 90 16 11
0929:AD 00 D0 18 C9 2D B0 01 13
0931:60 CE 00 D0 CE 02 D0 CE 23
0939:04 D0 CE 06 D0 60 AD 08 27
0941:D0 38 ED 00 D0 C9 12 B0 0A
0949:01 60 AD 00 D0 38 C9 DC 81
0951:90 01 60 EE 00 D0 EE 02 0A
0959:D0 EE 04 D0 EE 06 D0 60 AE
0961:AD 50 03 F0 03 4C 29 09 72
0969:AD 4F 03 F0 01 60 AD 00 7A
0971:DC 29 0F C9 0F D0 01 60 D8
0979:38 C9 08 90 22 AD 0E D0 D8
0981:18 C9 28 B0 01 60 AD 08 0F
0989:D0 38 ED 00 D0 C9 12 B0 52
0991:01 60 CE 08 D0 CE 0A D0 3D
0999:CE 0C D0 CE 0E D0 60 AD 3F
09A1:0E D0 C9 EB 90 01 60 EE 1F
09A9:08 D0 EE 0A D0 EE 0C D0 9D
09B1:EE 0E D0 60 78 AD 14 03 84
09B9:8D 82 03 AD 15 03 8D 83 C1
09C1:03 A9 0E 8D 14 03 A9 08 62
09C9:8D 15 03 A9 00 8D 4A 03 B0
09D1:8D 4B 03 8D 4C 03 8D 4F 8F
09D9:03 8D 50 03 8D E4 03 8D 9E
09E1:E5 03 A9 01 8D 47 03 8D 0A
09E9:51 03 A9 04 8D 48 03 58 C6
09F1:20 92 14 A9 B7 8D 15 D0 C4
09F9:A9 FF 8D 1C D0 A9 0A 8D 23
0A01:25 D0 A9 09 8D 26 D0 A9 F1
0A09:0D 8D 2B D0 8D 2C D0 8D C5
0A11:2D D0 8D 2E D0 A9 04 8D 47
0A19:27 D0 8D 28 D0 8D 29 D0 09
0A21:8D 2A D0 A9 0B 8D 11 D0 BC
0A29:A9 00 8D 21 D0 A9 8E 20 40
0A31:D2 FF 20 95 0F 20 87 0E 22
0A39:20 56 0E 20 A7 0D A9 C6 42
0A41:8D 40 03 8D 41 03 A9 1B E9
0A49:8D 11 D0 A9 E0 8D 02 DC 3B
0A51:A9 00 8D 3C 03 8D 3D 03 7B
0A59:8D 3E 03 8D 3F 03 A9 01 57
0A61:8D 42 03 8D E8 03 AD 1E D2
0A69:D0 AD 1E D0 AD E8 03 D0 0A
0A71:2D AD 42 03 C9 03 D0 06 02
0A79:20 56 0E 4C DB 0E 20 56 67
0A81:0E 20 A7 0D 20 45 0D 20 BA
0A89:9D 0E EE 42 03 20 D1 0E 3C
0A91:20 82 0D 20 45 0D 20 56 EE
0A99:0E A9 01 8D E8 03 AD 01 C7
0AA1:DC 29 10 D0 41 A5 FB D0 E6
0AA9:3D AD 01 DC 29 0F C9 06 D4
0AB1:D0 0D A9 C4 8D F8 07 A9 FA
0AB9:C5 8D F9 07 4C E0 0A C9 87
0AC1:05 D0 0D A9 C6 8D F8 07 2E
0AC9:A9 C7 8D F9 07 4C E0 0A 2B
0AD1:A9 C8 8D F8 07 A9 C9 8D 2E
0AD9:F9 07 A9 03 8D 57 03 A9 8B
0AE1:01 85 FB 4C F2 0A AD 01 38
0AE9:DC 29 10 F0 04 A9 00 85 14
0AF1:FB AD 00 DC 29 10 D0 41 A9
0AF9:A5 FC D0 3D AD 00 DC 29 5E
0B01:0F C9 0A D0 0D A9 D0 8D 9D
0B09:FC 07 A9 D1 8D FD 07 4C 70
0B11:33 0B C9 09 D0 0D A9 D2 2E
0B19:8D FC 07 A9 D3 8D FD 07 88
0B21:4C 33 0B A9 D4 8D FC 07 04
0B29:A9 D5 8D FD 07 A9 03 8D 8D
0B31:58 03 A9 01 85 FC 4C 46 78
0B39:0B AD 00 DC 29 10 F0 04 7D
0B41:A9 00 85 FC EA AD F8 07 B3
0B49:C9 C0 D0 17 C9 C8 D0 13 26
0B51:AD FC 07 C9 CC D0 0C C9 86
0B59:D4 D0 08 A2 5F 20 94 0E EB
0B61:4C 6A 0A AD F8 07 AA AD 3B
0B69:FC 07 A8 E0 C4 D0 06 CE 27
0B71:40 03 CE 40 03 E0 C6 D0 40
0B79:09 AD 40 03 38 E9 05 8D B8
0B81:40 03 C0 D0 D0 06 CE 41 1B
0B89:03 CE 41 03 C0 D2 D0 09 29
0B91:AD 41 03 38 E9 05 8D 41 72
0B99:03 A2 50 20 94 0E AD 1E 3C
0BA1:D0 C9 B7 F0 11 20 B8 0E 21
0BA9:20 82 0D A2 5F 20 94 0E EE
0BB1:20 02 0E 4C 6A 0A AD F8 AE
0BB9:07 AA AD FC 07 A8 C0 CC AC
0BC1:D0 3C E0 C4 D0 1A AD 41 43
0BC9:03 38 E9 05 8D 41 03 AD 22
0BD1:3C 03 18 69 05 8D 3C 03 3A
0BD9:AD 3D 03 69 00 8D 3D 03 C0
0BE1:E0 C6 D0 1A AD 41 03 38 86
0BE9:E9 0A 8D 41 03 AD 3C 03 87
0BF1:18 69 0A 8D 3C 03 AD 3D 0F
0BF9:03 69 00 8D 3D 03 E0 C0 3D
0C01:D0 3C C0 D0 D0 1A AD 40 40
0C09:03 38 E9 05 8D 40 03 AD 5F
0C11:3E 03 18 69 05 8D 3E 03 80
0C19:AD 3F 03 69 00 8D 3F 03 86
0C21:C0 D2 D0 1A AD 40 03 38 B6
0C29:E9 0A 8D 40 03 AD 3E 03 BC
0C31:18 69 0A 8D 3E 03 AD 3F 62
0C39:03 69 00 8D 3F 03 20 2F 7B
0C41:0E 20 B8 0E 20 82 0D 20 A5
0C49:5F 0D AD F8 07 C9 C0 D0 4B
0C51:15 AD FC 07 C9 CC F0 0E E0
0C59:C9 D4 F0 0A AD 40 03 C9 88
0C61:32 B0 03 20 8E 0C AD FC 1E
0C69:07 C9 CC D0 15 AD F8 07 76
0C71:C9 C0 F0 0E C9 C8 F0 0A FA
0C79:AD 41 03 C9 32 B0 03 20 30
0C81:C2 0C A2 01 20 94 0E 20 F1
0C89:02 0E 4C 6A 0A A9 01 8D DC
0C91:4F 03 AD FA 07 48 A9 BC E0
0C99:8D 15 D0 A9 CA 8D FA 07 FB
0CA1:A9 CB 8D FB 07 20 02 0E BD
0CA9:AD 40 03 4A 8D E7 03 A9 69
0CB1:19 38 ED E7 03 C9 0A 90 84
0CB9:3C A9 01 8D E4 03 4C DB FA
0CC1:0E A9 01 8D 50 03 AD FF 2E
0CC9:07 48 A9 C7 8D 15 D0 A9 35
0CD1:D6 8D FE 07 A9 D7 8D FF D0
0CD9:07 20 02 0E AD 41 03 4A 61
0CE1:8D E7 03 A9 19 38 ED E7 23
0CE9:03 C9 0A 90 08 A9 01 8D B6
0CF1:E5 03 4C DB 0E AA AD 93 0F
0CF9:07 18 69 01 49 80 8D 93 D3
0D01:07 8A 48 A2 FF 20 94 0E 2C
0D09:A2 FF 20 94 0E 68 AA CA F3
0D11:D0 E4 A9 30 8D 93 07 AD 7B
0D19:E4 03 F0 01 60 AD E5 03 1D
0D21:F0 01 60 78 A9 B7 8D 15 E3
0D29:D0 20 02 0E 68 C9 C8 B0 81
0D31:06 8D FA 07 4C 3B 0D 8D 78
0D39:FF 07 A9 00 8D 4F 03 8D 87
0D41:50 03 58 60 A2 FF 20 94 3F
0D49:0E A2 FF 20 94 0E A2 FF 37
0D51:20 94 0E A2 FF 20 94 0E 44
0D59:A2 FF 20 94 0E 60 A2 08 51
0D61:A0 22 18 20 F0 FF AD 3D 79
0D69:03 AE 3C 03 20 CD BD A2 BE
0D71:0D A0 22 18 20 F0 FF AD 72
0D79:3F 03 AE 3E 03 20 CD BD 9F
0D81:60 A2 04 A0 04 18 20 F0 B0
0D89:FF AD 40 03 4A AA A9 00 97
0D91:20 CD BD A2 04 A0 19 18 FD
0D99:20 F0 FF AD 41 03 4A AA 30
0DA1:A9 00 20 CD BD 60 A9 C0 F4
0DA9:8D F8 07 A9 C1 8D F9 07 83
0DB1:A9 C2 8D FA 07 A9 CC 8D B8
0DB9:FC 07 A9 CD 8D FD 07 A9 42
0DC1:CE 8D FF 07 A9 AA 8D 05 2F
0DC9:D0 8D 07 D0 8D 0D D0 8D 6D
0DD1:0F D0 A9 96 8D 01 D0 8D E5
0DD9:03 D0 8D 0B D0 8D 09 D0 AB
0DE1:A9 2E 8D 00 D0 8D 04 D0 A3
0DE9:A9 46 8D 02 D0 8D 06 D0 D5
0DF1:A9 EA 8D 0A D0 8D 0E D0 97
0DF9:A9 D2 8D 08 D0 8D 0C D0 75
0E01:60 AD F8 07 C9 C8 D0 05 60
0E09:CE 57 03 D0 0A A9 C0 8D D5
0E11:F8 07 A9 C1 8D F9 07 AD CC
0E19:FC 07 C9 D4 D0 05 CE 58 8C
0E21:03 D0 0A A9 CC 8D FC 07 6C
0E29:A9 CD 8D FD 07 60 A9 0F 3B
0E31:8D 08 D4 8D 18 D4 A9 0B FB
0E39:8D 0C D4 A9 07 8D 0D D4 B1
0E41:A9 DC 8D 07 D4 A9 81 8D 69
0E49:0B D4 A2 0A 20 94 0E A9 2E
0E51:80 8D 0B D4 60 8D 00 D4 CD
0E59:A9 1E 8D 01 D4 A9 0F 8D 8C
0E61:05 D4 A9 09 8D 06 D4 A9 D2
0E69:78 8D 0E D4 A9 41 8D 0F B0
0E71:D4 A9 CF 8D 18 D4 A9 15 B1
0E79:8D 04 D4 A2 64 20 94 0E FC
0E81:A9 14 8D 04 D4 60 A2 00 D6
0E89:A9 00 9D 00 D4 E8 E0 18 52
0E91:D0 F8 60 A0 00 88 D0 FD 2C
0E99:CA D0 F8 60 AD 40 03 C9 B2
0EA1:BC B0 06 18 69 0A 8D 40 59
0EA9:03 AD 41 03 C9 BC B0 06 B3
0EB1:18 69 0A 8D 41 03 60 AD D2
0EB9:40 03 C9 15 B0 05 A9 15 43
0EC1:8D 40 03 AD 41 03 C9 15 AE
0EC9:B0 05 A9 15 8D 41 03 60 DD
0ED1:AD 42 03 18 69 30 8D F3 52
0ED9:06 60 AD E4 03 F0 42 A2 18
0EE1:09 20 F7 0C A9 31 8D 92 0A
0EE9:07 A2 FF 20 94 0E A9 00 64
0EF1:8D E8 03 85 FB 20 56 0E E2
0EF9:A9 D4 8D FC 07 A9 D5 8D B9
0F01:FD 07 A2 FF 20 94 0E A9 4D
0F09:CC 8D FC 07 A9 CD 8D FD 9E
0F11:07 A2 FF 20 94 0E E6 FB 04
0F19:A5 FB C9 05 D0 D7 4C 81 92
0F21:0F AD E5 03 F0 42 A2 09 FD
0F29:20 F7 0C A9 31 8D 92 07 5D
0F31:A2 FF 20 94 0E A9 00 8D 92
0F39:E8 03 85 FB 20 56 0E A9 1D
```

```
0F41:C8 8D F8 07 A9 C9 8D F9 40
0F49:07 A2 FF 20 94 0E A9 C0 86
0F51:8D F8 07 A9 C1 8D F9 07 2F
0F59:A2 FF 20 94 0E E6 FB A5 BF
0F61:FB C9 05 D0 D7 4C 81 0F 9F
0F69:38 AD 3C 03 ED 3E 03 85 BA
0F71:00 AD 3D 03 ED 3F 03 05 4A
0F79:00 B0 BA F0 03 4C EF 0E 61
0F81:A2 FF 20 45 0D A9 00 85 DD
0F89:C6 A5 C6 F0 FC 20 8A FF D9
0F91:4C B5 09 EA A9 AF 85 FB 26
0F99:A9 0F 85 FC A0 00 B1 FB 35
0FA1:F0 0B 20 D2 FF C8 D0 F6 E7
0FA9:E6 FC 4C 9D 0F 60 93 20 1F
0FB1:12 05 20 20 20 20 20 20 02
0FB9:20 20 20 20 20 20 20 20 D7
0FC1:20 20 20 20 20 20 20 20 DF
0FC9:20 20 20 20 20 20 20 20 E7
0FD1:20 20 20 20 20 20 20 20 EF
0FD9:92 20 12 20 92 20 20 20 03
0FE1:20 20 20 20 20 20 12 97 5B
0FE9:20 20 20 20 9E 20 20 20 FB
0FF1:20 97 20 20 20 20 92 20 D2
0FF9:20 20 20 20 20 20 20 20 18
1001:12 05 20 92 20 20 20 20 7A
1009:20 20 20 12 20 20 92 20 2D
1011:20 12 9C 50 55 52 50 4C 3F
1019:05 92 20 20 12 97 20 20 B5
1021:20 20 20 20 20 20 20 20 41
1029:20 20 92 20 20 12 99 47 79
1031:52 45 45 4E 92 05 20 20 62
1039:12 05 20 92 20 20 20 20 B2
1041:20 20 20 12 20 20 92 20 65
1049:20 12 9C 45 4E 45 52 47 59
1051:92 05 20 20 DF 12 20 20 A9
1059:20 20 20 20 20 20 20 20 79
1061:92 A9 20 20 12 99 45 4E 0B
1069:45 52 47 92 05 20 20 12 CD
1071:05 20 92 20 20 20 20 20 52
1079:20 20 12 20 20 92 20 20 A1
1081:20 39 39 25 20 20 20 20 5B
1089:20 20 20 20 20 20 20 20 A9
1091:20 20 20 20 20 20 39 39 FC
1099:25 20 20 A0 12 05 20 92 D9
10A1:20 20 20 20 20 20 20 12 B3
10A9:20 20 92 20 20 20 20 20 18
10B1:20 20 20 20 20 20 20 20 D1
10B9:20 20 20 20 20 20 20 20 D9
10C1:20 20 20 20 20 20 20 20 E1
10C9:20 12 05 20 92 20 12 9C F6
10D1:50 55 52 50 4C 05 92 20 7A
10D9:12 20 20 92 20 20 20 20 1A
10E1:20 20 20 20 20 20 20 20 02
10E9:20 20 20 20 20 20 20 20 0A
10F1:20 20 20 20 20 20 20 20 12
10F9:20 20 12 20 92 20 12 9C 4C
1101:53 43 4F 52 45 05 92 20 30
1109:12 20 20 92 20 20 20 20 4B
1111:20 20 20 20 20 20 20 20 33
1119:20 20 20 20 20 20 20 20 3B
1121:20 20 20 20 20 20 20 20 43
1129:20 20 12 20 92 20 20 20 1D
1131:20 20 20 20 12 20 20 92 55
1139:20 96 D1 20 D1 20 D1 20 20
1141:D1 20 D1 20 D1 20 D1 20 63
1149:D1 20 D1 20 D1 20 D1 20 6B
1151:D1 20 D1 20 D1 20 D1 12 65
1159:05 20 92 20 20 30 20 20 7C
1161:20 20 12 20 20 92 96 D1 2A
1169:20 D1 20 D1 20 D1 20 D1 8B
1171:20 D1 20 D1 20 D1 20 D1 93
1179:20 D1 20 D1 20 D1 20 D1 9B
1181:20 D1 20 D1 20 12 05 20 BC
1189:92 20 20 20 20 20 20 20 E4
1191:12 20 20 92 20 96 D1 20 11
1199:D1 20 D1 20 D1 20 D1 20 BB
11A1:D1 20 D1 20 D1 20 D1 20 C3
11A9:D1 20 D1 20 D1 20 D1 20 CB
11B1:D1 20 D1 12 05 20 92 20 0E
11B9:12 99 47 52 45 45 4E 05 3A
11C1:92 20 12 05 20 20 92 96 05
11C9:D1 20 12 95 A9 20 92 A9 15
11D1:9E B7 B7 B7 B7 B7 B7 B7 67
11D9:B7 B7 B7 B7 B7 B7 B7 B7 FB
11E1:B7 B7 B7 B7 B7 95 DF 12 26
11E9:20 DF 92 96 D1 20 12 05 08
11F1:20 92 20 12 99 53 43 4F DD
11F9:52 45 92 20 12 05 20 20 EF
1201:92 20 9E CE 96 D1 12 95 EC
1209:20 92 96 20 D1 20 D1 20 89
1211:D1 20 D1 20 D1 20 D1 20 35
1219:D1 20 D1 20 D1 20 D1 20 3D
1221:D1 95 12 20 92 96 D1 9E 09
1229:CD 96 D1 12 05 20 92 20 23
1231:20 20 20 20 20 20 12 20 39
1239:20 92 9E CE 96 D1 9E CE DA
1241:12 95 20 92 9E B7 B7 B7 FB
1249:B7 B7 B7 B7 B7 B7 B7 B7 6D
1251:B7 B7 B7 B7 B7 B7 B7 B7 75
1259:B7 B7 B7 95 12 20 92 9E 6C
1261:CD 96 D1 9E CD 12 05 20 17
1269:92 20 20 30 20 20 20 20 C7
1271:12 20 20 92 96 D1 9E CE DB
1279:1C D1 12 95 20 92 1C D1 11
1281:20 D1 20 D1 20 D1 20 D1 A5
1289:20 D1 20 D1 20 D1 20 D1 AD
1291:20 D1 20 D1 20 95 12 20 F6
1299:92 1C D1 9E CD 96 D1 12 B0
12A1:05 20 92 20 20 20 20 20 86
12A9:20 20 12 20 20 92 9E CE 81
12B1:1C D1 12 9B A9 20 20 20 82
12B9:20 20 20 20 20 20 20 20 DD
12C1:20 20 20 20 20 20 20 20 E5
12C9:20 20 20 20 20 DF 92 1C CD
12D1:D1 9E CD 12 05 20 92 20 4F
12D9:20 20 20 20 20 20 12 20 E1
12E1:20 92 1C D1 12 9B A9 20 CD
12E9:20 20 20 20 20 20 20 20 0E
12F1:20 20 20 20 20 20 20 20 16
12F9:20 20 20 20 20 20 20 20 1E
1301:20 DF 92 1C D1 12 05 20 44
1309:92 20 12 9E 52 4F 55 4E 75
1311:44 92 05 20 12 20 20 92 84
1319:12 9B A9 20 20 20 20 20 48
1321:20 20 20 20 20 20 20 20 47
1329:20 20 20 20 20 20 20 20 4F
1331:20 20 20 20 20 20 20 DF 17
1339:92 12 05 20 92 20 20 20 45
1341:20 20 20 20 12 20 20 9B 72
1349:20 20 20 20 20 20 20 20 6F
1351:20 20 20 20 20 20 20 20 77
1359:20 20 20 20 20 20 20 20 7F
1361:20 20 20 20 20 20 05 20 51
1369:92 20 20 20 31 20 20 20 51
1371:12 20 20 9B 20 20 20 20 48
1379:20 20 20 20 20 20 20 20 9F
1381:20 20 20 20 20 20 20 20 A7
1389:20 20 20 20 20 20 20 20 AF
1391:20 20 05 20 92 20 20 20 E7
1399:20 20 20 20 12 20 20 98 C7
13A1:20 20 20 20 20 20 20 20 C7
13A9:20 20 20 20 20 20 20 20 CF
13B1:20 20 20 20 20 20 20 20 D7
13B9:20 20 20 20 20 20 05 20 A9
13C1:92 20 12 9F 43 4F 55 4E C5
13C9:54 92 05 20 12 20 20 92 45
13D1:20 97 DF 12 20 92 A9 20 C9
13D9:20 20 20 20 20 20 20 20 FF
13E1:20 20 20 20 20 20 20 20 08
13E9:20 20 20 20 20 DF 12 20 F2
13F1:92 A9 20 12 05 20 92 20 DE
13F9:20 20 20 20 20 20 12 20 04
1401:20 92 20 97 20 12 20 20 05
1409:20 20 20 52 49 4E 47 53 D7
1411:49 44 45 20 20 42 4F 58 9A
1419:49 4E 47 20 20 20 20 20 46
1421:92 20 20 12 05 20 92 20 AD
1429:20 20 30 20 20 20 12 20 37
1431:20 92 20 97 20 12 20 92 A7
1439:20 20 20 20 20 20 20 20 61
1441:20 20 20 20 20 20 20 20 69
1449:20 20 20 20 20 20 20 20 71
1451:12 20 92 20 20 12 05 20 52
1459:92 20 20 20 20 20 20 20 BA
1461:12 20 92 20 12 05 20 20 F3
1469:20 20 20 20 20 20 20 20 91
1471:20 20 20 20 20 20 20 20 99
1479:20 20 20 20 20 20 20 20 A1
1481:20 20 20 20 20 20 20 20 A9
1489:20 20 20 20 91 91 92 00 C7
1491:00 A9 AE 85 5F A9 14 85 A1
1499:60 A9 AF 85 5A A9 1A 85 DD
14A1:5B A9 01 85 58 A9 36 85 B5
14A9:59 20 BF A3 60 00 0F F0 CA
14B1:00 3F FC 00 FF FC 00 FF 3D
14B9:5C 00 FF 7C 00 F7 54 00 60
14C1:F5 5C 00 35 5C 00 35 70 0D
14C9:00 D5 70 0F 55 70 35 55 92
14D1:5C 35 75 57 35 5F 57 35 A4
14D9:55 DE 0D 55 FE 03 D5 57 62
14E1:03 7D 5F 03 57 F7 00 D5 77
14E9:5C 00 D5 5C 0F 00 00 00 39
14F1:00 00 00 00 00 00 00 00 1A
14F9:00 00 00 00 00 00 00 00 22
1501:00 00 00 00 00 00 00 00 2B
1509:00 00 00 0C 00 00 FB 00 EB
1511:00 BB 00 00 BB 00 00 BB C3
1519:00 00 BC 00 00 C0 00 00 DD
1521:00 00 00 00 00 00 00 00 4B
1529:00 00 00 00 0B 00 D5 5C B3
1531:00 FF FC 03 AA AC 03 AA E3
1539:AC 03 AA AC 03 AA EC 00 37
1541:FF FC 00 D5 DC 00 D5 DC 77
1549:00 D5 DC 00 D5 DC 00 D5 7C
1551:DC 00 35 DC 00 D7 70 00 9E
1559:D7 70 00 D7 70 00 FF F0 7D
1561:00 EB B0 00 EB B0 00 EA A9
1569:EC 00 FF FF 0F 00 D5 5C 8A
1571:00 FF FC 03 AA AC 03 AA 24
1579:AC 03 AA AC 03 AA EC 00 77
1581:FF FC 00 D5 DC 00 D5 DC B7
1589:00 D5 DC 00 D5 DC 00 D5 BC
1591:DC 00 35 DC 03 D5 DC 0D D4
1599:57 5C 0D 5F 5C 3B 73 FC D1
15A1:EB C3 AC EC 03 AC EB FF B9
15A9:AB FF FF FF FF 00 03 FC AC
15B1:00 0F FF 00 3F FF 00 3F D8
15B9:D7 00 3F DF 00 3D D5 00 56
15C1:3D 57 00 0D 57 00 0D 5C 62
15C9:00 35 5C 03 55 7F 0D 55 15
15D1:55 0D 5D 55 0D 77 55 0D E8
15D9:5D DD 0D 5D 55 0D 75 7D E8
15E1:0D 5F FB 03 55 EB 00 D5 4A
15E9:EB 00 FF 3C 00 00 00 00 CD
15F1:00 00 00 00 00 00 00 00 1C
15F9:00 00 00 00 00 00 30 00 84
1601:00 EC 00 00 EC 00 03 EB C1
1609:00 0D EB FF F5 7C 55 55 97
1611:C0 55 57 00 55 5C 00 55 4F
1619:70 00 FF C0 00 C0 00 00 8C
1621:C0 00 00 C0 00 00 00 00 B9
1629:00 00 00 00 00 00 03 FC 58
1631:00 03 FF 00 0F FF 00 0F A5
1639:F5 00 0F F7 00 0F 75 00 E8
1641:0F 55 00 03 55 00 0D 5F 9E
1649:00 35 5C 00 F5 7F 03 55 57
1651:55 0D 55 55 35 55 55 37 4C
1659:55 55 37 D5 55 0F D5 5F BB
1661:00 F5 77 00 D5 57 00 D5 DB
1669:57 00 D5 5C 00 00 00 00 C1
1671:C0 00 00 C0 00 00 C0 00 8B
1679:00 C0 00 00 80 00 C0 C0 1C
1681:03 B0 C0 03 B0 00 0F AC F3
1689:00 37 AC FF D5 F0 55 57 8D
1691:00 55 5C 00 55 70 00 55 60
1699:C0 00 FF 00 00 00 00 00 26
16A1:00 00 00 00 00 00 00 00 CD
16A9:00 00 00 00 00 00 0F F0 E4
16B1:00 3F FC 00 FF FC 00 FF 41
16B9:5C 00 FF 7C 00 F7 54 00 64
16C1:F5 5C 00 35 5C 00 35 73 14
16C9:00 D5 73 03 55 73 03 5F E7
16D1:5F 03 55 F5 03 55 57 03 97
16D9:F5 5D 03 5D 77 00 D7 DC D6
16E1:00 F5 5C 00 D5 5C 00 D5 0D
16E9:5C 00 D5 5C 0F 00 00 00 3D
16F1:00 00 00 00 00 00 00 00 1E
16F9:00 3C 00 00 EB 00 00 EB 80
1701:00 00 EC 00 00 70 00 00 8E
1709:70 00 00 EC 00 00 AC 00 97
1711:00 EC 00 00 F0 00 00 40 42
1719:00 00 00 00 00 00 00 00 47
1721:00 00 00 00 00 00 00 00 4F
1729:00 00 00 00 00 00 00 00 57
1731:00 00 00 00 00 00 00 00 5F
```

```
1739:00 00 00 00 00 00 00 00 67
1741:00 00 00 00 00 00 00 00 6F
1749:00 00 00 03 00 00 3D CF F1
1751:F0 F5 75 DF F5 55 55 F5 C7
1759:55 D5 FF 55 5D FD 55 77 02
1761:FF 75 55 FF FD 55 3F 0F 6A
1769:55 00 03 FF 00 00 00 00 A2
1771:00 00 00 00 00 00 00 00 9F
1779:00 00 00 00 00 00 00 00 A7
1781:0C 00 00 37 00 00 D5 C0 95
1789:03 55 70 FF 55 70 EA DF BE
1791:70 EA B3 70 EA C3 F0 EA 63
1799:BF B0 EA B5 F2 FA B5 7E F9
17A1:5F ED 7A 57 AD 7A 57 AF 75
17A9:FF FF FF FF 00 00 00 00 D7
17B1:00 00 00 00 00 00 00 00 DF
17B9:00 00 00 00 00 00 00 00 E7
17C1:00 00 00 00 00 00 00 00 EF
17C9:00 00 00 00 00 30 00 00 B8
17D1:EF 00 00 EE 00 00 EE 00 C4
17D9:00 EE 00 00 3E 00 00 03 B8
17E1:00 00 00 00 00 00 00 00 10
17E9:00 00 00 00 0B 0F F0 00 8E
17F1:3F FC 00 3F FF 00 35 FF 5D
17F9:00 3D FF 00 15 DF 00 35 D4
1801:5F 00 35 5C 00 0D 5C 00 3A
1809:0D 57 00 0D 55 F0 35 55 94
1811:5C D5 5D 5C D5 F5 5C B7 4D
1819:55 5C BF 55 70 D5 57 C0 A2
1821:F5 7D C0 DF D5 C0 35 57 35
1829:00 35 57 00 0F 35 57 00 8D
1831:3F FF 00 3A AA C0 3A AA 1C
1839:C0 3A AA C0 3B AA C0 3F FE
1841:FF 00 37 57 00 37 57 00 59
1849:37 57 00 37 57 00 37 57 DE
1851:00 37 5C 00 0D D7 00 0D AF
1859:D7 00 0D D7 00 0F FF 00 D0
1861:0E EB 00 0E EB 00 3B AB F5
1869:00 FF FF 00 0F 35 57 00 95
1871:3F FF 00 3A AA C0 3A AA 5C
1879:C0 3A AA C0 3B AA C0 3F 3F
1881:FF 00 37 57 00 37 57 00 99
1889:37 57 00 37 57 00 37 57 1F
1891:00 37 5C 00 37 57 C0 35 E8
1899:D5 70 35 F5 70 3F CD EC DF
18A1:3A C3 EB 3A C0 3B EA FF C9
18A9:EB FF FF FF FF 00 00 00 CF
18B1:00 00 00 00 00 00 00 00 E1
18B9:00 00 00 00 0C 00 00 3B 85
18C1:00 00 3B 00 00 EB C0 00 8A
18C9:EB 70 00 3D 5F FF 03 55 36
18D1:55 00 D5 55 00 35 55 00 3C
18D9:0D 55 00 03 FF 00 00 03 19
18E1:00 00 03 00 00 03 00 00 7E
18E9:00 00 00 00 00 3F C0 00 98
18F1:FF F0 00 FF FC 00 D7 FC F2
18F9:00 F7 FC 00 57 7C 00 D5 4A
1901:7C 00 D5 70 00 35 70 00 E8
1909:35 5C 00 FD 55 C0 55 55 7A
1911:70 55 75 70 55 DD 70 77 01
1919:75 70 55 75 70 7D 5D 70 C8
1921:EF F5 70 EB 55 C0 EB 57 72
1929:00 3D FF 00 00 00 00 00 AA
1931:00 00 03 00 00 03 00 00 CF
1939:03 00 00 03 03 00 02 0E 47
1941:C0 03 0E C0 03 3A F0 00 45
1949:3A DC 00 0F 57 FF 00 D5 51
1951:55 00 35 55 00 0D 55 00 09
1959:03 55 00 00 FF 00 00 00 62
1961:00 00 00 00 00 00 00 00 93
1969:00 00 00 00 00 3F C0 00 1A
1971:FF C0 00 FF F0 00 5F F0 0B
1979:00 DF F0 00 5D F0 00 55 C5
1981:F0 00 55 C0 00 F5 70 00 9B
1989:35 5C 00 FD 5F 00 55 55 48
1991:C0 55 55 70 55 55 5C 55 39
1999:55 DC 55 57 DC 75 57 F0 2A
19A1:DD 5F 00 D5 57 00 D5 57 B5
19A9:00 35 57 00 00 00 00 00 14
19B1:00 00 00 00 00 00 00 00 E3
19B9:00 00 00 3C 00 00 EB 00 87
19C1:00 EB 00 00 3B 00 00 0D D5
19C9:00 00 0D 00 00 3B 00 00 8A
19D1:3A 00 00 3B 00 00 0F 00 F2
19D9:00 01 00 00 00 00 00 00 4C
19E1:00 00 00 00 00 00 00 00 14
19E9:00 00 00 00 00 0F F0 00 3A
19F1:3F FC 00 3F FF 00 35 FF 61
19F9:00 3D FF 00 15 DF 00 35 D8
1A01:5F 00 35 5C 00 CD 5C 00 41
1A09:CD 57 00 CD 55 C0 F5 F5 66
1A11:C0 5F 55 C0 D5 55 C0 75 2F
1A19:5F C0 DD 75 C0 37 D7 00 D2
1A21:35 5F 00 35 57 00 35 57 97
1A29:00 35 57 00 0F 00 00 00 0E
1A31:00 00 00 00 00 00 00 00 65
1A39:00 00 00 00 00 00 00 00 6D
1A41:30 00 00 DC 00 03 57 00 16
1A49:0D 55 C0 0D 55 FF 0D F7 FE
1A51:AB 0D CE AB 0F C3 AB 0D 1F
1A59:FE AB 8F 5E AB BD 5E AF 90
1A61:AD 7B F5 AD 7A D5 FF FA 0B
1A69:D5 FF FF FF 00 00 00 00 88
1A71:00 00 00 00 00 00 00 00 A5
1A79:00 00 00 00 00 00 00 00 AD
1A81:00 00 00 00 00 00 00 00 B5
1A89:00 00 00 00 00 C0 0F F3 D2
1A91:7C F7 5D 5F 55 55 5F 57 B9
1A99:55 5F 75 55 FF DD 55 7F F5
1AA1:55 5D FF 55 7F FF 55 F0 C4
1AA9:FC FF C0 00 00 00 00 00 74
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

64Key Relocated

Article on page 57.

Program 1: 64Key

```
PX 10 PRINT"{CLR} 64-KEY"
DG 20 FORI=52557TO52739
XM 30 READA:X=X+A:POKEI,A
BG 40 NEXTI
CD 50 IFX<>24016THENPRINT"THER
      E IS AN ERROR IN YOUR DA
      TA STATEMENTS":STOP
BH 60 PRINT"SYS52557 TO ACTIVA
      TE":END
ER 80 DATA 120 , 173 , 20 , 3
      {SPACE}, 72
RM 90 DATA 173 , 21 , 3 , 72 ,
      173
AP 100 DATA 116 , 205 , 208 ,
      {SPACE}2 , 169
GB 110 DATA 118 , 141 , 20 , 3
      , 173
CK 120 DATA 117 , 205 , 208 ,
      {SPACE}2 , 169
SK 130 DATA 205 , 141 , 21 , 3
      , 104
DD 140 DATA 141 , 117 , 205 ,
      {SPACE}104 , 141
ES 150 DATA 116 , 205 , 88 , 9
      6 , 0
DP 160 DATA 0 , 72 , 138 , 72
      {SPACE}, 152
MC 170 DATA 72 , 165 , 215 , 7
      2 , 165
XJ 180 DATA 212 , 240 , 4 , 10
      4 , 76
KM 190 DATA 221 , 205 , 104 ,
      {SPACE}201 , 193
CP 200 DATA 144 , 82 , 201 , 2
      19 , 176
DA 210 DATA 78 , 56 , 233 , 19
      3 , 170
EM 220 DATA 189 , 229 , 205 ,
      {SPACE}162 , 0
PB 230 DATA 134 , 198 , 170 ,
      {SPACE}166 , 158
EP 240 DATA 132 , 34 , 160 , 1
      60 , 132
BK 250 DATA 35 , 160 , 0 , 10
      {SPACE}, 240
EG 260 DATA 16 , 202 , 16 , 12
      , 230
QF 270 DATA 34 , 208 , 2 , 230
      , 35
HB 280 DATA 177 , 34 , 16 , 24
      6 , 48
JF 290 DATA 241 , 200 , 177 ,
      {SPACE}34 , 48
RE 300 DATA 17 , 8 , 142 , 255
      , 205
BX 310 DATA 230 , 198 , 166 ,
      {SPACE}198 , 157
FH 320 DATA 119 , 2 , 174 , 25
      5 , 205
CQ 330 DATA 40 , 208 , 234 , 2
      30 , 198
AE 340 DATA 166 , 198 , 41 , 1
      27 , 157
GH 350 DATA 119 , 2 , 169 , 20
      , 141
XG 360 DATA 119 , 2 , 230 , 19
      8 , 104
JB 370 DATA 168 , 104 , 170 ,
      {SPACE}104 , 76
FR 380 DATA 49 , 234 , 198 , 1
      69 , 199
HR 390 DATA 134 , 128 , 129 ,
      {SPACE}161 , 144
GF 400 DATA 133 , 137 , 141 ,
      {SPACE}200 , 202
PG 410 DATA 130 , 159 , 151 ,
      {SPACE}194 , 201
HG 420 DATA 196 , 163 , 183 ,
      {SPACE}197 , 131
DA 430 DATA 135 , 140 , 158 ,
      {SPACE}127 , 0
KJ 440 DATA 0 , 0 , 255 , 255
      {SPACE}, 255
```

Omicron

Article on page 20.

Program 1: Omicron—BASIC section

```
QS 2 POKE 792,193
AC 10 IFPEEK(13274)<>169THENLO
      AD"OMICRON ML",8,1
JP 20 SYS13274
BX 27 POKE 808,234
XC 30 POKE56,48:CLR:BL=51908:F
      =15:SCR=1024
QE 40 IFPEEK(788)=0THENFORT=0T
      O149:READD$:NEXT:GOSUB15
      10:GOTO60
CH 50 GOSUB1200
KH 60 GOSUB1000
ES 70 GOSUB780
KS 80 GOSUB580
RG 90 SYS51838:POKE56322,255:I
      F(PEEK(53280)ANDF)=0THEN
      220
PE 100 LV=LV-1:POKE251,48
SB 110 FORT=0TO64:POKE163,0:PO
      KE164,3
DA 120 POKE53280,T:POKE53232,(
      3ANDT)+220
SK 130 NEXT
EG 140 POKE164,0:CS=CS+1
EF 150 IFLV=0THEN170
KA 160 GOTO80
JQ 170 POKE53178,0:POKE53179,0
JS 180 POKE53272,21:PRINT"
      {CLR}{9 DOWN}"SPC(11)"
      {CYN}G A M E{3 SPACES}O
```

```
        V E R"
BD 190 FORT=0TO2000:NEXT
DS 200 FORT=0TO7:SL=SL+PEEK(83
       2+T)*10↑(7-T):NEXT
XX 210 LD=DF:GOTO60
MJ 220 REM :::::::::: GAME WON
       ::::::::::
KS 230 IFL<15THEN430
KD 240 POKE53178,0:POKE251,48:
       POKE53281,1:FORT=2TO7:W
       (T)=PEEK(1442+T)-176:NE
       XT
DH 250 PRINT"{CLR}{WHT},,,,,,,
       ,,,,,,,,,,,,,,,,,,,,,,,
       ,,,,,,,,,,":FORT=0TO9:P
       RINT:NEXT
MC 260 PRINTSPC(11)"{RVS}{BLK}
       CONGRATULATIONS!"
BC 270 PRINT"{RVS}"SPC(11)"CON
       GRATULATIONS{K}":FORT=0
       TO8:PRINT:NEXT
DJ 280 PRINT"{WHT},,,,,,,,,,,,
       ,,,,,,,,,,,,,,,,,,,,,,,
       ,,,,,";
DM 290 FORT=0TO75
JD 300 X=INT(RND(1)*34+4):Y=IN
       T(RND(1)*19+3):PL=X+(Y*
       40)+SCR:IFPEEK(PL)<>32T
       HEN340
HA 310 POKEPL+54272,1:POKEPL+5
       4273,0
MF 320 POKEPL,39+RND(0)*2:POKE
       PL+1,8
XE 330 SYS49680:POKE53280,T
FG 340 NEXT
XG 350 PRINT"{CLR}":POKE53272,
       21
PR 360 POKE251,80:PRINT"
       {5 DOWN}{BLU}{7 SPACES}
       YOU HAVE SURVIVED OMICR
       ON!"
MB 370 PRINT"{DOWN}{RED}
       {7 SPACES}BONUS"(DF+1)*
       5000"POINTS AWARDED FOR
       "
DG 380 PRINT"{DOWN}{12 SPACES}
       EACH REMAINING SHIP."
HD 390 PRINT"{DOWN}{PUR} "LV"S
       HIPS *"(DF+1)*5000"="(D
       F+1)*5000*LV"EXTRA POIN
       TS."
DX 400 FORT=0TO4500:NEXT:POKE2
       51,48
QR 410 FORT=0TO7:POKE832+T,W(T
       ):NEXT
FF 420 POKE836,PEEK(836)+(DF+1
       )*5*LV:GOTO170
GC 430 REM :::::::: LEVEL COMP
       LETED ::::::
AX 440 POKE53178,1:POKE53179,0
EH 450 IFCS<>0THEN510
CB 460 POKE 808,234:PRINT"
       {HOME}{3 DOWN}{YEL}
       {RVS}{2 RIGHT}
       {36 SPACES}"
QJ 470 PRINT"{2 RIGHT}{RVS}
       {2 SPACES}BONUS 1000 FO
       R EXCELLENT PLAY !!
       {2 SPACES}"
DS 480 PRINT"{UP}{2 RIGHT}
       {RVS}{2 SPACES}BONUS
       {E}{3 A} FOR EXCELLENT
       {SPACE}PLAY {2 K}
       {2 SPACES}"
XD 490 PRINT"{2 RIGHT}{RVS}
       {36 SPACES}"
XG 500 POKE251,64:FORT=0TO2000
       :NEXT:POKE251,0:POKE836
       ,PEEK(836)+1
EB 510 CS=0:L=L+1:IF(3AND(L))<
       >0THEN570
FQ 520 PRINT"{HOME}{17 DOWN}
       {RVS}{GRN}{10 RIGHT}
       {20 SPACES}"
RH 530 PRINT"{RVS}{10 RIGHT}
       {3 SPACES}BONUS SHIP !!
       !{3 SPACES}"
QS 540 PRINT"{RVS}{UP}
       {10 RIGHT}{3 SPACES}BON
       US SHIP {3 K}{3 SPACES}
       "
XP 550 PRINT"{RVS}{10 RIGHT}
       {20 SPACES}"
KA 560 POKE251,80:FORT=1TO2000
       :NEXT:POKE251,0:LV=LV+1
       :IFLV>9THENLV=9
JM 570 GOTO80
FJ 580 REM :::::: BETWEEN LEVE
       LS :::::::::
KF 590 IFCS>0THENPOKE53232,209
       :POKE53184,18:POKE53200
       ,60
AG 600 POKE53178,1:POKE53179,0
       :POKE251,48:POKE53216,1
QA 610 DR=INT(RND(0)*2)
BD 620 FORT=0TO7:POKE840+T,0:N
       EXT
RC 630 POKE844,L:POKE841,Q(DF,
       0,LANDF):POKE842,Q(DF,1
       ,LANDF):POKE165,Q(DF,2,
       LANDF)
KQ 640 SYS13262:POKE49408+32,3
SK 650 FORT=1TO15:POKE49408+32
       +T,1:POKE53232+T,216:PO
       KE53216+T,14
RC 660 POKE53200+T,RND(0)*48+1
       70:POKE53184+T,RND(0)*9
       0+40:POKE49408+48+T,DR
KJ 670 AD=0:IFDR=1THENPOKE4940
       8+96+T,2:AD=2
CC 680 POKE49408+T+80,T:POKE49
       408+7*16+T,((RND(0)*256
       )AND252)+AD
RR 690 NEXT
RD 700 POKE1531,LV+176:POKE157
       1,LV+240
AF 710 POKE1522,(L+1)/10+176:P
       OKE1562,(L+1)/10+240:LC
       =L+1
DG 720 IFLC>9THENLC=LC-10:GOTO
       720
GD 730 POKE1523,LC+176:POKE156
       3,LC+240
GJ 740 POKE164,80:POKE163,1
FJ 750 POKE49408,0:POKE49408+1
       6,0:SYS51908:FORT=0TO4:
       SYS49680:NEXT
BA 760 POKE53178,255:POKE53179
       ,255
BG 770 RETURN
SM 780 REM ::::::: NEW GAME SE
       TUP :::::::
AK 790 POKE53280,0:LV=5
BH 800 L=0:SL=0:CS=0:PRINT"
       {CLR}":POKE53272,31:SYS
       51800
FH 810 FORT=0TO11:POKE828+T,0:
       NEXT
MB 820 FORT=53160TO53248:POKET
       ,0:NEXT
KA 830 X=16384+SCR:FORT=0TO255
       STEP4
HA 840 POKEX+T,RND(0)*25+134:P
       OKEX+T+1,RND(0)*48+58:P
       OKEX+T+2,RND(0)*25+16
FC 850 POKEX+T+3,RND(0)*48+170
       :NEXT
HR 860 PRINT"{CLR}{BLU},,,,,,,
       ,,,,,,,,,,,,,,,,,,,,,,,
       ,,,,,,,,,,";
FS 870 FORT=1TO8:PRINT","SPC(3
       8)",";:NEXT
HK 880 PRINT",{8 SPACES},,,,,,
       ,,,,,,,,,,,,,,,
       {8 SPACES},";
XK 890 FORT=0TO3:PRINT",
       {8 SPACES},{7}
       {20 SPACES}{BLU},
       {8 SPACES},";:NEXT
AK 900 PRINT",{8 SPACES},,,,,,
       ,,,,,,,,,,,,,,,
       {8 SPACES},";
JQ 910 FORT=1TO8:PRINT","SPC(3
       8)",";:NEXT
FA 920 PRINT"{BLU},,,,,,,,,,,,
       ,,,,,,,,,,,,,,,,,,,,,,,
       ,,,,,";
FB 930 PRINT"{BLU},,,,,,,,,,,,
       ,,,,,,,,,,,,,,,,,,,,,,,
       ,,,,";:POKE2023,44:POKE
       56295,6
JM 940 PRINT"{HOME}{10 DOWN}
       {RVS}"SPC(10)"{7}
       {4 SPACES}SCORE=000000
       {4 SPACES}"
JH 950 PRINT"{RVS}{UP}"SPC(10)
       "{4 SPACES}SCORE{X}
       {6 A}{4 SPACES}"
MR 960 PRINT"{RVS}"SPC(10)"
       {2 SPACES}LEVEL=00 SHIP
       S=0{2 SPACES}"
PH 970 PRINT"{RVS}{UP}"SPC(10)
       "{2 SPACES}LEVEL{X}
       {2 A} SHIPS{X}{A}
       {2 SPACES}"
MK 980 POKE53184,18:POKE53200,
       60:POKE53232,209
CE 990 RETURN
AS 1000 REM :::::::: TITLE SCR
        EEN :::::::::
MR 1010 POKE53178,0:POKE53179,
        0:POKE251,48
HR 1020 POKE53280,0:POKE53281,
        0:PRINT"{CLR}";
MP 1030 L$(0)="ROOKIE":L$(1)="
        ADVANCED":L$(2)="PRO":
        L$(3)="EXPERT"
XQ 1040 SP$="{7}{RVS}{7 RIGHT}
        {27 SPACES}"
SX 1050 PRINT"{3 DOWN}"SP$:PRI
        NT"{RVS}"SPC(13)" O M
        {SPACE}I C R O N ":PRI
        NTSP$
XC 1060 PRINT"{DOWN}{6}
        {3 SPACES}SELECT A SKI
        LL LEVEL AND GET READY
        ."
BE 1070 PRINT"{2 DOWN}{13}"SPC(
        10)"YOUR SELECTIONS AR
        E...":SP=13
SF 1080 PRINTSPC(SP)"{WHT}
        {DOWN}F1 ... EXPERT"
EC 1090 PRINTSPC(SP)"F3 ... PR
        O"
XA 1100 PRINTSPC(SP)"F5 ... AD
        VANCED"
CB 1110 PRINTSPC(SP)"F7 ... RO
        OKIE"
HK 1120 IFSL>HYTHENHY=SL:WD=LD
SA 1130 PRINTSPC(2)"{DOWN}
        {CYN}LAST GAME'S SCORE
        :"SL"ON "L$(LD)"."
QR 1140 PRINTSPC(2)"HIGHEST SC
        ORE YET:"HY"ON "L$(WD)
        "."
FP 1150 A(3)=1:A(6)=2:A(5)=3:A
        (4)=4
PP 1160 Z=PEEK(203):IFZ>60RZ<3
        THEN1160
XG 1170 X=A(Z):DF=X-1
BH 1180 IFX<1THEN1160
MM 1190 RETURN
SG 1200 REM :::::: DOUBLE CHAR
        ACTERS ::::::
XK 1210 POKE 53280,0:POKE53281
        ,0
```

```
CR 1215 PRINT"{CLR}{10 DOWN}
        {YEL}CREATING NEW CHAR
        ACTER SET...PLEASE WAI
        T"
PG 1220 POKE56334,PEEK(56334)A
        ND254:POKE1,PEEK(1)AND
        251
CR 1230 B=15360:A=53252:FORK=0
        TO63:FORT=0TO3
QE 1240 POKEB+K*8+T*2,255-PEEK
        (53248+K*8+T):POKEB+1+
        K*8+T*2,255-PEEK(53248
        +K*8+T)
QE 1250 POKE15872+K*8+T*2,255-
        PEEK(A+K*8+T):POKE1587
        3+K*8+T*2,255-PEEK(A+K
        *8+T)
JP 1260 NEXT:NEXT:POKE1,55:POK
        E56334,1
MQ 1270 REM :::::::::: 16 SPRIT
        ES ::::::::::
XC 1280 SA=51968+40
DB 1290 FORT=0TO149:READV$
DM 1300 L$=LEFT$(V$,1)
HX 1310 IFASC(L$)>64THENHN=ASC
        (L$)-55
PE 1320 IFASC(L$)<65THENHN=ASC
        (L$)-48
PQ 1330 R$=RIGHT$(V$,1)
RG 1340 IFASC(R$)>64THENLN=ASC
        (R$)-55
CM 1350 IFASC(R$)<65THENLN=ASC
        (R$)-48
KS 1360 B=HN*16+LN:POKESA+T,B:
        NEXT
AM 1370 POKE53265,27:POKE56333
        ,127:POKE788,0:POKE789
        ,203
DH 1380 POKESA+109,PEEK(648)+3
        :POKE53274,129
JK 1390 DATA A5,FD,29,01,AA,49
        ,01,A8,BD,B2,CF,8D,1C,
        D0
FR 1400 DATA BD,B4,CF,8D,1D,D0
        ,BD,B6,CF,8D,17,D0,BD,
        B8
FG 1410 DATA CF,8D,1B,D0,BD,BA
        ,CF,8D,15,D0,AD,1E,D0,
        99
XP 1420 DATA BC,CF,AD,1F,D0,99
        ,BE,CF,A9,01,8D,19,D0,
        A5
KD 1430 DATA FD,29,01,0A,0A,0A
        ,AA,A0,00,84,FE,A9,01,
        85
DA 1440 DATA FC,BD,C0,CF,0A,99
        ,00,D0,90,06,A5,FC,05,
        FE
HF 1450 DATA 85,FE,BD,D0,CF,99
        ,01,D0,8A,84,FF,29,07,
        A8
JD 1460 DATA BD,E0,CF,99,27,D0
        ,BD,F0,CF,99,F8,07,A4,
        FF
KA 1470 DATA 18,26,FC,E8,C8,C8
        ,C0,10,D0,CD,A5,FE,8D,
        10
XP 1480 DATA D0,AD,1F,D0,E6,FD
        ,A9,00,8D,12,D0,AD,0D,
        DC
CG 1490 DATA 29,01,F0,03,4C,31
        ,EA,4C,BC,FE
BG 1500 REM :::::::::: GAME SE
        TUP ::::::::::
KK 1510 DIMQ(3,2,15)
EH 1520 FORD=0TO3:FORK=0TO2:FO
        RT=0TO15
PJ 1530 READV:IFK=2THENV=15-V
PJ 1540 IFK=1THENV=V*32+31
ES 1550 Q(D,K,T)=V:NEXT:NEXT:N
        EXT
KD 1560 RETURN
PS 1570 DATA 1,1,1,1,1,1,1,1,1
        ,1,2,2,2,2,2,2
CX 1580 DATA 1,1,2,2,3,3,4,4,5
        ,6,3,4,4,3,3,3
XR 1590 DATA 0,0,0,0,0,0,0,0,0
        ,0,0,0,0,0,0,0
AH 1600 DATA 1,1,1,1,1,1,2,2,2
        ,2,2,2,2,2,2,2
GX 1610 DATA 1,2,3,4,5,5,2,3,4
        ,4,5,5,7,6,6,6
MF 1620 DATA 0,1,1,1,1,1,1,1,1
        ,1,1,1,1,1,1,1
JE 1630 DATA 1,1,1,1,1,2,2,2,2
        ,2,2,3,3,3,3,3
DE 1640 DATA 3,4,5,7,7,2,3,3,4
        ,5,5,4,6,3,4,4
AG 1650 DATA 1,1,1,1,2,2,2,2,1
        ,1,2,2,1,2,2,2
QH 1660 DATA 4,4,4,3,3,2,2,1,3
        ,3,3,4,7,4,4,4
GJ 1670 DATA 1,3,5,2,3,6,7,7,3
        ,3,5,3,2,4,5,7
FG 1680 DATA 1,1,1,2,2,3,3,4,2
        ,3,1,2,2,2,2,3
```

Program 2: Omicron—ML section

See instructions in article on page 20 before typing in.

```
33CE:A2 00 8A 9D 00 C1 E8 E0 6B
33D6:80 D0 F8 60 A9 68 85 AE 7F
33DE:A9 39 85 AF A0 00 A2 00 5E
33E6:A9 90 85 B0 A9 C1 85 B1 13
33EE:B1 AE 91 B0 C8 D0 F9 E8 7D
33F6:E6 AF E6 B1 E0 0B D0 F0 7A
33FE:60 00 00 00 00 C0 00 00 98
3406:70 00 00 3C 00 00 1F 00 A8
340E:00 0F C0 00 7F F0 00 0F 21
3416:C0 00 1F 00 00 3C 00 00 B3
341E:70 00 00 C0 00 00 00 00 CA
3426:00 00 00 00 00 00 00 00 8E
342E:00 00 00 00 00 00 00 00 96
3436:00 00 00 00 00 00 00 00 9E
343E:00 00 00 00 00 00 38 00 17
3446:03 F8 00 3F F0 00 FF F0 DA
344E:00 0F F0 00 07 F0 00 0F A3
3456:E0 00 0E E0 00 00 60 00 BF
345E:00 60 00 00 60 00 00 00 E1
3466:00 00 00 00 00 00 00 00 CE
346E:00 00 00 00 00 00 00 00 D6
3476:00 00 00 00 00 00 00 00 DE
347E:00 00 00 00 00 06 00 00 FE
3486:06 00 00 0F 00 00 0F 00 01
348E:00 1F 80 00 3F C0 00 7F 4B
3496:E0 00 7F E0 00 F6 F0 00 2B
349E:C6 30 00 86 10 00 00 00 5F
34A6:00 00 00 00 00 00 00 00 0F
34AE:00 00 00 00 00 00 00 00 17
34B6:00 00 00 00 00 00 00 00 1F
34BE:00 00 00 00 00 E0 00 00 AA
34C6:FE 00 00 7F E0 00 7F FC A9
34CE:00 7F 80 00 7F 80 00 3F 64
34D6:C0 00 38 C0 00 38 00 00 93
34DE:18 00 00 18 00 00 00 00 D4
34E6:00 00 00 00 00 00 00 00 4F
34EE:00 00 00 00 00 00 00 00 57
34F6:00 00 00 00 00 00 00 00 5F
34FE:00 00 00 00 00 00 70 00 48
3506:01 E0 00 07 C0 00 1F C0 9E
350E:00 7F 80 00 FF F0 00 7F AB
3516:80 00 1F C0 00 07 C0 00 4E
351E:01 E0 00 00 70 00 00 00 C4
3526:00 00 00 00 00 00 00 00 90
352E:00 00 00 00 00 00 00 00 98
3536:00 00 00 00 00 00 00 00 A0
353E:00 00 18 00 00 18 00 00 0C
3546:38 00 00 38 C0 00 3F C0 95
354E:00 7F 80 00 7F 80 00 7F 26
3556:FC 00 7F E0 00 FE 00 00 39
355E:E0 00 00 00 00 00 00 00 39
3566:00 00 00 00 00 00 00 00 D0
356E:00 00 00 00 00 00 00 00 D8
3576:00 00 00 00 00 00 00 00 E0
357E:00 00 86 10 00 CF 30 00 5A
3586:EF 70 00 7F E0 00 7F E0 E3
358E:00 3F C0 00 1F 80 00 0F EA
3596:00 00 0F 00 00 06 00 00 FA
359E:06 00 00 00 00 00 00 00 0C
35A6:00 00 00 00 00 00 00 00 11
35AE:00 00 00 00 00 00 00 00 19
35B6:00 00 00 00 00 00 00 00 21
35BE:00 00 00 60 00 00 60 00 EF
35C6:00 60 00 0E E0 00 0F F0 40
35CE:00 07 F0 00 0F F0 00 FF 55
35D6:F0 00 3F F8 00 03 F8 00 2F
35DE:00 78 00 00 18 00 00 00 28
35E6:00 00 00 00 00 00 00 00 51
35EE:00 00 00 00 00 00 00 00 59
35F6:00 00 00 00 00 00 00 00 61
35FE:00 00 03 E0 00 07 F8 00 E5
3606:1F 1C 00 1F 00 00 0F C0 D9
360E:00 07 E0 00 07 E0 00 03 17
3616:F0 00 00 F8 00 30 F8 00 3D
361E:3F F0 00 0F E0 00 00 00 5E
3626:00 00 00 00 00 00 00 00 92
362E:00 00 00 00 00 00 00 00 9A
3636:00 00 00 00 00 00 00 00 A2
363E:00 00 0E 00 00 38 00 00 4D
3646:70 00 00 71 C0 00 7F F8 FF
364E:00 3F FC 00 1F FE 00 07 26
3656:E7 00 01 C3 00 00 07 00 21
365E:00 0E 00 00 7C 00 00 00 32
3666:00 00 00 00 00 00 00 00 D2
366E:00 00 00 00 00 00 00 00 DA
3676:00 00 00 00 00 00 00 00 E2
367E:00 00 00 00 00 00 00 00 EA
3686:C0 00 00 C0 78 00 C3 FC A7
368E:00 E7 FE 00 FF E7 00 7F F3
3696:E3 00 7F C3 00 1F 03 00 A3
369E:00 00 00 00 00 00 00 00 0B
36A6:00 00 00 00 00 00 00 00 13
36AE:00 00 00 00 00 00 00 00 1B
36B6:00 00 00 00 00 00 00 00 23
36BE:00 00 00 78 00 00 FC 00 AC
36C6:01 C6 00 03 E3 00 07 F3 B6
36CE:00 07 F0 00 07 F0 00 C3 DA
36D6:E0 00 61 C0 00 3F 80 00 E9
36DE:1F 00 00 00 00 00 00 00 DA
36E6:00 00 00 00 00 00 00 00 53
36EE:00 00 00 00 00 00 00 00 5B
36F6:00 00 00 00 00 00 00 00 63
36FE:00 00 00 00 00 00 00 00 6B
3706:00 00 00 01 E0 00 03 F0 82
370E:00 03 F0 00 03 F0 00 01 38
3716:E0 00 00 00 00 00 00 00 F4
371E:00 00 00 00 00 00 00 00 8C
3726:00 00 00 00 00 00 00 00 94
372E:00 00 00 00 00 00 00 00 9C
3736:00 00 00 00 00 00 00 00 A4
373E:00 00 00 00 00 1C 00 00 1D
3746:1F E0 00 0F FE 00 0F FC 80
374E:00 0F F8 00 0F F8 00 1F 1B
3756:F0 00 3D E0 00 38 70 00 B4
375E:00 30 00 00 00 00 00 00 D8
3766:00 00 00 00 00 00 00 00 D4
376E:00 00 00 00 00 00 00 00 DC
3776:00 00 00 00 00 00 00 00 E4
377E:00 00 F0 0C 00 7C 3C 00 36
3786:1F F8 00 0F FE 00 1E 3C 24
378E:00 7E 38 00 0F 3C 00 1F 2C
3796:FF 00 3D E3 00 78 70 00 AD
379E:F0 70 00 00 18 00 00 00 62
37A6:00 00 00 00 00 00 00 00 15
37AE:00 00 00 00 00 00 00 00 1D
37B6:00 00 00 00 00 00 00 00 25
37BE:00 00 F3 CE 00 7F FC 00 90
37C6:38 3C 00 F0 0F 00 F0 0F D8
37CE:00 F8 0F 00 38 0F 00 38 93
37D6:1E 00 3C 3C 00 7F F0 00 7F
37DE:F1 E0 00 70 78 00 00 00 49
37E6:00 00 00 00 00 00 00 00 55
37EE:00 00 00 00 00 00 00 00 5D
37F6:00 00 00 00 00 00 00 00 65
37FE:00 00 00 00 00 FF FF 00 6D
3806:00 00 00 00 00 00 07 1E A2
380E:78 E0 03 07 0E 1C 38 70 85
3816:E0 C0 38 38 38 70 70 F0 07
```

```
381E:E0 E0 18 18 18 18 18 18 25
3826:18 18 1C 1C 1C 0E 0E 07 2A
382E:07 07 C0 E0 70 38 1C 0E B4
3836:07 03 00 00 00 00 F0 3C 09
383E:0F 03 00 00 00 FF FF 00 F6
3846:00 00 07 1E 78 E0 00 00 C0
384E:00 00 03 07 0E 1C 38 70 51
3856:E0 C0 03 03 03 03 0C 0C 40
385E:0C 0C 18 18 18 18 18 18 C5
3866:18 18 C0 C0 C0 C0 30 30 A6
386E:30 30 C0 E0 70 38 1C 0E D3
3876:07 03 F0 3C 0F 03 00 00 91
387E:00 00 00 00 00 FF FF 00 EE
3886:00 00 07 1E 78 E0 00 00 01
388E:00 00 03 07 0E 1C 38 70 91
3896:E0 C0 07 07 0F 0E 0E 1C E1
389E:1C 1C 18 18 18 18 18 18 12
38A6:18 18 E0 E0 E0 70 70 38 35
38AE:38 38 C0 E0 70 38 1C 0E 1A
38B6:07 03 F0 3C 0F 03 00 00 D1
38BE:00 00 00 00 00 FF FF 00 2F
38C6:00 00 00 00 00 00 07 1E 63
38CE:78 E0 03 07 0E 1C 38 70 46
38D6:E0 C0 30 30 30 30 C0 C0 75
38DE:C0 C0 18 18 18 18 18 18 CD
38E6:18 18 0C 0C 0C 0C 03 03 45
38EE:03 03 C0 E0 70 38 1C 0E 72
38F6:07 03 00 00 00 00 F0 3C C9
38FE:0F 03 00 00 00 00 00 00 B7
3906:00 00 0C 00 30 0E 60 00 74
390E:1C 00 42 81 19 28 31 18 D2
3916:43 18 77 59 83 7A A9 96 A4
391E:B4 66 ED FC AE BC CC 97 AB
3926:EA 3E 18 3C 66 DB DB 66 25
392E:3C 18 F0 0F F0 0F F0 0F 88
3936:F0 0F C3 66 3C FF 3C 66 84
393E:C3 00 3F 3F 30 3E 3E 30 95
3946:30 30 3C C3 03 3C 30 30 3A
394E:00 30 00 FC E3 E3 FC E3 29
3956:E3 FC E7 E7 E7 3C 3C FF 1E
395E:FF FF EF D7 BB 7D FE FF 1E
3966:FF FF 08 09 01 05 04 06 94
396E:02 0A 03 03 00 83 83 83 8D
3976:00 03 00 83 83 83 00 03 0F
397E:03 03 01 02 03 28 29 2A A8
3986:2B 2C 50 51 53 54 78 79 0F
398E:7A 7B 7C A1 A2 A3 03 0D 7D
3996:04 07 01 03 05 07 09 0B 7E
399E:0D 0F 53 2B 2A 29 51 79 8A
39A6:7A 7B 10 11 12 13 14 15 62
39AE:16 17 18 19 1A 1B 1C 1D 19
39B6:1E 1F 00 01 02 03 04 05 39
39BE:06 07 08 09 0A 0B 0C 0D 29
39C6:0E 0F 53 53 2B 2A 2A 2A 24
39CE:29 51 51 51 79 7A 7A 7A 8E
39D6:7B 53 53 2B 2B 2B 2A 29 7C
39DE:29 29 51 79 79 79 7A 7B 14
39E6:7B 7B A9 D7 85 AE A2 00 D4
39EE:A9 03 85 AF A0 52 B1 AE 03
39F6:C9 20 B0 22 9D 00 40 A5 A1
39FE:AE 9D 00 41 A5 AF 9D 00 6B
3A06:42 18 48 69 D4 85 AF B1 0F
3A0E:AE 29 0F 9D 00 43 E8 68 27
3A16:85 AF A9 20 91 AE B1 AE C9
3A1E:C9 21 90 09 C9 25 B0 05 AB
3A26:38 E9 01 91 AE A5 AE C9 9D
3A2E:98 D0 06 A5 AF C9 07 F0 E1
3A36:08 E6 AE D0 B9 E6 AF D0 E4
3A3E:B5 A9 FF 9D 00 40 30 12 45
3A46:C0 52 F0 03 20 BE C3 A6 28
3A4E:02 20 FA C2 A6 02 EA 4C B6
3A56:E4 C2 A2 00 BD 00 40 C9 7A
3A5E:FF F0 5E A8 B9 D0 C1 85 7F
3A66:B3 B9 F0 C1 A8 BD 00 41 DA
3A6E:85 AE BD 00 42 85 AF 86 17
3A76:02 B1 AE C9 21 B0 1F C9 9E
3A7E:20 F0 06 A9 24 91 AE 10 6F
3A86:35 A5 B3 91 AE A5 AF 18 12
3A8E:69 D4 85 AF A6 02 20 08 1E
3A96:C4 91 AE 10 21 EA 20 E8 86
3A9E:C2 C9 25 90 19 C9 2B B0 8B
3AA6:15 C9 26 D0 9B A5 B3 49 0E
3AAE:18 91 AE 20 81 C5 69 D4 36
3AB6:85 AF A9 0F 91 AE E8 D0 E9
3ABE:9B 60 48 BD 00 43 C9 02 A0
3AC6:B0 02 68 60 68 A9 81 60 74
3ACE:EA 60 00 00 A5 AE 85 B0 74
3AD6:A5 AF 18 69 D4 85 B1 A0 64
3ADE:52 EA EA EA EA EA EA EA 07
3AE6:EA EA EA EA B1 AE C9 25 98
3AEE:D0 0F EE 46 03 A9 FF 20 B0
3AF6:2C CA A9 24 91 AE 38 B0 14
3AFE:D0 C9 27 D0 2E A5 AF 20 C7
3B06:4A CA AD 46 03 18 69 05 BE
3B0E:8D 46 03 A9 20 91 AE A2 1F
3B16:00 BD C8 C1 A8 B1 AE C9 64
3B1E:20 D0 09 BD C0 C1 91 AE B4
3B26:A9 0A 91 B0 E8 E0 08 D0 DC
3B2E:E8 F0 A2 C9 28 D0 26 EE 06
3B36:45 03 A9 FF 20 41 CA A9 8A
3B3E:20 91 AE A2 00 BC A8 C1 2F
3B46:B1 AE C9 20 D0 08 A9 24 9A
3B4E:91 AE A9 08 91 B0 E8 E0 F0
3B56:14 D0 EA 60 EA C9 29 D0 10
3B5E:15 20 35 CA A5 8F 29 03 7B
3B66:AA BD BC C1 A0 52 91 B0 77
3B6E:8A 18 69 25 91 AE C9 2A B4
3B76:D0 1D A9 FF 8D FF FF A9 E7
3B7E:20 91 AE 91 B0 A5 A2 29 E2
3B86:03 AA EE FF FF CA E0 FF F3
3B8E:D0 F8 A9 FF 8D FF FF 60 AD
3B96:C0 52 B0 15 84 AB 38 A9 56
3B9E:52 E5 AB 85 AC 38 A5 AE C5
3BA6:E5 AC 85 AE B0 02 C6 AF A1
3BAE:60 38 98 E9 52 18 65 AE 81
3BB6:85 AE 90 02 E6 AF 60 EA 6F
3BBE:EA EA 20 20 20 25 25 25 70
3BC6:26 20 26 26 27 26 27 27 C6
3BCE:27 28 28 28 25 2C 29 20 B6
3BD6:04 0C 00 08 29 29 29 29 3C
3BDE:29 29 BD 00 43 D0 02 A9 F6
3BE6:01 60 EA EA EA EA EA EA C5
3BEE:EA 4C 08 C3 4C 78 C5 4C 47
3BF6:07 C5 A2 00 BD 20 C1 F0 99
3BFE:F3 C9 03 D0 F2 BD 00 C1 9F
3C06:30 0A 18 7D C0 CF 9D C0 35
3C0E:CF 18 90 0D 38 29 7F 85 42
3C16:02 BD C0 CF E5 02 9D C0 47
3C1E:CF BD 10 C1 30 0A 18 7D 63
3C26:D0 CF 9D D0 CF 18 90 0D C8
3C2E:29 7F 85 02 BD D0 CF 38 F4
3C36:E5 02 9D D0 CF BD C0 CF A9
3C3E:C9 0F B0 10 BD 00 C1 29 11
3C46:7F 9D 00 C1 A9 0F 9D C0 87
3C4E:CF 20 00 C6 BD C0 CF C9 7D
3C56:A4 90 10 A9 A3 9D C0 CF C6
3C5E:20 00 C6 BD 00 C1 09 80 35
3C66:9D 00 C1 BD D0 CF C9 37 52
3C6E:B0 10 A9 38 9D D0 CF BD 89
3C76:10 C1 29 7F 9D 10 C1 20 55
3C7E:0D C6 BD D0 CF C9 DD 90 E5
3C86:10 A9 DC 9D D0 CF BD 10 38
3C8E:C1 09 80 9D 10 C1 20 0D E8
3C96:C6 BD C0 CF C9 2C 90 41 58
3C9E:C9 87 B0 3D BD D0 CF C9 62
3CA6:A8 B0 36 C9 70 90 32 20 4D
3CAE:4B C6 BD C0 CF C9 30 B0 F8
3CB6:02 A9 2B C9 82 90 02 A9 A0
3CBE:88 9D C0 CF BD D0 CF C9 92
3CC6:74 B0 02 A9 6F C9 A4 90 FC
3CCE:02 A9 A8 9D D0 CF EA EA 28
3CD6:EA EA EA EA EA EA EA EA 4F
3CDE:EA BD 20 C1 C9 03 F0 6A 03
3CE6:BC 20 C1 F0 65 84 B3 BD 6F
3CEE:60 C1 D0 11 FE C0 CF BC 8A
3CF6:70 C1 B9 00 44 DD C0 CF 3A
3CFE:B0 03 20 28 C6 BD 60 C1 C6
3D06:C9 01 D0 11 DE D0 CF BC 67
3D0E:70 C1 B9 00 44 DD D0 CF 73
3D16:90 03 20 28 C6 BD 60 C1 CF
3D1E:C9 02 D0 11 DE C0 CF BC 7F
3D26:70 C1 B9 00 44 DD C0 CF 6B
3D2E:90 03 20 28 C6 BD 60 C1 E7
3D36:C9 03 D0 11 FE D0 CF BC 19
3D3E:70 C1 B9 00 44 DD D0 CF A3
3D46:B0 03 20 28 C6 A4 B3 88 19
3D4E:D0 9B E8 E0 10 F0 03 4C D9
3D56:22 C4 60 AD 46 03 18 69 D1
3D5E:02 8D 46 03 A9 E5 85 BD E3
3D66:A9 E4 85 BE 18 A5 AF 60 A2
3D6E:18 1C 08 0C 0E 10 12 14 A6
3D76:18 1C 18 14 12 10 0E 0C 41
3D7E:08 06 28 24 20 1C 18 14 7B
3D86:10 0C 28 24 20 1C 18 14 09
3D8E:10 0C 60 30 20 10 58 28 3D
3D96:10 0C 50 28 10 08 50 40 2A
3D9E:30 20 00 00 00 00 00 00 39
3DA6:00 00 00 00 00 00 00 00 21
3DAE:00 00 A0 10 80 20 60 30 B3
3DB6:50 40 40 50 30 60 20 70 2A
3DBE:10 80 30 30 30 18 18 18 94
3DC6:50 50 28 28 28 28 20 20 47
3DCE:00 00 EA EA EA EA EA EA 19
3DD6:EA EA E0 00 D0 09 AD 3C DF
3DDE:03 49 FF 8D 3C 03 60 E0 95
3DE6:00 D0 08 AD 3D 03 49 FF F9
3DEE:8D 3D 03 60 EA EA 60 3D E6
3DF6:03 EE 10 C1 EE 10 C1 60 68
3DFE:60 EA BD 30 C1 D0 0F FE 8D
3E06:60 C1 BD 60 C1 29 03 9D 37
3E0E:60 C1 FE 70 C1 60 DE 60 BF
3E16:C1 BD 60 C1 29 03 9D 60 FB
3E1E:C1 DE 70 C1 60 BD D0 CF C8
3E26:38 E9 70 85 B1 BD D0 CF 95
3E2E:38 E9 A8 49 FF C5 B1 B0 16
3E36:02 85 B1 BD C0 CF 38 E9 C6
3E3E:2A 85 B0 BD C0 CF 38 E9 C2
3E46:87 49 FF C5 B0 B0 02 85 07
3E4E:B0 A5 B0 C5 B1 B0 0C 20 87
3E56:00 C6 BD 00 C1 49 80 9D 0E
3E5E:00 C1 60 20 0D C6 BD 10 68
3E66:C1 49 80 9D 10 C1 60 60 A8
3E6E:60 60 A2 00 8E 02 DC AD 6B
3E76:00 DC 29 0F 49 0F DD 90 13
3E7E:C1 D0 10 AD 00 DC 29 10 C2
3E86:D0 09 8A 18 69 D0 BD F0 1B
3E8E:CF 90 05 E8 E0 08 D0 DF EE
3E96:AD 00 DC 29 10 D0 08 A5 91
3E9E:B4 F0 03 20 DC C6 EA AD 99
3EA6:00 DC 29 10 D0 03 CA 86 2F
3EAE:AC 20 4A C7 60 60 AD C0 EF
3EB6:CF 85 AE AD D0 CF 85 AF AD
3EBE:20 1C C7 B1 B0 C9 21 B0 06
3EC6:23 AD F0 CF 29 07 0A 91 66
3ECE:B0 18 A5 B1 69 D4 85 B1 D4
3ED6:A9 00 91 B0 A5 BE D0 09 38
3EDE:A9 10 85 BD A9 70 8D BE A9
3EE6:00 E6 AC 60 A9 02 8D 20 49
3EEE:D0 E6 AC 60 EA EA A9 00 7F
3EF6:85 B0 A9 04 85 B1 A5 AF C5
3EFE:38 E9 2A 4A 4A 4A AA E0 AD
3F06:00 F0 0F 18 A5 B0 69 28 0F
3F0E:85 B0 90 02 E6 B1 CA 38 79
3F16:B0 ED A5 AE 38 E9 0A 4A CF
3F1E:4A A8 60 EA AE 3C 03 E0 F3
3F26:FB D0 01 E8 E0 04 D0 01 3F
3F2E:CA 8E 3C 03 AE 3D 03 E0 BE
3F36:FB D0 01 E8 E0 04 D0 01 4F
3F3E:CA 8E 3D 03 AD 00 DC 29 ED
3F46:01 D0 03 CE 3D 03 AD 00 18
3F4E:DC 29 02 D0 03 EE 3D 03 24
3F56:AD 00 DC 29 04 D0 03 CE 12
3F5E:3C 03 AD 00 DC 29 08 D0 DD
3F66:03 EE 3C 03 A9 FF 8D 02 44
3F6E:DC AD 3C 03 10 02 49 7F 19
3F76:8D 00 C1 AD 3D 03 10 02 E6
3F7E:49 7F 8D 10 C1 60 A5 AD BC
3F86:D0 03 20 98 C6 A9 00 85 1E
3F8E:AD 60 EA EA EA EA EA EA CB
3F96:EA 00 AD 4B 03 CD 49 03 D9
3F9E:B0 5F A2 00 8E 48 03 E8 26
3FA6:BD 20 C1 C9 01 D0 39 AD 4C
3FAE:4A 03 18 7D 40 C1 9D 40 72
3FB6:C1 90 03 FE 50 C1 BD 50 DF
3FBE:C1 C9 04 90 05 A9 03 9D 8C
3FC6:E0 CF BD 50 C1 C9 20 90 6C
3FCE:17 A9 07 9D E0 CF A9 02 99
3FD6:9D 20 C1 A5 A2 29 01 9D 18
3FDE:30 C1 EE 4B 03 38 B0 19 EB
3FE6:BD 20 C1 F0 07 C9 03 F0 E9
3FEE:03 EE 48 03 AD 48 03 CD 46
3FF6:49 03 B0 05 E8 E0 10 D0 FC
3FFE:A7 20 EB C8 A2 01 BD 20 18
4006:C1 F0 3F BD F0 CF 18 69 C7
400E:01 29 DB 9D F0 CF 86 02 84
```

```
4016:BD E0 CF C9 03 D0 0E 20 DB
401E:97 E0 A6 02 A5 8E 29 1F 70
4026:D0 03 20 80 C8 BD E0 CF AA
402E:C9 07 D0 03 20 BA C8 BD DA
4036:E0 CF C9 08 D0 0C A5 A2 79
403E:29 07 D0 03 20 80 C8 20 14
4046:BA C8 E8 E0 10 D0 B7 A5 5A
404E:B4 F0 02 A2 00 86 B4 4C 9F
4056:EA C8 BD D0 C0 85 B0 BD 7E
405E:E0 C0 85 B1 BC F0 C0 AD 24
4066:4C 03 29 0F 0A A6 A2 30 44
406E:03 18 69 01 AA B1 B0 C9 FA
4076:20 D0 16 BD E8 C3 91 B0 04
407E:E9 01 29 03 AA A5 B1 18 F0
4086:69 D4 85 B1 BD BC C1 91 B2
408E:B0 A6 02 60 BD D0 C0 85 8F
4096:B0 BD E0 C0 85 B1 86 02 09
409E:20 97 E0 A6 02 BC F0 C0 41
40A6:B1 B0 C9 20 D0 14 A5 8F 19
40AE:29 0F 91 B0 18 A5 B1 69 E8
40B6:D4 85 B1 A9 07 91 B0 EA 9E
40BE:EA EA A6 02 60 A2 01 BD B1
40C6:C0 CF 85 AE BD D0 CF 85 8D
40CE:AF 86 02 20 1C C7 A6 02 5A
40D6:A5 B0 9D D0 C0 A5 B1 9D B4
40DE:E0 C0 98 9D F0 C0 BD F0 E3
40E6:CF C9 DC 90 0F FE F0 CF 8C
40EE:C9 DF 90 08 A9 0E 9D F0 90
40F6:CF 9D F0 CF BD 20 C1 C9 9D
40FE:00 F0 4D B1 B0 C9 20 B0 1E
4106:02 90 07 C8 B1 B0 C9 20 1F
410E:B0 3E A5 B1 69 D4 85 B1 A3
4116:B1 B0 29 0F C9 02 B0 30 9B
411E:BD 20 C1 0A 18 6D 45 03 64
4126:8D 45 03 A9 00 9D 20 C1 34
412E:A9 DC 9D F0 CF 20 20 CA 89
4136:BD E0 C0 85 B1 A9 20 91 46
413E:B0 BD E0 CF C9 07 D0 03 B0
4146:CE 4B 03 A9 0A 9D E0 CF 56
414E:E8 E0 10 F0 03 4C ED C8 7C
4156:AD 4E 03 C5 A5 D0 33 A2 79
415E:01 BD 20 C1 F0 27 EE 4E 41
4166:03 A9 02 AC 4C 03 C0 03 D2
416E:90 01 EA C0 08 90 01 0A 71
4176:9D 00 C1 9D 10 C1 A9 03 B7
417E:9D 20 C1 A9 08 9D E0 CF F2
4186:A9 FF 8D FF FF E8 E0 10 05
418E:D0 CF AD C0 CF 85 AE AD CE
4196:D0 CF 85 AF 20 1C C7 A5 C7
419E:B1 85 AF 18 69 D4 85 B1 2E
41A6:B1 B0 29 0F C9 01 F0 13 8B
41AE:A5 AF 85 B1 B1 B0 C9 20 BF
41B6:F0 09 C9 2A F0 05 A9 02 C0
41BE:8D 20 D0 A2 07 BD 40 03 07
41C6:C9 0A 90 0A E9 0A FE 3F 18
41CE:03 9D 40 03 B0 EF 69 B0 3B
41D6:9D A2 05 69 40 9D CA 05 1B
41DE:CA E0 01 D0 E0 A0 00 AD 63
41E6:4B 03 F0 02 A0 10 AD 4E FC
41EE:03 C5 A5 90 02 A0 20 84 79
41F6:FB 60 A9 3E 85 BD A9 44 63
41FE:85 BE EE 4E 03 60 A9 95 39
4206:85 BD A9 8D 85 BE 60 20 D2
420E:97 E0 A9 21 85 BD A9 39 8D
4216:85 BE 60 A9 01 85 A3 A9 C2
421E:1E 85 A4 60 C9 08 B0 09 86
4226:A9 FD 85 A3 A9 FE 85 A4 E2
422E:60 60 A9 0F 8D 18 D4 A9 41
4236:F0 8D 06 D4 8D 14 D4 8D 98
423E:0D D4 A9 11 8D 04 D4 A9 94
4246:21 8D 0B D4 A9 81 8D 12 ED
424E:D4 A9 00 85 FB 85 A4 60 9F
4256:20 20 C4 20 98 C6 20 10 BD
425E:C2 4C BF CB 20 C0 C7 AD 4D
4266:60 39 48 A2 00 BD 61 39 8F
426E:9D 60 39 E8 E0 07 D0 F5 4A
4276:68 8D 67 39 AD 8D 02 D0 8B
427E:FB A9 0E 38 E5 A5 18 69 10
4286:10 CD 4E 03 D0 01 60 AD 79
428E:20 D0 29 0F F0 01 60 A5 5F
4296:CB C9 3E D0 BB 00 A9 00 79
429E:85 AE 85 B0 A9 04 85 AF 65
42A6:A9 D8 85 B1 A0 00 B1 B0 1B
42AE:29 0F C9 0E F0 0E C9 06 FE
42B6:F0 0A B1 AE C9 20 F0 04 0C
42BE:A9 24 91 AE E6 AE E6 B0 AE
42C6:D0 E4 E6 AF E6 B1 A5 AF BD
42CE:C9 08 D0 DA 60 A3 60 EA 3F
42D6:EA EA E6 FA A5 FA 29 0F 92
42DE:18 65 FB AA BD 98 C5 8D 5C
42E6:01 D4 A5 BE F0 05 38 E5 B3
42EE:BD 85 BE 8D 08 D4 A5 A4 E7
42F6:F0 03 38 E5 A3 85 A4 8D 24
42FE:0F D4 A5 FD 29 01 AA 49 C0
4306:01 A8 BD B2 CF 8D 1C D0 D7
430E:BD B4 CF 8D 1D D0 BD B6 D1
4316:CF 8D 17 D0 BD B8 CF 8D D5
431E:1B D0 BD BA CF 8D 15 D0 79
4326:AD 1E D0 99 BC CF AD 1F 5E
432E:D0 99 BE CF A9 01 8D 19 DD
4336:D0 A5 FD 29 01 0A 0A 0A 2F
433E:AA A0 00 84 FE A9 01 85 B0
4346:FC BD C0 CF 0A 99 00 D0 57
434E:90 06 A5 FC 05 FE 85 FE 51
4356:BD D0 CF 99 01 D0 8A 84 68
435E:FF 29 07 A8 BD E0 CF 99 45
4366:27 D0 BD F0 CF 99 F8 07 59
436E:A4 FF 18 26 FC E8 C8 C8 92
4376:C0 10 D0 CD A5 FE 8D 10 AC
437E:D0 AD 1F D0 E6 FD A9 00 4C
4386:8D 12 D0 AD 0D DC 29 01 7C
438E:F0 03 4C 31 EA 4C BC FE EB
4396:EA A2 00 86 AE E8 BD 20 58
439E:C1 F0 02 E6 AE E8 E0 10 DB
43A6:D0 F4 A5 AE D0 01 60 4C 0A
43AE:8A CA 00 00 00 00 00 00 2D
43B6:00 00 00 00 00 00 00 00 3D
```

Chain Reaction

Article on page 14.

```
CK 10 Y=30:DIM U(Y),H(Y),L(Y),
      J(Y),T1(Y),CA(Y),CD(Y),R
      T(Y),TR(Y)
MC 20 POKE 53269,0:PRINT"{CLR}
      ":POKE 53280,11:POKE 532
      81,11:Y=RND(-TI)
HR 30 ML$="{I}"+CHR$(8)+"{X}<"
      +CHR$(3)+"{2}XJ"+CHR$(16
      )+CHR$(248)+"L{B}{T}":PO
      KE 835,0
XJ 40 POKE 53272,PEEK(53272)AN
      D 240 OR 12
PR 50 POKE 836,208:POKE 830,0:
      POKE 831,216:POKE828,0:P
      OKE 829,56:POKE 56334,0
QH 60 POKE 1,51:ML$=ML$:SYS(PE
      EK(51)+256*PEEK(52)):POK
      E 1,55:POKE56334,1
PP 70 FOR I=12296 TO 12487:REA
      D J:POKE I,J:NEXT
DH 80 YY=7:XX=12:GOSUB530:PRIN
      T"{CYN}CUAGI MDACOGKI"
HE 90 YY=11:XX=8:GOSUB530:PRIN
      T"IBJPDM KE LHAYDMN
      {YEL}Q{CYN} KM {YEL}R
EK 100 GOSUB700:IF KT$<>"1" AN
       D KT$<>"2" THEN100
FQ 110 NP=VAL(KT$):IF NP=2 THE
       N YY=15:GOTO160
CS 120 COMPUTER=1:YY=15:XX=9:G
       OSUB530:PRINT"{CYN}CKJL
       BODM EGMNO {YEL}Y{CYN}
       {SPACE}KM {YEL}I
SE 130 GOSUB700:IF KT$<>"Y" AN
       D KT$<>"N" THEN130
HE 140 TU=0:IF KT$="Y" THEN TU
       =1
XX 150 GOTO190
HF 160 YY=15:XX=7:GOSUB530:PRI
       NT"{CYN}IBJPDM KE SKYNO
       GCTN {YEL}Q{CYN} KM
       {YEL}R
AD 170 GOSUB700:IF KT$<>"1" AN
       D KT$<>"2" THEN170
HF 180 POKE 828,VAL(KT$)-1:F2=
       PEEK(828)
CQ 190 YY=19:XX=14:GOSUB530:PR
       INT"{CYN}LHDAND WAGO"
EK 200 M$="O{2 Y}P{DOWN}
       {4 LEFT}{H}{2 SPACES}
       {N}{DOWN}{4 LEFT}{H}
       {2 SPACES}{N}{DOWN}
       {4 LEFT}L{2 P}@":B$="
       {3 UP}":PL(0)=15:PL(1)=
       10
PR 210 P$(1)="V {DOWN}{2 LEFT}
       {2 SPACES}":P$(2)="V
       {DOWN}{2 LEFT} V":P$(3)
       ="VV{DOWN}{2 LEFT} V":P
       $(4)="VV{DOWN}{2 LEFT}V
       V"
HP 220 UP=79:LM=91:RM=251:DN=2
       07:CL(0)=15:CL(1)=10:MV
       =1:MA=1:MD=1
FJ 230 FOR J=0 TO 24:POKE 5427
       2+J,0:READ X:POKE 54272
       +J,X:NEXT
QM 240 FOR V=1 TO 30:READ J(V)
       :NEXT
PS 250 SV=53248:MX=79:MY=92:PO
       KE 2042,11:FOR I=0 TO62
       :READ Q:POKE 704+I,Q:NE
       XT
BR 260 FOR I=1 TO 30:READ CD(I
       ),CA(I):NEXT
BM 270 POKE 53280,6:POKE 53281
       ,6:PRINT CHR$(31)"{CLR}
       {BLU}";
QP 280 FOR I=0 TO 23:PRINT"
       {RVS}{39 SPACES}"
CM 290 POKE 1063+(40*I),160:PO
       KE 55335+(40*I),6:NEXT
EX 300 PRINT"{RVS}{39 SPACES}
       {HOME}{BLK}":POKE2023,1
       60:POKE56295,6
RE 310 POKE 53281,11:PRINT SPC
       (14)"{CYN}CUAGI MDACOGK
       I{BLK}{DOWN}"
AC 320 FOR I=1 TO 5:PRINT SPC(
       8)M$B$M$B$M$B$M$B$M$B$M
       $:NEXT
HB 330 PRINT"{DOWN}{RVS}{BLU}
       {32 SPACES}{HOME}":POKE
        SV+41,7:GOSUB690
KF 340 X=TU:GOTO370
EB 350 FG=0:FOR H=1 TO 30:IF U
       (H)>=J(H) THEN FG=1:GOS
       UB710
EC 360 NEXT:IF FG=1 THEN350
ED 370 X=-X+1:PN=PN+1
BR 380 IF COMPUTER AND NOT X T
       HEN POKE 53269,0:RD=0:G
       OTO970
RH 390 POKE SV+41,CL(X):GOTO54
       0
EC 400 H=MV
MA 410 IF L(H)<>X+1 AND L(H) T
       HEN380
PK 420 U(H)=U(H)+1:FS(X)=FS(X)
       +1:IF L(H)=0 THEN L(H)=
       X+1
XK 430 GOSUB510:PRINT P$(U(H))
AJ 440 IF U(H)>=J(H) THEN POKE
        53269,0:GOSUB710:GOTO3
       50
AB 450 GOTO370
MR 460 XX=15:YY=24:GOSUB530:PR
       INT"{CYN}FAJD{RVS}{BLU}
        {OFF}{CYN}KXDM{HOME}":
       FOR Z=1 TO 2000:NEXT
DJ 470 XX=8:YY=24:GOSUB530:PRI
       NT"LMDNN{RVS}{BLU}
       {OFF}{CYN}EGMDPBOOKI
       {RVS}{BLU} {OFF}{CYN}OK
       {RVS}{BLU} {OFF}{CYN}LH
       AY{HOME}"
HS 480 WAIT 56320+X*F2,16,16:R
       UN
```

```
QD 490 IF L(O)<>X+1 THEN FS(X)
       =FS(X)+U(O):FS(-X+1)=FS
       (-X+1)-U(O)
JP 500 L(O)=X+1:U(O)=U(O)+1:GO
       SUB520:PRINT P$(U(O)):R
       ETURN
RB 510 XX=5+(4*CA(H)):YY=4*CD(
       H):POKE 646,CL(X):GOSUB
       530:RETURN
MG 520 XX=5+(4*CA(O)):YY=4*CD(
       O):POKE 646,CL(X):GOSUB
       530:RETURN
KF 530 POKE 783,0:POKE 781,YY:
       POKE 782,XX:SYS 65520:R
       ETURN
PC 540 POKE 53269,4
FF 550 JY=15-(PEEK(56320+X*F2)
       AND15):JB=PEEK(56320+X*
       F2)AND16
QB 560 IF JB=0 THEN400
SC 570 IF JY=8 THEN JY=3
FF 580 IF JY<1 OR JY>4 THEN550
PC 590 ON JY GOTO600,620,640,6
       60
QC 600 IF MX-4<UP THEN550
XR 610 MX=MX-32:MV=MV-6:MD=MD-
       1:GOTO680
MF 620 IF MX+4>DN THEN550
RJ 630 MX=MX+32:MV=MV+6:MD=MD+
       1:GOTO680
GC 640 IF MY+4>RM THEN550
FQ 650 MY=MY+32:MV=MV+1:MA=MA+
       1:GOTO680
FR 660 IF MY-4<LM THEN550
AC 670 MY=MY-32:MV=MV-1:MA=MA-
       1
JG 680 GOSUB690:GOTO550
XA 690 POKE SV+4,MY:POKE SV+5,
       MX:RETURN
MH 700 KT$="":POKE 198,0:WAIT
       {SPACE}198,1:GET KT$:RE
       TURN
PF 710 POKE 54276,64:POKE 5429
       0,128:POKE 24276,65:POK
       E 54290,129:FOR V=1 TO
       {SPACE}4
CA 720 GOSUB510:POKE 646,PL(X)
       :PRINT"{RVS}{UP}{LEFT}"
       M$:GOSUB510:PRINT"{BLK}
       {UP}{LEFT}"M$
XQ 730 FOR TD=1 TO 125:NEXT:NE
       XT
KH 740 U(H)=U(H)-J(H)
MR 750 IF U(H)>0 THEN GOSUB510
       :PRINT P$(U(H)):GOTO770
GJ 760 L(H)=0
QX 770 IF CD(H)=1 THEN790
ED 780 O=H-6:GOSUB490
AG 790 IF CA(H)=1 THEN810
CR 800 O=H-1:GOSUB490
FJ 810 IF CA(H)=6 THEN830
RC 820 O=H+1:GOSUB490
MQ 830 IF CD(H)=5 THEN850
RG 840 O=H+6:GOSUB490
XK 850 IF FS(0)<1 OR FS(1)<1 T
       HEN460
MR 860 RETURN
RH 870 LP=0:IF CD(H)=1 THEN890
KQ 880 IF L(H-6)=2 THEN KZ=H-6
       :GOTO960
GX 890 IF CA(H)=1 THEN910
JC 900 IF L(H-1)=2 THEN KZ=H-1
       :GOTO960
MB 910 IF CA(H)=6 THEN930
CA 920 IF L(H+1)=2 THEN KZ=H+1
       :GOTO960
RP 930 IF CD(H)=5 THEN RETURN
SP 940 IF L(H+6)=2 THEN KZ=H+6
       :GOTO960
CD 950 RETURN
FQ 960 LP=1:RETURN
BF 970 FOR I=1 TO 30:RT(I)=0:T
       R(I)=0:NEXT
XE 980 XT=0:FOR I=1 TO 30:IF L
       (I)=2 THEN1000
PP 990 T1(XT+1)=I:XT=XT+1
AH 1000 NEXT:RD=0
HH 1010 XX=16:YY=24:GOSUB530:P
       RINT"{CYN}OUGITGIF
       {HOME}"
BR 1020 FOR I=30 TO 30-XT STEP
       -1:RT(I)=0:NEXT:FOR I=
       1 TO XT
XM 1030 H=T1(I):GOSUB870
JD 1040 IF FG=1 AND LP AND U(H
       )>0 THEN1220
GS 1050 IF U(H)+1=J(H) AND LP=
       1 AND U(KZ)+1=J(KZ) TH
       EN RT(I)=6:GOTO1170
PB 1060 IF U(H)+1=J(H) AND LP=
       1 THEN RT(I)=2:GOTO117
       0
RE 1070 IF U(H)+1=J(H) AND LP=
       0 THEN RT(I)=1:GOTO117
       0
CH 1080 IF J(H)=2 AND LP=0 AND
        U(H)=1 THEN RT(I)=1:G
       OTO1170
JS 1090 IF J(H)=2 AND LP=0 AND
        U(H)=0 THEN RT(I)=4:G
       OTO1170
JQ 1100 IF J(H)=2 AND LP=1 AND
        U(H)=1 THEN RT(I)=4:G
       OTO1170
FE 1110 IF U(KZ)+1=J(KZ) THEN
       {SPACE}RT(I)=1:GOTO117
       0
SJ 1120 IF U(H)+2>=J(H) AND LP
       =1 AND U(KZ)+1< J(KZ)
       {SPACE}THEN RT(I)=5:GO
       TO1170
AH 1130 IF U(H)+2>=J(H) AND LP
       =0 THEN RT(I)=3:GOTO11
       70
AP 1140 IF U(H)+2>=J(H) THEN R
       T(I)=2:GOTO1170
MJ 1150 IF LP=0 THEN RT(I)=2:G
       OTO1170
SE 1160 RT(I)=1
MQ 1170 NEXT I:TC=1:ZT=0:AB=6
KS 1180 FOR I=1 TO XT:IF RT(I)
       =AB THEN TR(TC)=I:ZT=Z
       T+1:TC=TC+1
QQ 1190 NEXT:IF ZT>0 THEN1210
CF 1200 AB=AB-1:GOTO1180
HP 1210 DH=INT(ZT*RND(1))+1:HD
       =TR(DH):H=T1(HD)
HJ 1220 XX=15:YY=24:GOSUB530:P
       RINT"{RVS}{BLU}
       {11 SPACES}{HOME}":FG=
       FG+1:GOTO410
HC 1230 DATA 126,102,102,126,1
       02,102,102,0,102,102,1
       02,102,102,102,126,0
DX 1240 DATA 126,98,96,96,96,9
       8,126,0,126,98,96,120,
       96,98,126,0,126,98
JS 1250 DATA 98,120,96,96,96,0
       ,126,102,96,110,102,10
       2,126,0,126,90,24,24
QP 1260 DATA 24,90,126,0,96,96
       ,96,96,98,98,126,0,102
       ,118,126,126,110,102
MX 1270 DATA 102,0,99,119,127,
       107,99,99,99,0,126,102
       ,102,102,102,102,126
QC 1280 DATA 0,126,102,102,126
       ,96,96,96,0,126,102,10
       2,126,120,108,102,0
GJ 1290 DATA 126,102,96,126,6,
       102,126,0,126,90,24,24
       ,24,24,24,0,124,102
SS 1300 DATA 102,124,102,102,1
       24,0,56,24,24,24,24,24
       ,60,0,60,54,6,12,48
BR 1310 DATA 48,62,0,30,12,12,
       12,12,108,124,0,230,10
       8,120,112,120,108
GS 1320 DATA 230,0,231,102,102
       ,126,102,102,231,0,0,8
       ,16,124,222,190,254
AH 1330 DATA 124,198,198,198,2
       14,254,238,198,0,231,1
       02,102,102,102,60,24
HQ 1340 DATA 0,0,4,0,12,64,10,
       0,0,0,0,12,64,12,0,0,4
       ,0,12,128,12,0,0,50
CF 1350 DATA 244,47,2,3,3,3,3,
       2,3,4,4,4,4,3,3,4,4,4,
       4,3,3,4,4,4,4,3,2,3
XK 1360 DATA 3,3,3,2,0,0,0,0,0
       ,0,120,0,30,192,0,3,19
       2,0,3,192,0,3,0,0,0
JA 1370 DATA 0,0,0,0,0,0,0,0,0
       ,0,0,0,0,0,0,0,0,0,0,0
       ,0,0,0,0,0,0,0,192,0
CB 1380 DATA 3,192,0,3,192,0,3
       ,120,0,30,0,0,0,1,1,1,
       2,1,3,1,4,1,5,1,6,2,1
AD 1390 DATA 2,2,2,3,2,4,2,5,2
       ,6,3,1,3,2,3,3,3,4,3,5
       ,3,6,4,1,4,2,4,3,4
DQ 1400 DATA 4,4,5,4,6,5,1,5,2
       ,5,3,5,4,5,5,5,6
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How To Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

DOS Window

See instructions in article on page 30 before typing in.

Program 1: DOS Window Object File

```
CA60:4C 94 CA 3D 00 00 00 00 6E
CA68:00 00 00 00 00 00 00 00 FD
CA70:00 00 00 00 00 00 00 00 06
CA78:00 00 00 00 00 00 00 00 0E
CA80:00 00 00 00 00 00 00 00 16
CA88:00 00 00 FF 00 03 01 0E 3A
CA90:FE F6 31 EA 78 AD 14 03 DD
CA98:8D 92 CA AD 15 03 8D 93 31
CAA0:CA A9 B2 8D 14 03 A9 CA FF
CAA8:8D 15 03 A9 00 8D 8B CA 5D
CAB0:58 60 AD 8D 02 C9 04 D0 29
CAB8:0B A5 C5 C9 12 D0 05 AD 1E
CAC0:8B CA F0 03 6C 92 CA A9 0A
CAC8:FF 8D 8B CA 20 E7 FF A9 2A
CAD0:00 85 D4 20 90 FF AD 86 CA
CAD8:02 8D 8F CA AD 20 D0 8D 8E
CAE0:90 CA AD 21 D0 8D 91 CA E3
CAE8:78 A2 00 B5 00 9D 00 A0 D5
CAF0:BD 00 04 9D 00 A1 BD 00 C1
CAF8:05 9D 00 A2 BD 00 06 9D 3A
CB00:00 A3 BD 00 07 9D 00 A4 8B
CB08:BD 00 D8 9D 00 A5 BD 00 85
CB10:D9 9D 00 A6 BD 00 DA 9D A7
CB18:00 A7 BD 00 DB 9D 00 A8 4F
CB20:E8 D0 C8 58 A9 0E 8D 21 C0
CB28:D0 A9 93 20 D2 FF A2 00 E2
CB30:8E 20 D0 8E 86 02 8E 87 FA
CB38:02 20 66 CD A9 92 20 D2 2D
CB40:FF A2 27 A9 40 9D C8 04 0E
CB48:CA 10 FA 20 93 CD A2 00 C3
CB50:8A 9D 00 D8 E8 D0 FA 20 C2
CB58:CB CE 20 E4 FF F0 F8 8D 1F
CB60:8D CA C9 44 D0 0F A2 09 FF
CB68:86 FD 20 84 CD A9 01 8D B3
```

```
CB70:29 D8 4C C7 CC C9 49 D0 C9
CB78:08 A9 01 8D 51 D8 4C 95 93
CB80:CE C9 56 D0 08 A9 01 8D 40
CB88:79 D8 4C 95 CE C9 0D D0 7E
CB90:6B 78 A9 36 85 01 A2 00 0A
CB98:BD 00 A1 9D 00 04 BD 00 A8
CBA0:A2 9D 00 05 BD 00 A3 9D 14
CBA8:00 06 BD 00 A4 9D 00 07 1C
CBB0:BD 00 A5 9D 00 D8 BD 00 94
CBB8:A6 9D 00 D9 BD 00 A7 9D 83
CBC0:00 DA BD 00 A8 9D 00 DB 5E
CBC8:E8 D0 CD A2 39 BD 00 A0 4E
CBD0:95 00 E8 D0 F8 A2 2A BD C1
CBD8:00 A0 95 00 CA D0 F8 58 2F
CBE0:AD 8F CA 8D 86 02 AD 90 8D
CBE8:CA 8D 20 D0 AD 91 CA 8D 31
CBF0:21 D0 A9 00 8D 8B CA 85 38
CBF8:C6 4C BC FE C9 53 D0 0B D6
CC00:A9 01 8D B0 D8 20 D2 CD 26
CC08:4C 92 CE C9 52 D0 08 A9 72
CC10:01 8D A1 D8 4C 89 CE C9 3F
CC18:43 D0 08 A9 01 8D 88 D8 4B
CC20:4C 89 CE C9 5E D0 08 A9 A8
CC28:01 8D 6B D8 4C 38 CC C9 47
CC30:4C D0 48 A9 01 8D 43 D8 65
CC38:20 D2 CD 20 0D CE A2 08 43
CC40:86 BA AD 8D CA C9 4C D0 41
CC48:03 A9 00 2C A9 01 85 B9 A6
CC50:AD 8E CA A2 63 A0 CA 20 3B
CC58:BD FF A6 2B A4 2C A9 00 81
CC60:20 D5 FF B0 13 AD 8D CA BF
CC68:C9 4C D0 0C 86 2D 84 2E F4
CC70:86 2F 84 30 86 31 84 32 E0
CC78:4C B5 CE C9 5F D0 25 A9 4E
CC80:01 8D 60 D8 20 D2 CD 20 9F
CC88:0D CE A2 08 86 BA AD 8E 3A
CC90:CA A2 63 A0 CA 20 BD FF 01
CC98:A6 2D A4 2E A9 2B 20 D8 5B
CCA0:FF 4C B5 CE C9 4E D0 0B 25
CCA8:A9 01 8D 38 D8 20 CB CD 39
CCB0:4C 92 CE C9 55 D0 0D 20 B3
CCB8:BE CE A9 55 20 A8 FF A9 3D
CCC0:3B 4C A5 CE 4C 57 CB 20 24
CCC8:6E CE A9 09 85 FD 20 77 EE
CCD0:CD A9 01 A2 FF A0 CE 20 46
CCD8:BD FF A9 60 85 B9 20 D5 B5
CCE0:F3 A5 BA 20 B4 FF A5 B9 E1
CCE8:20 96 FF A9 00 85 90 A0 AA
CCF0:03 84 B7 20 A5 FF 85 C3 22
CCF8:20 A5 FF 85 C4 A4 90 D0 0F
CD00:52 A4 B7 88 D0 EB A6 C3 B4
CD08:A5 C4 20 CD BD 20 A5 FF 42
CD10:A6 90 D0 3F C9 00 F0 06 67
CD18:20 D2 FF 4C 0D CD A9 0D 3D
CD20:20 D2 FF E6 FD A5 FD C9 3B
CD28:17 D0 20 E6 FD 20 77 CD 23
CD30:A2 BB 20 66 CD 20 E1 FF 29
CD38:F0 19 20 CB CE 20 E4 FF 14
CD40:C9 0D D0 F1 20 80 CD A9 85
CD48:0A 85 FD 20 77 CD A0 02 42
CD50:4C F1 CC A2 18 86 FD 20 49
CD58:77 CD 20 B8 CD 20 42 F6 1D
CD60:20 AB FF 4C 4B CB BD 00 C0
CD68:CF F0 06 20 D2 FF E8 D0 24
CD70:F5 A9 20 20 D2 FF 60 A6 75
CD78:FD A0 06 18 20 F0 FF 60 A2
CD80:A9 0A 85 FD 20 77 CD 20 9E
CD88:B8 CD E6 FD A5 FD C9 19 82
CD90:D0 F2 60 20 6E CE 20 C2 11
CD98:CD A9 08 85 BA 20 B4 FF 9E
CDA0:A9 6F 85 B9 20 96 FF 20 B4
CDA8:A5 FF C9 0D F0 06 20 D2 D3
CDB0:FF 4C A7 CD 20 AB FF 60 41
CDB8:A2 1B A9 20 20 D2 FF CA BA
CDC0:10 F8 A2 06 A0 0A 18 20 D4
CDC8:F0 FF 60 20 6E CE A2 F4 D3
CDD0:D0 0C 20 6E CE A2 D8 D0 46
CDD8:05 20 6E CE A2 D3 20 66 C4
CDE0:CD A9 63 85 FB A9 CA 85 34
CDE8:FC A9 00 8D 8E CA A0 28 4F
CDF0:91 FB 88 10 FB 60 20 6E 76
CDF8:CE A2 E4 20 66 CD A9 3D 3E
CE00:A0 00 8C 8E CA 91 FB E6 E3
CE08:FB EE 8E CA 60 A4 CC F0 FD
CE10:08 A4 D3 B1 D1 09 80 91 B5
CE18:D1 20 CB CE 20 E1 FF D0 66
CE20:05 68 68 4C 29 CB 20 E4 C9
CE28:FF F0 E2 C9 0D F0 2E C9 4D
CE30:14 F0 2B AE 8E CA E0 13 D8
CE38:F0 D3 C9 7F 10 CF C9 20 E7
CE40:30 CB C9 22 F0 C7 A0 00 2C
CE48:91 FB E6 FB EE 8E CA 20 B1
CE50:D2 FF A4 D3 B1 D1 29 7F CF
CE58:91 D1 4C 0D CE 60 AE 8E 71
CE60:CA F0 AA C6 FB CE 8E CA 64
CE68:20 D2 FF 4C 0D CE A2 06 7E
CE70:A0 0A 18 20 F0 FF A2 46 F8
CE78:A9 20 20 D2 FF CA 10 FA 6A
CE80:A2 07 A0 00 18 20 F0 FF 68
CE88:60 20 D9 CD 20 0D CE 20 69
CE90:F6 CD 20 0D CE 20 BE CE 35
CE98:AD 8D CA 20 A8 FF A9 30 94
CEA0:20 A8 FF A9 3A 20 A8 FF B6
CEA8:A0 00 B9 63 CA F0 06 20 4A
CEB0:A8 FF C8 D0 F5 20 AE FF 56
CEB8:20 E7 FF 4C 4B CB A5 BA B4
CEC0:20 B1 FF A9 6F 85 B9 20 9A
CEC8:93 FF 60 AD 0B DD A2 00 2C
CED0:48 29 10 A0 02 10 0C A9 FA
CED8:BA 9D BF 04 B9 09 DD E8 0A
CEE0:48 29 F0 4A 4A 4A 4A 18 D7
CEE8:69 B0 9D BF 04 E8 68 29 D4
CEF0:0F 18 69 B0 9D BF 04 E8 31
CEF8:88 10 DC AD 08 DD 60 24 F1
CF00:12 20 20 20 20 20 20 20 98
CF08:20 20 20 20 20 20 20 20 A7
CF10:20 20 20 20 20 20 20 20 AF
CF18:20 20 20 20 20 20 20 20 B7
CF20:20 20 20 20 20 20 20 20 BF
CF28:20 92 20 44 3A 44 49 52 8C
CF30:45 43 54 4F 52 59 20 20 1B
CF38:20 20 4E 3A 46 4F 52 4D BE
CF40:41 54 20 20 20 4C 3A 4C 8E
CF48:4F 41 44 2C 38 0D 20 49 AA
CF50:3A 49 4E 49 54 49 41 4C 54
CF58:49 5A 45 20 20 20 5F 3A 58
CF60:53 41 56 45 20 20 20 20 FA
CF68:20 5E 3A 4C 4F 41 44 2C EF
CF70:38 2C 31 0D 20 56 3A 56 53
CF78:41 4C 49 44 41 54 45 20 3F
CF80:20 20 20 20 43 3A 43 4F 17
CF88:50 59 20 20 20 20 20 55 C3
CF90:3A 52 45 53 45 54 0D 20 75
CF98:52 3A 52 45 4E 41 4D 45 E5
CFA0:20 20 20 20 20 20 20 53 73
CFA8:3A 53 43 52 41 54 43 48 F1
CFB0:0D 0D 12 53 54 41 54 55 37
CFB8:53 3A 00 12 50 52 45 53 5B
CFC0:53 20 52 45 54 55 52 4E 9B
CFC8:2C 20 4F 52 20 53 54 4F DB
CFD0:50 92 00 12 4E 45 57 20 B4
CFD8:12 46 49 4C 45 4E 41 4D 34
CFE0:45 3A 92 00 12 4F 4C 44 AE
CFE8:20 46 49 4C 45 4E 41 4D 4B
CFF0:45 3A 92 00 12 4E 41 4D AD
CFF8:45 2C 49 44 3A 92 00 00 CF
```

Program 2: DOS Window Boot

```
XA 10 PRINT"{CLR}":IFA=0THENA=
      1:PRINT"{2 DOWN}{8}
      {2 SPACES}LOADING...":LO
      AD"DOS-WINDOW.OBJ",8,1
EF 20 CLR:DIM T(4):A$=CHR$(157
      )+CHR$(157)+CHR$(157):N$
      ="DOS-WINDOW"
HC 30 PRINTCHR$(147)SPC(134)N$
      :PRINTSPC(43)"WANT TO SE
      T THE T.O.D. CLOCK ? Y";
JK 40 PRINT A$;:INPUT B$:IFB$<
      >"Y"THEN80
QE 50 PRINT:INPUT"{2 SPACES}HO
      URS{4 SPACES}{5 LEFT}";T
      (1):IFT(1)>12THENPRINT"
      {3 UP}":GOTO50
PR 60 INPUT"{DOWN}MINUTES
      {4 SPACES}{4 LEFT}";T(2)
      :IFT(2)>59THENPRINT"
      {3 UP}":GOTO60
GE 70 INPUT"{DOWN}SECONDS
      {4 SPACES}{4 LEFT}";T(3)
      :IFT(3)>59THENPRINT"
      {3 UP}":GOTO70
ER 80 FORI=1TO3:H=INT(T(I)/10)
      :L=T(I)-10*H:T(I)=16*H+L
      :NEXT
AB 90 C=56587:POKEC+4,PEEK(C+4
      )AND127
DG 100 FORI=0TO3:POKEC-I,T(I+1
      ):NEXT
GA 110 PRINT"{2 DOWN}1.TO USE
      {SPACE}"N$", PRESS CTRL
      -D.":PRINT
MQ 120 PRINT"2.PRESS RETURN AT
       THE "CHR$(18)"STATUS:"
      CHR$(146)" LINE, TO"
JA 130 PRINTSPC(2)"CLOSE WINDO
      W AND RETURN."
GH 140 PRINT"{DOWN}3.TO DEACTI
      VATE, PRESS RUN/STOP-RE
      STORE."
XE 150 PRINT"4.SYS 51808, TO R
      EACTIVATE.":SYS51808:NE
      W
```

Power BASIC: Text Framer

Article on page 27.

Program 1: Text Framer—128 Version

```
10 REM COPYRIGHT 1987 COMPUTE!
    PUBLICATIONS, INC. - ALL R
   IGHTS RESERVED
20 PRINT"{CLR}{3 RIGHT}COPYRIG
   HT 1987 COMPUTE! PUB., INC.
   "
30 PRINTTAB(9)"ALL RIGHTS RESE
   RVED":PRINT"{DOWN}...LOADIN
   G ML"
40 FORI=49152TO49448:READA:POK
   EI,A:X=X+A:NEXT
50 IFX<>37505THENPRINT"ERROR I
   N DATA STATEMENTS.":STOP
60 DATA 76,17,192,0,0,10,10,1,
   64,66
70 DATA 85,73,74,75,0,0,0,173,
   3,192
80 DATA 201,38,176,28,173,4,19
   2,201,23,176
90 DATA 21,173,5,192,240,16,17
   3,6,192,240
100 DATA 11,24,173,3,192,109,5
   ,192,201,39
110 DATA 144,3,76,72,178,24,17
   3,4,192,109
120 DATA 6,192,201,24,176,242,
   173,136,2,133
130 DATA 252,56,169,216,229,25
   2,141,15,192,174
140 DATA 3,192,134,251,174,4,1
   92,240,11,169
150 DATA 40,141,14,192,32,238,
   192,202,208,250
160 DATA 173,10,192,141,16,192
   ,32,12,193,174
170 DATA 5,192,169,1,141,14,19
   2,173,8,192
180 DATA 141,16,192,32,238,192
   ,32,12,193,202
190 DATA 208,247,173,11,192,14
   1,16,192,32,238
200 DATA 192,32,12,193,173,9,1
   92,141,16,192
```

```
210 DATA 174,6,192,169,40,141,
    14,192,32,238
220 DATA 192,32,12,193,202,208
    ,247,173,13,192
230 DATA 141,16,192,32,238,192
    ,32,12,193,173
240 DATA 8,192,141,16,192,169,
    1,141,14,192
250 DATA 174,5,192,32,253,192,
    32,12,193,202
260 DATA 208,247,173,12,192,14
    1,16,192,32,253
270 DATA 192,32,12,193,173,9,1
    92,141,16,192
280 DATA 169,40,141,14,192,174
    ,6,192,32,253
290 DATA 192,32,12,193,202,208
    ,247,96,24,165
300 DATA 251,109,14,192,133,25
    1,165,252,105,0
310 DATA 133,252,96,56,165,251
    ,237,14,192,133
320 DATA 251,165,252,233,0,133
    ,252,96,160,0
330 DATA 173,16,192,145,251,24
    ,165,252,109,15
340 DATA 192,133,252,173,7,192
    ,145,251,56,165
350 DATA 252,237,15,192,133,25
    2,96
```

Program 2: Text Framer—64 Version

```
10 REM COPYRIGHT 1987 COMPUTE!
    PUBLICATIONS, INC. - ALL R
    IGHTS RESERVED
20 PRINT"{CLR}{3 RIGHT}COPYRIG
    HT 1987 COMPUTE! PUB., INC.
    "
30 PRINTTAB(9)"ALL RIGHTS RESE
    RVED":PRINT"{DOWN}...LOADIN
    G ML"
40 FORI=3072TO3368:READA:POKEI
    ,A:X=X+A:NEXT
50 IFX<>26833THENPRINT"ERROR I
    N DATA STATEMENTS.":STOP
60 DATA 76,17,12,0,0,10,10,1,6
    4,66
70 DATA 85,73,74,75,0,0,0,173,
    3,12
80 DATA 201,38,176,28,173,4,12
    ,201,23,176
90 DATA 21,173,5,12,240,16,173
    ,6,12,240
100 DATA 11,24,173,3,12,109,5,
    12,201,39
110 DATA 144,3,76,133,134,24,1
    73,4,12,109
120 DATA 6,12,201,24,176,242,1
    73,59,10,133
130 DATA 252,56,169,216,229,25
    2,141,15,12,174
140 DATA 3,12,134,251,174,4,12
    ,240,11,169
150 DATA 40,141,14,12,32,238,1
    2,202,208,250
160 DATA 173,10,12,141,16,12,3
    2,12,13,174
170 DATA 5,12,169,1,141,14,12,
    173,8,12
180 DATA 141,16,12,32,238,12,3
    2,12,13,202
190 DATA 208,247,173,11,12,141
    ,16,12,32,238
200 DATA 12,32,12,13,173,9,12,
    141,16,12
210 DATA 174,6,12,169,40,141,1
    4,12,32,238
220 DATA 12,32,12,13,202,208,2
    47,173,13,12
230 DATA 141,16,12,32,238,12,3
    2,12,13,173
240 DATA 8,12,141,16,12,169,1,
    141,14,12
250 DATA 174,5,12,32,253,12,32
    ,12,13,202
260 DATA 208,247,173,12,12,141
    ,16,12,32,253
270 DATA 12,32,12,13,173,9,12,
    141,16,12
280 DATA 169,40,141,14,12,174,
    6,12,32,253
290 DATA 12,32,12,13,202,208,2
    47,96,24,165
300 DATA 251,109,14,12,133,251
    ,165,252,105,0
310 DATA 133,252,96,56,165,251
    ,237,14,12,133
320 DATA 251,165,252,233,0,133
    ,252,96,160,0
330 DATA 173,16,12,145,251,24,
    165,252,109,15
340 DATA 12,133,252,173,7,12,1
    45,251,56,165
350 DATA 252,237,15,12,133,252
    ,96
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How To Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Off-Screen Trace

See instructions in article on page 50 before typing in.

```
8800:78 A9 C0 20 6E 89 A9 80 3F
8808:20 6E 89 A9 80 20 96 89 CB
8810:A0 00 20 50 8A A9 00 8D 03
8818:C0 8B A9 0C 8D EE 8B A9 4B
8820:84 8D EF 8B A9 3F 8D F0 E3
8828:8B A9 95 8D F1 8B A9 14 1A
8830:8D F2 8B A9 01 8D F3 8B 82
8838:A9 1B 8D F4 8B A9 00 8D 76
8840:F5 8B AD 02 03 8D F3 8A C5
8848:AD 03 03 8D F4 8A AD 08 5F
8850:03 8D F5 8A AD 09 03 8D D2
8858:F6 8A AD 14 03 8D F7 8A 47
8860:AD 15 03 8D F8 8A A9 C4 D0
8868:8D 02 03 A9 88 8D 03 03 3F
8870:A9 D3 8D 08 03 A9 88 8D DA
8878:09 03 A9 53 8D 14 03 A9 A5
8880:89 8D 15 03 20 BA 8A A9 37
8888:93 20 D2 FF 20 BA 8A A9 70
8890:84 85 38 58 20 60 A6 4C ED
8898:86 E3 78 20 B1 8A AD F3 FD
88A0:8A 8D 02 03 AD F4 8A 8D AE
88A8:03 03 AD F5 8A 8D 08 03 AE
88B0:AD F6 8A 8D 09 03 AD F7 28
88B8:8A 8D 14 03 AD F8 8A 8D 19
88C0:15 03 58 60 A9 7F 2D F9 CD
88C8:8A 8D F9 8A A9 00 85 C6 89
88D0:6C F3 8A A5 7A 48 A5 7B 7C
88D8:48 C9 03 B0 22 20 73 00 64
88E0:C9 9A F0 11 C9 89 F0 0D 10
88E8:C9 8D F0 09 C9 8A D0 0D 18
88F0:A9 00 8D FA 8A A9 80 0D 41
88F8:F9 8A 8D F9 8A D0 4A 20 47
8900:B1 8A 20 1C 8A 20 E1 FF EC
8908:F0 16 A6 C5 E0 26 D0 0A 95
8910:A9 04 2C 8D 02 F0 03 20 51
8918:BA 8A A5 C5 C9 3C D0 E5 03
8920:20 B1 8A 20 73 00 C9 8D BF
8928:D0 09 EE FA 8A EE FA 8A 04
8930:4C 49 89 C9 8E D0 09 CE 22
8938:FA 8A CE FA 8A 4C 49 89 96
8940:C9 8A D0 05 A9 00 8D FA A8
8948:8A 68 85 7B 68 85 7A 6C DD
8950:F5 8A 60 08 78 A9 04 2C 2C
8958:8D 02 F0 0E A5 C5 C9 26 AF
8960:D0 08 2C F9 8A 30 03 20 3E
8968:BA 8A 28 6C F7 8A 08 78 B9
8970:85 05 A9 8B 85 06 A9 C7 D4
8978:85 03 A9 00 85 04 8D FC 98
8980:8A A9 2E 8D FB 8A 20 76 A2
8988:8A AD 86 02 91 05 C8 AD 1D
8990:88 02 91 05 28 60 08 78 36
8998:85 05 A9 8B 85 06 A0 30 53
89A0:AD 02 DD 91 05 C8 AD 00 86
89A8:DD 91 05 C8 AD 18 D0 91 3D
89B0:05 C8 AD 21 D0 91 05 C8 DF
89B8:AD 11 D0 91 05 C8 AD 15 D5
89C0:D0 91 05 28 60 08 78 85 5D
89C8:03 A9 8B 85 04 A0 30 B1 46
89D0:03 8D 02 DD C8 B1 03 8D 87
89D8:00 DD C8 B1 03 8D 18 D0 E6
89E0:C8 B1 03 8D 21 D0 C8 B1 8D
89E8:03 8D 11 D0 C8 B1 03 8D B0
89F0:15 D0 28 60 08 78 85 03 FD
89F8:A9 8B 85 04 A9 C7 85 05 31
8A00:A9 00 85 06 8D FC 8A A9 1A
8A08:2E 8D FB 8A 20 76 8A B1 61
8A10:03 8D 86 02 C8 B1 03 8D 9B
8A18:88 02 28 60 A9 80 20 6E FA
8A20:89 A0 00 20 50 8A A9 C0 E4
8A28:20 F4 89 A9 20 AC FA 8A 8A
8A30:F0 06 20 D2 FF 88 D0 FA 2F
8A38:20 C9 BD A9 0D 20 D2 FF B0
8A40:A9 C0 20 6E 89 A9 80 20 59
8A48:F4 89 A0 01 20 50 8A 60 16
8A50:A9 D8 A2 8C C0 00 D0 06 3B
8A58:85 04 86 06 F0 04 85 06 0B
8A60:86 04 A9 00 85 03 85 05 37
8A68:A9 E8 8D FB 8A A9 03 8D 8C
8A70:FC 8A 20 76 8A 60 A0 00 29
8A78:AE FC 8A F0 0E B1 03 91 53
8A80:05 C8 D0 F9 E6 04 E6 06 1F
8A88:CA D0 F2 AE FB 8A F0 08 74
8A90:B1 03 91 05 C8 CA D0 F8 CD
8A98:60 A9 0C A0 00 99 00 D8 13
8AA0:99 00 D9 99 00 DA C0 E8 2D
8AA8:B0 03 99 00 DB 88 D0 ED 9A
8AB0:60 2C F9 8A 50 03 20 BA 72
8AB8:8A 60 2C F9 8A 50 0F A9 AD
8AC0:C0 20 6E 89 A0 01 20 50 3E
8AC8:8A A9 80 4C E2 8A A9 80 77
8AD0:20 96 89 A9 80 20 6E 89 52
8AD8:A0 00 20 50 8A 20 99 8A D9
8AE0:A9 C0 48 20 F4 89 68 20 C4
8AE8:C5 89 A9 40 4D F9 8A 8D 71
8AF0:F9 8A 60 00 00 00 00 00 B1
8AF8:00 00 00 00 00 00 00 00 0E
```

64+

See instructions in article on page 52 before typing in.

```
0801:1F 08 00 FA 8F 20 12 20 93
0809:36 34 2B 20 42 59 20 42 A2
0811:41 52 52 59 20 4C 2E 20 E4
0819:43 41 4D 50 20 00 37 08 41
0821:40 FA 9E C2 28 34 33 29 B1
0829:AA C2 28 34 34 29 AC 32 59
0831:35 36 AA 35 36 00 00 00 C3
0839:78 A2 FF 9A A5 01 09 03 1E
0841:85 01 A0 00 A9 BF 84 58 16
0849:85 59 84 5A 85 5B A2 20 A7
0851:20 EC A3 A9 FF 85 59 85 0A
0859:5B A2 20 20 EC A3 A2 00 01
0861:8E DA EC A2 06 8E 3C EC 07
0869:8E D9 EC E8 E8 8E DA E1 7C
0871:8E 2D E7 E8 E8 8E 3D EA 86
0879:E8 8E 36 EC 8E AB EC A2 D6
0881:10 8E 04 EC 8E 79 EC E8 A9
0889:E8 8E C5 FD A2 15 8E FB CC
0891:EB 8E 03 EC 8E 78 EC E8 83
0899:8E A0 EC 8E D1 EC E8 8E 42
08A1:A3 EC E8 8E A8 EC E8 E8 78
08A9:8E E5 EB 8E 26 EC E8 8E 26
```

```
08B1:F7 EB 8E 38 EC 8E AD EC F7
08B9:A2 20 8E FC E8 8E 07 EB 40
08C1:8E 68 FD 8E 6B FD 8E C4 11
08C9:FD E8 E8 8E E9 EC A2 2A 8B
08D1:8E 77 EB 8E EA EC A2 3F FC
08D9:8E 86 E7 A2 4C 8E 2C E7 D6
08E1:8E 85 E7 8E 44 EB A2 80 18
08E9:8E 07 EC E8 E8 8E 08 EC AD
08F1:E8 E8 8E 23 E5 8E 09 EC 1D
08F9:A2 8F 8E 06 EC A2 A1 8E 35
0901:C7 E3 A2 C1 8E FD E8 8E 2D
0909:69 FD A2 C6 8E 6C FD A2 D4
0911:C9 8E C9 E3 8E CC E5 8E 25
0919:09 EB 8E 46 EB 8E 6A FD 4D
0921:8E 6D FD E8 8E 2E E7 8E AF
0929:87 E7 E8 8E 28 E5 8E 78 6D
0931:EB 8E C6 FD A2 CF 8E E8 EF
0939:EC A2 D7 8E 08 EB A2 E0 64
0941:8E 45 EB A2 E3 8E 87 FD F9
0949:E8 8E FE E8 E8 8E D6 FD 0F
0951:A2 EA 8E CB E5 8E 2F E7 AD
0959:8E 30 E7 8E 88 E7 8E 89 2F
0961:E7 8E 8A E7 8E 0A EB A6 F5
0969:2C E8 E8 E8 18 A5 2B 69 8E
0971:C1 85 5A 90 01 E8 86 5B 2E
0979:A9 A1 85 58 A9 CB 85 59 DF
0981:A2 03 A0 5F 20 EC A3 18 C3
0989:A5 5A 69 5F 85 5A A5 5B 64
0991:69 03 85 5B A0 1D B1 5A B6
0999:99 B5 E4 88 C0 0B D0 F6 D5
09A1:B1 5A 99 9D E4 88 10 F8 92
09A9:A9 07 8D 6F A3 A8 88 B9 85
09B1:69 A3 99 68 A3 88 D0 F7 F3
09B9:A5 01 29 FD 85 01 4C F2 9F
09C1:FC A5 15 48 A5 14 48 20 11
09C9:F7 B7 A0 00 B1 14 85 63 26
09D1:C8 B1 14 85 62 68 85 14 63
09D9:68 85 15 A2 90 38 4C 49 95
09E1:BC A9 03 85 C2 60 E6 C2 9D
09E9:A5 C2 C9 A0 90 08 68 68 A0
09F1:A9 FD 48 A9 87 48 60 F0 0B
09F9:06 C9 11 F0 02 C9 15 60 74
0A01:8D 00 DC AC 8D 02 8C 8E 5E
0A09:02 60 20 16 E7 AD 13 03 BA
0A11:A4 D3 C0 4A 90 0C 29 01 31
0A19:D0 0D EE 13 03 A9 07 4C 01
0A21:16 E7 29 FE 8D 13 03 60 6E
0A29:D0 03 4C 91 E8 C9 15 D0 72
0A31:13 98 D0 02 E6 D6 C4 D5 21
0A39:F0 22 A5 D8 F0 02 C6 D8 86
0A41:C8 4C 50 E7 C9 1B F0 03 F4
0A49:4C 31 E7 20 12 CB A9 00 E1
0A51:85 C7 85 D4 85 D8 8D 91 54
0A59:02 8D 8A 02 4C A8 E6 C9 DF
0A61:03 D0 10 38 98 E9 0A B0 E1
0A69:FC 49 FF 65 D3 85 D3 A8 A9
0A71:4C 96 E7 C9 06 D0 09 C0 31
0A79:28 90 02 C6 D6 4C A6 E7 8F
0A81:C9 07 D0 31 78 20 1A CB AD
0A89:A9 1E 8D 01 D4 A9 F2 8D 7C
0A91:06 D4 A9 21 8D 04 D4 A2 ED
0A99:7F 20 B3 EE CA D0 FA 20 8A
0AA1:9F FF A9 20 8D 04 D4 A2 85
0AA9:37 20 B3 EE CA D0 FA 8E E4
0AB1:04 D4 4C A8 E6 C9 10 F0 80
0AB9:2C C9 0B D0 3F A6 D6 E8 EF
0AC1:B5 D9 30 01 E8 E0 19 10 4A
0AC9:0B B5 D9 09 80 95 D9 20 CA
0AD1:FF E9 30 F0 38 A5 D3 E9 5F
0AD9:28 10 02 69 28 85 D3 A8 84
0AE1:A6 D6 20 F0 E9 20 24 EA 15
0AE9:E6 D5 AE 86 02 A9 20 20 3C
0AF1:1E EA C8 C4 D5 90 F6 C6 DA
0AF9:D5 4C A8 E6 C9 16 90 0E 65
0B01:C9 19 B0 0A AA BD D5 CA BB
0B09:4C 39 CA 80 40 00 C9 1A A4
0B11:D0 17 AD 86 02 A2 FA 9D A1
0B19:FF D7 9D F9 D8 9D F3 D9 77
0B21:9D ED DA CA D0 F1 4C A8 19
0B29:E6 C9 12 D0 02 85 C7 4C 76
0B31:8B E7 AD 13 03 29 7F 8D 37
0B39:13 03 A0 17 A9 00 99 00 9F
0B41:D4 88 10 FA A9 0F 8D 18 52
0B49:D4 60 AD 8D 02 C9 04 D0 80
0B51:0E A4 CB C4 C5 F0 08 C0 20
0B59:03 90 04 C0 07 90 03 4C 6E
0B61:E0 EA B9 6A CB CD 89 02 2B
0B69:90 03 AD 89 02 AA 98 0A CC
0B71:A8 B9 6B CB 85 B0 B9 6C 43
0B79:CB 85 B1 8A A8 88 B1 B0 31
0B81:99 77 02 88 10 F8 86 C6 43
0B89:A2 FF 4C 26 EB 03 04 04 54
0B91:03 79 CB 7C CB 80 CB EC AD
0B99:EC 53 59 53 44 41 54 41 6C
0BA1:4C 49 53 54 AD 8D 02 F0 78
0BA9:21 C9 06 D0 05 CD 8E 02 0F
0BB1:D0 03 4C 4B EB AD 13 03 6E
0BB9:49 80 8D 13 03 30 03 A9 FF
0BC1:9C 2C A9 A3 AE 86 02 20 54
0BC9:1C EA 2C 13 03 30 05 A2 E4
0BD1:00 4C 6C EB A9 BF 85 F5 94
0BD9:A9 CB 85 F6 4C E0 EA 14 A7
0BE1:0D 1D 9A 05 9F 99 11 33 22
0BE9:35 37 34 30 38 36 01 35 C3
0BF1:0D 39 36 2C 9D 91 2E 37 2D
0BF9:93 1D 38 1B 13 34 11 39 9E
0C01:35 31 30 30 32 36 94 2B C7
0C09:10 33 2D 2E 3A 40 2C 23 CC
0C11:2A 3B 13 01 3D 5E 2F 31 72
0C19:14 04 32 20 02 34 03 FF 6B
0C21:2B 20 49 4E 53 54 41 4C 9F
0C29:4C 45 44 21 CE A5 02 78 DC
0C31:A9 FD 8D 00 DC AD 01 DC CB
0C39:C9 7F F0 F9 58 60 00 00 18
```

Sequential File Editor

Article on page 56.

```
RE 10 Z$=CHR$(20):A=3000:IFPEE
      K(56)<64THENA=100
RE 20 DIMA$(A):PRINT"{CLR}"CHR
      $(14)CHR$(8);:OPEN1,0,0:
      V=203:IFPEEK(771)=164THE
      NV=212
FE 30 LI$="@40 T@"
EF 40 PRINT"{8 SPACES}SEQUENTI
      AL FILE EDITOR":PRINTLI$
      "{3 DOWN}"
BA 50 PRINT"ENTER THE NAME OF
      {SPACE}THE FILE YOU WISH
       TO{2 SPACES}EDIT AND PR
      ESS RETURN{2 DOWN}"
CS 60 PRINT">";:INPUT#1,FI$:PR
      INT"{DOWN}":FI$=LEFT$(FI
      $,16)
SC 70 PRINT"FILENAME IS "CHR$(
      34)FI$CHR$(34):PRINT"
      {DOWN}IS THIS CORRECT? (
      Y/N) Y{LEFT}";
RJ 80 INPUT#1,A$:PRINT:IFLEN(A
      $)=0THENPRINT"{UP}":GOTO
      80
DE 90 A$=LEFT$(A$,1):IFA$="Y"T
      HENPRINT:GOTO110
QE 100 RUN
HR 110 PRINT"DISK DRIVE # (8-1
       0) 8{LEFT}";:INPUT#1,A$
      :DR=INT(VAL(A$))
HX 120 IFDR<8ORDR>10THENPRINT:
       PRINTSPC(22)"{UP}
      {7 SPACES}{UP}":GOTO110
KA 130 PRINT"{CLR}{2 SPACES}RE
      ADING "CHR$(34)FI$CHR$(
      34):PRINTLI$
GQ 140 F2$=FI$+",S,R":LN=1:PRI
      NT"{2 SPACES}1 : ";
XQ 150 OPEN8,DR,8,F2$:GET#8,A$
HM 160 IFST<>0THENPRINT"FILE N
      OT FOUND":GOSUB1150:GOS
      UB1140:RUN
EA 170 GOTO190
HM 180 GET#8,A$
MG 190 IFA$<>CHR$(13)THENA$(LN
      )=A$(LN)+A$
AE 200 IFST<>0THENGOSUB1150:GO
      TO260
RG 210 IFA$=CHR$(13)THEN230
AS 220 PRINTA$;:GOTO180
XS 230 B$="":IFLN<9THENB$=" "
GM 240 PRINT:LN=LN+1:PRINTB$;L
      N": ";:IFST<>0THEN260
AE 250 GOTO180
GC 260 CLOSE8:PRINT"{CLR}"SPC(
      13)"MAIN MENU":PRINTLI$
BX 270 PRINT"[1] LIST FILE":PR
      INT"{DOWN}[2] EDIT LINE
      ":PRINT"{DOWN}[3] INSER
      T LINE"
KM 280 PRINT"{DOWN}[4] DELETE
      {SPACE}LINE":PRINT"
      {DOWN}[5] RE-RUN PROGRA
      M"
RS 290 PRINT"{DOWN}[6] RE-SAVE
       FILE AFTER EDITING":PR
      INT"{DOWN}[7] PRINT FIL
      E TO PRINTER"
GF 300 PRINT"{DOWN}[8] EXIT PR
      OGRAM{2 DOWN}"
CM 310 PRINT"ENTER YOUR CHOICE
       AND PRESS RETURN{DOWN}
      "
CB 320 PRINT">";:INPUT#1,A$:A=
      INT(VAL(A$)):IFA<1ORA>8
      THEN260
PG 330 ONAGOTO340,520,600,700,
      800,830,890,1000
HJ 340 PRINT"{CLR}"SPC(13)"LIS
      T FILE":PRINTLI$
QS 350 PRINT"PRESS (A) TO ABOR
      T OR (SPACE) TO PAUSE"
CQ 360 PRINT"{DOWN}({RVS}W
      {OFF})ITH OR WITH({RVS}
      O{OFF})UT LINE NUMBERS?
       W{LEFT}";
XP 370 INPUT#1,A$:A$=LEFT$(A$,
      1):IFA$<>"W"ANDA$<>"O"T
      HEN370
GP 380 PRINT"{DOWN}":FORR=1TOL
      N:IFA$="O"THEN410
GR 390 B$="":IFR<10THENB$=" "
AR 400 PRINTB$;R": ";
XH 410 PRINTA$(R)
QM 420 FORT=1TO100:NEXT
AF 430 GETC$:IFC$=""THENNEXT:G
      OTO490
PJ 440 IFC$="A"THENPRINT:PRINT
      "{DOWN}{RVS}* ABORTED *
      ";GOTO490
XD 450 IFC$<>" "THEN430
BM 460 PRINT"{RVS}* PAUSED *"
EM 470 GETC$:IFC$=""THEN470
SD 480 PRINT"{UP}{11 SPACES}":
       PRINT"{UP}";:GOTO430
JR 490 PRINT"{DOWN}{2 SPACES}P
      RESS ANY KEY FOR MAIN M
      ENU"
AD 500 GETC$:IFC$=""THEN500
QE 510 GOTO260
PS 520 PRINT"{CLR}"SPC(13)"EDI
      T LINE":PRINTLI$:PRINT"
      {DOWN}WHICH LINE NUMBER
       TO EDIT? ";
MQ 530 INPUT#1,A$:A=INT(VAL(A$
      )):PRINT
GS 540 IFA>0ANDA<=LNTHEN560
AB 550 PRINT"{DOWN}BAD LINE NU
      MBER":GOSUB1150:GOTO520
CC 560 PRINT"{DOWN} "A$(A):PRI
      NT"{DOWN}RETYPE LINE OR
       PRESS RETURN{DOWN}"
EP 570 PRINT">";:GOSUB1040:PRI
      NT
SC 580 IFT$=CHR$(13)ANDP=0THEN
      PRINT"{DOWN}UN-CHANGED"
      :GOSUB1150:GOTO260
SF 590 PRINT"{DOWN}CHANGED":GO
```

```
      SUB1150:A$(A)=A$:GOTO26
      0
DR 600 PRINT"{CLR}"SPC(13)"INS
      ERT A LINE":PRINTLI$
CJ 610 PRINT"{2 DOWN}WHERE SHO
      ULD THE LINE BE INSERTE
      D?{DOWN}"
FF 620 PRINT">";:INPUT#1,A$:A=
      INT(VAL(A$)):PRINT
XE 630 IFA>0ANDA<=LNTHEN650
XX 640 PRINT:PRINT"BAD LINE NU
      MBER":GOSUB1150:GOTO600
AD 650 PRINT"{DOWN}WHAT WOULD
      {SPACE}YOU LIKE THAT LI
      NE TO BE?{4 SPACES}(PRE
      SS RETURN TO ABORT)
      {DOWN}"
MK 660 PRINT">";:GOSUB1040
MQ 670 IFT$=CHR$(13)ANDP=0THEN
      260
FC 680 FORR=LNTOASTEP-1:A$(R+1
      )=A$(R):NEXT:LN=LN+1
JX 690 PRINT:PRINT"{DOWN}DONE"
      :GOSUB1150:A$(A)=A$:GOT
      O260
GX 700 PRINT"{CLR}"SPC(13)"DEL
      ETE LINE":PRINTLI$
FK 710 PRINT"{DOWN}WHICH LINE
      {SPACE}WOULD YOU LIKE T
      O DELETE?{DOWN}"
FM 720 PRINT">";:INPUT#1,A$:A=
      INT(VAL(A$))
FH 730 IFA<1ORA>LNTHEN700
ED 740 PRINT:PRINT:PRINTA$(A):
      PRINT"{DOWN}DELETE THIS
       LINE (Y/N)? Y{LEFT}";
DR 750 INPUT#1,A$:A$=LEFT$(A$,
      1):PRINT
MK 760 IFA$="Y"THEN780
MB 770 PRINT"{DOWN}NOT DELETED
      ":GOSUB1150:GOTO260
BC 780 FORR=ATOLN:A$(R)=A$(R+1
      ):NEXT:LN=LN-1:PRINT"
      {DOWN}DELETED"
KK 790 GOSUB1150:GOTO260
FA 800 PRINT:PRINT"{DOWN}ARE Y
      OU SURE (Y/N)? Y{LEFT}"
      ;:INPUT#1,A$:A$=LEFT$(A
      $,1)
CM 810 IFA$="Y"THENRUN
SG 820 GOTO260
XC 830 PRINT"{CLR}"SPC(12)"RE-
      SAVE FILE":PRINTLI$
DS 840 PRINT"{DOWN}ENTER THE F
      ILE NAME":PRINT"('*' FO
      R SAME NAME AS BEFORE)
      {DOWN}"
DH 850 F3$=FI$:PRINT">";:INPUT
      #1,A$:IFA$<>"*"THENF3$=
      A$
AX 860 OPEN15,DR,15,"S0:"+F3$:
      CLOSE15:F4$="0:"+F3$+",
      S,W":OPEN8,DR,8,F4$
KR 870 PRINT"{2 DOWN}":FORR=1T
      OLN:PRINT"LINE: "R"{UP}
      ":PRINT#8,A$(R):NEXT
JA 880 CLOSE8:PRINT:PRINT:PRIN
      T"DONE":GOSUB1150:GOTO2
      60
HC 890 PRINT"{CLR}"SPC(13)"PRI
      NTING FILE":PRINTLI$
EK 900 PRINT"{DOWN}HOW MANY LI
      NES PER PAGE DOES YOUR
      {SPACE}PAPER"
XG 910 PRINT"HAVE? (0 = CONTIN
      UOUS PRINTING){DOWN}"
JH 920 PRINT">";:INPUT#1,LP:LL
      =0:IFLP<0ORLP<>INT(LP)T
      HEN890
KA 930 PRINT:PRINT:PRINT"PRESS
       (SPACE) TO PRINT OR 'A
      ' TO ABORT"
EA 940 GETA$:IFA$="A"THEN260
SG 950 IFA$=" "THEN970
JC 960 GOTO940
SA 970 OPEN9,4,7:FORR=1TO5:PRI
      NT#9:NEXT:FORR=1TOLN:PR
      INT#9,A$(R)
MB 980 LL=LL+1:IFLL=LP-10THENF
      ORT=1TO10:PRINT#9:NEXT:
      LL=0
PF 990 NEXT:PRINT#9:CLOSE9:GOT
      O260
EQ 1000 PRINT:PRINT"{DOWN}ARE
      {SPACE}YOU SURE (Y/N)?
       Y{LEFT}";
SJ 1010 INPUT#1,A$:A$=LEFT$(A$
      ,1):IFA$="Y"THENPRINT"
      {CLR}":END
BQ 1020 GOTO260
BX 1030 REM RETURN AS A$
JM 1040 P=0:A$=""
MA 1050 PRINTCHR$(166)CHR$(157
      );
KX 1060 GETT$:IFT$=""OR(T$=CHR
      $(24)ANDA$="")THEN1060
CE 1070 PRINT" "CHR$(157);
ER 1080 IFT$=Z$ANDLEN(A$)>0THE
      NA$=LEFT$(A$,LEN(A$)-1
      ):PRINTT$;:P=P-1:GOTO1
      050
HK 1090 IFT$=CHR$(24)THENFORZ=
      1TOLEN(A$):PRINTZ$;:NE
      XT:GOTO1040
JH 1100 IFT$=CHR$(13)THENPRINT
      T$:RETURN
BH 1110 IF(ASC(T$)AND127)<32TH
      EN1050
KQ 1120 IFP>253THEN1050
FX 1130 PRINTT$;:A$=A$+T$:POKE
      V,0:P=P+1:GOTO1050
JC 1140 OPEN15,8,15:INPUT#15,A
      $,B$,C$,D$:CLOSE15:RET
      URN
EM 1150 FORR=1TO1000:NEXT:RETU
      RN
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

One-Touch Function Keys

See instructions in article on page 55 before typing in.

```
C000:78 A9 0D 8D 14 03 A9 C0 63
C008:8D 15 03 58 60 A2 00 BD C6
C010:E9 C0 9D 00 04 E8 E0 28 18
C018:D0 F5 A5 C5 C9 40 D0 06 87
C020:8D 11 C1 4C 31 EA CD 11 8B
C028:C1 F0 F8 8D 11 C1 C9 04 E5
C030:F0 42 C9 05 F0 2C C9 06 16
C038:F0 16 C9 03 D0 E5 A2 00 84
C040:BD 12 C1 9D 77 02 E8 E0 AD
C048:04 D0 F5 86 C6 4C 31 EA DB
C050:A2 00 BD 16 C1 9D 77 02 B1
C058:E8 E0 04 D0 F5 86 C6 4C B7
C060:31 EA A2 00 BD 1A C1 9D 01
C068:77 02 E8 E0 05 D0 F5 86 2F
C070:C6 4C 31 EA 78 A9 31 8D 97
C078:14 03 A9 EA 8D 15 03 58 C7
C080:A9 93 20 D2 FF A9 08 AA 4E
C088:A0 00 20 BA FF A9 01 A2 55
C090:E8 A0 C0 20 BD FF 20 C0 B7
C098:FF A2 08 20 C6 FF 20 E4 21
C0A0:FF 20 E4 FF 20 E1 FF F0 40
C0A8:31 20 E4 FF 20 E4 FF A5 A1
C0B0:90 D0 27 20 E4 FF 8D 11 E8
C0B8:C1 20 E4 FF AE 11 C1 20 1D
C0C0:CD BD A9 20 20 D2 FF 20 3C
C0C8:E4 FF F0 06 20 D2 FF 4C D3
C0D0:C7 C0 A9 0D 20 D2 FF 4C 05
C0D8:A4 C0 20 CC FF A9 08 20 84
C0E0:C3 FF 20 00 C0 4C 31 EA CC
C0E8:24 20 86 B1 2D 2D 04 09 9F
C0F0:12 20 20 86 B3 2D 2D 0C A8
C0F8:09 13 14 20 20 86 B5 2D FB
C100:2D 13 01 16 05 20 20 86 CF
C108:B7 2D 2D 0C 0F 01 04 20 BD
C110:20 3F 4C 4F 41 44 53 41 F4
C118:56 45 4C 49 53 54 0D 00 3C
```

64 RAMdisk

See instructions in article on page 48 before typing in.

```
0801:0B 08 00 00 9E 32 30 36 EC
0809:31 00 00 00 4C FD 0F 4C 76
0811:1D C2 4C 30 C2 4C 5F C5 B8
0819:4C 10 C2 4C 5B C2 4C 47 36
0821:C3 4C 07 C4 4C 3E C3 4C 82
0829:55 C5 4C 1C C9 20 55 C0 DA
0831:A9 A9 8D 04 03 A9 C0 8D 40
0839:05 03 A9 69 8D 06 03 A9 8C
0841:C1 8D 07 03 A9 D0 8D 08 5A
0849:03 A9 C1 8D 09 03 A9 94 92
0851:A0 C0 4C 1E AB A9 00 8D DE
0859:FE FF 8D 78 CA A9 80 8D AD
0861:FF FF 8D 79 CA A9 DC 85 F6
0869:FB 8D 7A CA A9 FF 85 FC 2C
0871:8D 7B CA 60 0F B9 84 C0 AF
0879:99 EE FF 88 10 F7 A9 60 AE
0881:8D 73 C0 60 32 34 4B 20 6C
0889:52 41 4D 20 44 49 53 4B F7
0891:00 00 00 00 0D 52 41 4D 23
0899:20 44 49 53 4B 20 41 43 C9
08A1:54 49 56 41 54 45 44 0D 5A
08A9:00 A6 7A A0 04 84 0F BD CA
08B1:00 02 10 07 C9 FF F0 3E 23
08B9:E8 D0 F4 C9 20 F0 37 85 66
08C1:08 C9 22 F0 56 24 0F 70 6D
08C9:2D C9 3F D0 04 A9 99 D0 A2
08D1:25 C9 30 90 04 C9 3C 90 46
08D9:1D 84 71 A0 00 84 0B 88 82
08E1:86 7A CA C8 E8 BD 00 02 F9
08E9:38 F9 9E A0 F0 F5 C9 80 E5
08F1:D0 30 05 0B A4 71 E8 C8 4D
08F9:99 FB 01 B9 FB 01 F0 38 8F
0901:38 E9 3A F0 04 C9 49 D0 AA
0909:02 85 0F 38 E9 55 D0 9F C8
0911:85 08 BD 00 02 F0 DF C5 F8
0919:08 F0 DB C8 99 FB 01 E8 1B
0921:D0 F0 A6 7A E6 0B C8 B9 03
0929:9D A0 10 FA B9 9E A0 D0 3E
0931:B4 F0 0F BD 00 02 10 BC 7C
0939:99 FD 01 C6 7B A9 FF 85 2C
0941:7A 60 A0 FF CA C8 E8 BD C5
0949:00 02 38 F9 9E C1 F0 F5 56
0951:C9 80 D0 04 05 0B D0 9C 55
0959:A6 7A E6 0B C8 B9 9D C1 15
0961:10 FA B9 9E C1 D0 E0 F0 5F
0969:CA 30 03 4C F3 A6 C9 FF DF
0971:F0 F9 24 0F 30 F5 C9 CC A9
0979:B0 03 4C 24 A7 38 E9 CB 2E
0981:AA 84 49 A0 FF CA F0 08 52
0989:C8 B9 9E C1 10 FA 30 F5 21
0991:C8 B9 9E C1 30 05 20 47 83
0999:AB D0 F5 4C EF A6 4E 41 31
09A1:4D C5 52 45 4E 41 4D C5 42
09A9:53 43 52 41 54 43 C8 52 28
09B1:4E 45 D7 52 53 41 56 C5 6E
09B9:52 4C 4F 41 C4 44 49 D2 A2
09C1:52 52 55 CE 52 45 50 4C BD
09C9:41 43 C5 45 58 49 D4 00 EB
09D1:20 73 00 20 D9 C1 4C AE EF
09D9:A7 C9 CC 90 14 C9 D6 B0 FA
09E1:10 38 E9 CC 0A A8 B9 F8 73
09E9:C1 48 B9 F7 C1 48 4C 73 E0
09F1:00 20 79 00 4C ED A7 0E B2
09F9:C0 11 C0 14 C0 17 C0 1A 08
```

```
0A01:C0 1D C0 20 C0 23 C0 26 11
0A09:C0 29 C0 2C C0 2F C0 60 47
0A11:20 79 00 D0 03 4C 55 C0 55
0A19:20 1D C2 30 F8 20 81 C7 F2
0A21:A8 A9 00 99 EE FF 88 B1 C7
0A29:22 99 EE FF 88 10 F8 60 69
0A31:20 81 C7 20 2D C7 90 03 5D
0A39:4C 9E C7 20 FD AE 20 81 82
0A41:C7 48 A6 FB 86 6B A6 FC 0C
0A49:86 6C 20 2D C7 90 1D A5 F2
0A51:6B 85 FB A5 6C 85 FC 68 32
0A59:4C 90 C7 20 79 00 D0 03 23
0A61:4C A1 C7 20 81 C7 48 20 DA
0A69:2D C7 B0 03 4C AD C7 A5 9A
0A71:2D 38 E5 2B 85 FD 85 6D 36
0A79:A5 2E E5 2C 85 FE 85 6E 0D
0A81:AD 7A CA 38 ED 78 CA 85 54
0A89:6B AD 7B CA ED 79 CA 85 4B
0A91:6C AD 78 CA C9 00 AD 79 26
0A99:CA E9 C0 B0 0D A5 6B 38 BE
0AA1:E9 00 85 6B A5 6C E9 20 E4
0AA9:85 6C A5 6B 38 E9 12 85 1A
0AB1:6B A5 6C E9 00 85 6C 90 90
0AB9:0A A5 6B C5 FD A5 6C E5 4B
0AC1:FE B0 03 4C B4 C7 A5 2B E1
0AC9:8D DD C2 A5 2C 8D DE C2 E6
0AD1:AD 78 CA 8D E0 C2 AD 79 F3
0AD9:CA 8D E1 C2 AD FF FF 8D 1A
0AE1:FF FF EE DD C2 D0 03 EE FF
0AE9:DE C2 EE E0 C2 D0 03 EE 58
0AF1:E1 C2 AD E0 C2 C9 00 D0 79
0AF9:11 AD E1 C2 C9 C0 D0 0A 67
0B01:A9 00 8D E0 C2 A9 E0 8D B7
0B09:E1 C2 A5 6D D0 02 C6 6E D6
0B11:C6 6D D0 C8 A6 6E D0 C4 E1
0B19:A0 10 AD E0 C2 91 FB 8D 29
0B21:78 CA C8 AD E1 C2 91 FB 53
0B29:8D 79 CA 68 20 90 C7 AD C4
0B31:7A CA 38 E9 12 8D 7A CA 63
0B39:B0 03 CE 7B CA 60 20 51 63
0B41:C3 20 90 FF 4C 59 A6 24 8C
0B49:9D 10 F3 20 51 C3 4C 6C 51
0B51:CA 20 79 00 D0 03 4C A1 D0
0B59:C7 20 81 C7 20 2D C7 90 DD
0B61:03 4C 9E C7 A0 22 78 A5 80
0B69:01 48 A9 35 85 01 B1 FB 2A
0B71:85 6D C8 B1 FB 85 6E A0 4D
0B79:10 B1 FB 85 2D C8 B1 FB C7
0B81:85 2E 68 85 01 58 A5 2D 2D
0B89:38 E5 6D 85 FD A5 2E E5 04
0B91:6E 85 FE A5 2D C9 00 A5 B0
0B99:2E E9 C0 90 17 A5 6D C9 56
0BA1:00 A5 6E E9 E0 B0 0D A5 17
0BA9:FD 38 E9 00 85 FD A5 FE 78
0BB1:E9 20 85 FE A5 2B 85 FB 46
0BB9:18 65 FD 85 2D A5 2C 85 2B
0BC1:FC 65 FE 85 2E A0 00 78 54
0BC9:A5 01 48 A9 35 85 01 B1 0A
0BD1:6D 91 FB 68 85 01 58 E6 D0
0BD9:6D D0 02 E6 6E A5 6D C9 38
0BE1:00 D0 0E A5 6E C9 C0 D0 35
0BE9:08 A9 00 85 6D A9 E0 85 20
0BF1:6E E6 FB D0 02 E6 FC A5 D0
0BF9:FD 38 E9 01 85 FD B0 02 F1
0C01:C6 FE 05 FE D0 C1 60 A9 C4
0C09:0D 20 D2 FF A9 30 20 D2 2B
0C11:FF A9 20 20 D2 FF A9 12 95
0C19:20 D2 FF A9 22 20 D2 FF C7
0C21:78 A5 01 48 A9 35 85 01 B1
0C29:AD FE FF 85 6D AD FF FF 52
0C31:85 6E A9 EE 85 FB A9 FF 3B
0C39:85 FC A0 00 B1 FB 99 EC 05
0C41:C7 F0 03 C8 D0 F6 68 85 1F
0C49:01 58 A9 EC A0 C7 20 1E 7E
0C51:AB A9 22 20 D2 FF A9 0D E6
0C59:20 D2 FF 20 E1 FF F0 A6 CF
0C61:A5 FB 38 E9 12 85 FB B0 40
0C69:02 C6 FC A5 FB CD 7A CA 05
0C71:D0 0A A5 FC CD 7B CA D0 BB
0C79:03 4C 09 C5 A0 10 78 A5 7F
0C81:01 48 A9 35 85 01 B1 FB 44
0C89:85 69 C8 B1 FB 85 6A 68 26
0C91:85 01 58 A5 69 38 E5 6D 77
0C99:85 FD A5 6A E5 6E 85 FE 42
0CA1:A5 69 38 E9 00 A5 6A E9 E1
0CA9:C0 90 18 A5 6D 38 E9 00 C3
0CB1:A5 6E E9 E0 B0 0D A5 FD 86
0CB9:38 E9 00 85 FD A5 FE E9 2F
0CC1:20 85 FE A5 69 48 A5 6A A7
0CC9:48 A6 FD A5 FE 20 CD BD 9B
0CD1:68 85 6E 68 85 6D A9 05 0E
0CD9:85 D3 20 6C E5 A9 22 20 AE
0CE1:D2 FF 78 A5 01 48 A9 35 7E
0CE9:85 01 A0 00 B1 FB 99 EC B6
0CF1:C7 F0 03 C8 D0 F6 68 85 CF
0CF9:01 58 A9 EC A0 C7 20 1E 2F
0D01:AB A9 22 20 D2 FF 4C 56 27
0D09:C4 AD 7A CA 38 ED 78 CA 22
0D11:85 6D AD 7B CA ED 79 CA 82
0D19:85 6E AD 78 CA C9 00 AD F9
0D21:79 CA E9 C0 B0 0D A5 6D 66
0D29:38 E9 00 85 6D A5 6E E9 FA
0D31:20 85 6E A6 6D A5 6E 20 F3
0D39:CD BD A9 06 85 D3 20 6C 67
0D41:E5 A9 48 A0 C5 4C 1E AB 13
0D49:42 59 54 45 53 20 46 52 B3
0D51:45 45 2E 0D 00 20 81 C7 41
0D59:48 20 62 C5 4C 6F C2 20 0E
0D61:81 C7 20 2D C7 90 03 4C D7
0D69:9E C7 A0 22 78 A5 01 48 9F
0D71:A9 35 85 01 B1 FB 85 66 5D
0D79:C8 B1 FB 85 67 A0 10 B1 CB
0D81:FB 85 69 C8 B1 FB 85 6A A7
0D89:68 85 01 58 A5 69 38 E5 08
0D91:66 85 FD A5 6A E5 67 85 99
0D99:FE A5 69 C9 00 A5 6A E9 BB
0DA1:C0 90 17 A5 66 C9 00 A5 7D
0DA9:67 E9 E0 B0 0D A5 FD 38 4C
0DB1:E9 00 85 FD A5 FE E9 20 6E
0DB9:85 FE AD 7A CA 18 69 12 4F
0DC1:C5 FB D0 28 AD 7A CA 18 5F
0DC9:69 12 AD 7B CA 69 00 C5 4C
0DD1:FC D0 19 A5 66 8D 78 CA 41
0DD9:A5 67 8D 79 CA AD 7A CA B6
0DE1:18 69 12 8D 7A CA 90 03 A0
0DE9:EE 7B CA 60 A0 00 78 A5 55
0DF1:01 48 A9 35 85 01 B1 69 24
0DF9:91 66 68 85 01 58 E6 69 7C
0E01:D0 02 E6 6A A5 69 C9 00 EF
0E09:D0 0E A5 6A C9 C0 D0 08 67
0E11:A9 00 85 69 A9 E0 85 6A 8F
0E19:E6 66 D0 02 E6 67 A5 66 03
0E21:C9 00 D0 0E A5 67 C9 C0 3C
0E29:D0 08 A9 00 85 66 A9 E0 DE
0E31:85 67 A5 69 CD 78 CA D0 EB
0E39:B5 A5 6A CD 79 CA D0 AE 0B
0E41:AD 78 CA 38 E5 FD 85 6B CC
0E49:AD 79 CA E5 FE 85 6C AD E6
0E51:78 CA C9 00 AD 79 CA E9 68
0E59:C0 90 17 A5 6B C9 00 A5 5F
0E61:6C E9 E0 B0 0D A5 6B 38 63
0E69:E9 00 85 6B A5 6C E9 20 B4
0E71:85 6C A5 6B 8D 78 CA A5 60
0E79:6C 8D 79 CA A5 FB 38 E9 82
0E81:01 85 6D A5 FC E9 00 85 9C
0E89:6E A0 00 78 A5 01 48 A9 F7
0E91:35 85 01 B1 6D A0 12 91 88
0E99:6D 68 85 01 58 A6 6D D0 50
0EA1:02 C6 6E C6 6D A5 6D CD 55
0EA9:7A CA D0 DD A5 6E CD 7B AB
0EB1:CA D0 D6 AD 7A CA 18 69 B5
0EB9:12 8D 7A CA 90 03 EE 7B 28
0EC1:CA 78 A5 01 48 A9 35 85 FE
0EC9:01 A0 10 B1 FB 85 6B C8 41
0ED1:B1 FB 85 6C 68 85 01 58 F0
0ED9:A5 6B C9 00 A5 6C E9 C0 50
0EE1:08 A5 6B 38 E5 FD 85 6B F9
0EE9:A5 6C E5 FE 85 6C 28 90 5F
0EF1:17 A5 6B C9 00 A5 6C E9 66
0EF9:E0 B0 0D A5 6B 38 E9 00 BE
0F01:85 6B A5 6C E9 20 85 6C 7F
0F09:A0 10 A5 6B 91 FB C8 A5 9A
0F11:6C 91 FB A5 FB 38 E9 12 4A
0F19:85 FB B0 02 C6 FC A5 FB A0
0F21:CD 7A CA D0 9C A5 FC CD 6E
0F29:7B CA D0 95 60 85 6F A9 CC
0F31:DC 85 FB A9 FF 85 FC A5 EE
0F39:FB CD 7A CA D0 07 A5 FC AF
0F41:CD 7B CA F0 3B A0 00 78 62
0F49:A5 01 48 A9 35 85 01 B1 91
0F51:FB AA 68 85 01 58 8A D1 CD
0F59:22 D0 18 C8 C4 6F D0 E7 B9
0F61:18 78 A5 01 48 A9 35 85 47
0F69:01 B1 FB AA 68 85 01 58 52
0F71:8A F0 0D A5 FB 38 E9 12 B3
0F79:85 FB B0 BB C6 FC D0 B7 AE
0F81:60 20 9E AD 20 A3 B6 C9 4D
0F89:00 F0 16 C9 10 B0 17 60 15
0F91:A8 A9 00 91 FB 88 B1 22 0F
0F99:91 FB 88 10 F9 60 A2 04 2C
0FA1:2C A2 08 4C 37 A4 A2 C2 98
0FA9:A0 C7 4C BB C7 A2 D4 A0 62
0FB1:C7 4C BB C7 A2 DF A0 C7 58
0FB9:4C BB C7 86 22 84 23 4C 04
0FC1:47 A4 46 49 4C 45 20 4E 10
0FC9:41 4D 45 20 54 4F 4F 20 25
0FD1:4C 4F 4E C7 46 49 4C 45 65
0FD9:20 45 58 49 53 54 D3 52 DE
0FE1:41 4D 20 44 49 53 4B 20 8A
0FE9:46 55 4C CC 00 00 FF FF D6
0FF1:00 00 FF FF 00 00 FF FF 10
0FF9:00 00 FF FF A5 33 C9 01 A6
1001:A5 34 E9 80 B0 0A A5 37 76
1009:C9 01 A5 38 E9 80 90 0C 05
1011:A9 00 85 33 85 37 A9 80 C6
1019:85 34 85 38 A9 00 85 61 F6
1021:A9 C0 85 62 A9 7C 38 E5 B2
1029:61 85 63 A9 CA E5 62 85 9A
1031:64 A5 63 18 69 01 85 65 9A
1039:A5 64 69 08 85 66 A5 65 69
1041:18 69 04 85 69 A5 66 69 B8
1049:00 85 6A A0 00 B1 65 8D 41
1051:78 CA 38 E9 00 85 6B C8 BB
1059:B1 65 8D 79 CA E9 80 85 79
1061:6C AD 78 CA C9 00 AD 79 02
1069:CA E9 E0 90 0D A5 6B 38 9C
1071:E9 00 85 6B A5 6C E9 20 C0
1079:85 6C A9 01 85 02 A9 08 4C
1081:85 03 A0 00 B1 02 91 61 53
1089:E6 02 D0 02 E6 03 E6 61 4A
1091:D0 02 E6 62 A6 63 D0 02 04
1099:C6 64 C6 63 A5 63 05 64 6E
10A1:D0 E2 A0 02 B1 65 8D 7A CF
10A9:CA C8 B1 65 8D 7B CA A9 87
10B1:00 85 02 A9 80 85 03 A5 D3
10B9:6B 05 6C F0 34 A0 00 B1 43
10C1:69 91 02 E6 69 D0 02 E6 23
10C9:6A E6 02 D0 02 E6 03 A5 7D
10D1:02 C9 00 D0 0E A5 03 C9 49
10D9:C0 D0 08 A9 00 85 02 A9 ED
10E1:E0 85 03 A6 6B D0 02 C6 08
10E9:6C C6 6B A5 6B 05 6C D0 D2
10F1:CE AD 7A CA 85 02 AD 7B EB
10F9:CA 85 03 A0 00 B1 69 91 76
1101:02 E6 69 D0 02 E6 6A E6 7F
1109:02 D0 02 E6 03 A5 02 05 C6
1111:03 D0 EA 20 30 C0 20 42 4F
1119:A6 4C 86 E3 A9 31 A0 C9 CD
1121:20 1E AB 20 E4 FF C9 59 66
1129:F0 2B C9 4E D0 F5 4C D2 76
1131:FF 0D 45 58 49 54 20 52 F2
1139:41 4D 20 44 49 53 4B 0D D2
1141:41 52 45 20 59 4F 55 20 16
1149:53 55 52 45 3F 20 28 59 2D
1151:2F 4E 29 20 00 20 D2 FF EB
1159:4C 8F C9 02 A9 C0 85 03 3D
1161:A0 00 B1 02 38 E9 00 91 25
1169:02 C8 B1 02 E9 C0 91 02 8C
1171:A0 00 B1 02 18 69 01 91 34
1179:02 C8 B1 02 69 08 91 02 B5
1181:A9 4C 8D 58 C9 A9 8F 8D 64
1189:59 C9 A9 C9 8D 5A C9 A9 AF
1191:00 85 61 A9 C0 85 62 A9 66
1199:01 85 2B 85 02 A9 08 85 A7
11A1:2C 85 03 A0 00 B1 61 91 C0
11A9:02 E6 61 D0 02 E6 62 E6 17
11B1:02 D0 02 E6 03 A5 61 C9 F2
11B9:7C D0 EA A5 62 C9 CA D0 A6
11C1:E4 AD 78 CA 91 02 C8 AD 51
11C9:79 CA 91 02 C8 AD 7A CA 6A
11D1:91 02 C8 AD 7B CA 91 02 5D
11D9:A5 02 18 69 04 85 02 90 B3
11E1:02 E6 03 A9 00 85 61 A9 3C
11E9:80 85 62 A0 00 A5 61 CD 2B
11F1:78 CA D0 07 A5 62 CD 79 59
11F9:CA F0 32 78 A5 01 48 A9 F6
1201:35 85 01 B1 61 91 02 68 1A
1209:85 01 58 E6 61 D0 02 E6 E2
```

```
1211:62 E6 02 D0 02 E6 03 A5 C4
1219:61 C9 00 D0 D0 A5 62 C9 19
1221:C0 D0 CA A9 00 85 61 A9 50
1229:E0 85 62 D0 C0 AD 7A CA F4
1231:85 61 AD 7B CA 85 62 78 87
1239:A5 01 48 A9 35 85 01 B1 87
1241:61 91 02 68 85 01 58 E6 09
1249:61 D0 02 E6 62 E6 02 D0 84
1251:02 E6 03 A5 61 05 62 D0 9F
1259:DE A5 02 85 2D A5 03 85 7A
1261:2E A0 05 B9 72 CA 99 04 F6
1269:03 88 10 F7 20 60 A6 4C CE
1271:86 E3 7C A5 1A A7 E4 A7 9C
1279:00 80 DC FF 00 80 DC FF 15
1281:00 00 FF FF 00 00 FF FF A5
1289:00 00 FF FF 00 00 FF FF AD
1291:00 00 32 34 4B 20 52 41 FF
1299:4D 20 44 49 53 4B 00 00 51
12A1:00 00 00 80 00 00 00 00 CD
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

No-SYS Loader

See instructions in article on page 58 before typing in.

```
0801:0B 08 0A 00 9E 32 30 36 2E
0809:31 00 00 00 20 E7 FF A9 FB
0811:00 A2 36 A0 03 20 BD FF AE
0819:A0 00 20 1E 09 A9 08 85 E3
0821:BA A9 C0 85 9D A9 00 85 82
0829:93 85 90 A5 BA A4 B7 D0 79
0831:05 A2 08 4C 37 A4 20 B3 72
0839:F5 A9 60 85 B9 20 D5 F3 01
0841:A5 BA 20 09 ED A5 B9 20 01
0849:C7 ED 20 13 EE 8D 3B 0A 1C
0851:AA A5 90 4A 4A 90 05 A2 18
0859:04 4C 37 A4 A0 63 20 2F B1
0861:F1 20 C1 F5 A0 55 20 3B DF
0869:09 20 13 EE 8D 41 0A 8E 6B
0871:83 0A 8D 84 0A 20 CD BD E9
0879:AD 41 0A C9 09 B0 0B 20 CF
0881:28 F5 A0 33 20 3B 09 4C B6
0889:74 A4 38 A9 85 85 AE E9 28
0891:00 8D 2E 0A A9 0A 85 AF 9B
0899:E9 02 8D 36 0A 20 F0 F4 DB
08A1:A0 12 20 3B 09 20 60 A5 6D
08A9:86 7A 84 7B 20 73 00 B0 63
08B1:0D 20 6B A9 A5 14 8D 83 74
08B9:0A A5 15 8D 84 0A A0 75 B6
08C1:20 1E 09 A9 20 D0 03 B9 29
08C9:36 03 99 11 0A 88 10 F7 84
08D1:A0 64 20 1E 09 38 A5 AE 54
08D9:E9 85 8D 2A 0A A5 AF E9 C4
08E1:0A 8D 32 0A A9 01 85 AC 4A
08E9:A9 0A 85 AD A9 61 85 B9 74
08F1:A4 B7 20 D5 F3 20 8F F6 D9
08F9:A5 BA 20 0C ED A5 B9 20 E9
0901:B9 ED A0 00 A9 01 20 DD EE
0909:ED A9 08 20 DD ED 20 24 8A
0911:F6 A5 90 F0 05 A0 47 20 82
0919:3B 09 4C 74 A4 20 3B 09 01
0921:A0 00 20 CF FF C9 22 F0 E0
0929:F9 C9 0D F0 0A 99 36 03 81
0931:C8 C0 15 90 ED A0 14 84 22
0939:B7 60 B9 47 09 F0 06 20 23
0941:D2 FF C8 D0 F5 60 53 4F 0A
0949:55 52 43 45 20 46 49 4C 50
0951:45 2D 4E 41 4D 45 3A 00 23
0959:0D 0D 53 59 53 2D 41 44 4B
0961:44 52 28 44 45 43 29 20 1D
0969:43 52 3D 53 54 41 52 54 2F
0971:20 4F 46 20 50 47 4D 3A A6
0979:00 0D 46 49 4C 45 20 41 25
0981:44 44 52 20 54 4F 4F 20 B1
0989:4C 4F 57 21 00 0D 53 41 AE
0991:56 45 20 45 52 52 4F 52 45
0999:21 21 00 20 20 20 53 54 03
09A1:41 52 54 20 41 44 44 52 6B
09A9:3A 00 0D 0D 4E 45 57 20 A1
09B1:46 49 4C 45 2D 4E 41 4D 89
09B9:45 3A 00 0D 52 45 4D 41 51
09C1:52 4B 53 3A 00 8D 5C 08 D4
09C9:A0 2D A2 A8 86 0E 84 0F DA
09D1:A0 00 98 91 0E A0 67 A2 C4
09D9:FF 86 14 84 15 A0 68 A2 F6
09E1:00 8E 76 09 8C 77 09 20 6B
09E9:E9 0B A9 01 8D 4F 08 A9 5C
09F1:00 8D 50 08 A9 05 A0 FF 94
09F9:A2 00 20 BA FF A9 00 20 D3
0A01:0B 08 0A 00 9E 32 30 38 34
0A09:38 00 26 08 14 00 8F 20 5E
0A11:20 20 20 20 20 20 20 20 25
0A19:20 20 20 20 20 20 20 20 2D
0A21:20 20 20 20 00 00 00 18 6B
0A29:A9 00 85 FB 69 00 85 25 FD
0A31:A9 00 85 FC 69 00 85 26 17
0A39:18 A9 00 65 FB 85 27 A9 08
0A41:00 65 FC 85 28 A2 FF A0 13
0A49:00 8C 01 08 8C 02 08 F0 8E
0A51:04 B1 25 91 27 C6 25 E4 15
0A59:25 D0 02 C6 26 C6 27 E4 60
0A61:27 D0 02 C6 28 C6 FB E4 23
0A69:FB D0 E6 C6 FC E4 FC D0 3F
0A71:E0 A5 28 C9 A0 B0 0A 85 62
0A79:34 85 38 A5 27 85 33 85 A5
0A81:37 4C 00 80 00 00 00 00 4C
```

128 File Viewer

See instructions in article on page 46 before typing in.

```
1350:20 97 13 20 FA 51 78 A9 88
1358:6F 8D 14 03 A9 13 8D 15 16
1360:03 58 A9 56 8D 00 0A A9 E2
1368:13 8D 01 0A 4C 03 40 A0 CB
1370:06 88 30 1B B9 00 08 D9 2B
1378:C9 13 F0 F5 20 97 13 BA 06
1380:A9 00 9D 01 01 A9 13 9D B1
1388:07 01 A9 91 9D 06 01 4C 14
1390:65 FA A2 10 4C 29 14 A0 4D
1398:06 A9 08 8D 01 03 85 FB 21
13A0:85 30 A9 00 8D 00 03 85 C2
13A8:FA 84 2F A9 EF 8D 27 03 F4
13B0:A9 79 8D 26 03 88 A9 FA A6
13B8:8D B9 02 A2 01 B9 C9 13 14
13C0:91 FA 20 77 FF 88 10 F3 20
13C8:60 8D 03 FF 4C CF 13 20 CA
13D0:51 16 E0 0B D0 53 A0 FF 07
13D8:A2 04 A5 3E 85 FB A5 3D 8E
13E0:38 E9 04 85 FA B0 02 C6 DB
13E8:FB C8 CA 30 09 B9 36 14 4B
13F0:D1 FA F0 F5 D0 1C 24 7F FA
13F8:30 21 A0 00 C6 FA 30 3A 50
1400:B1 FA C9 20 F0 F4 C9 3A 24
1408:D0 08 F0 2E A2 0E 2C A2 E3
1410:17 2C A2 0B 2C A2 09 2C FD
1418:A2 04 2C A2 22 2C A2 08 51
1420:2C A2 05 2C A2 1E 2C A2 F2
1428:80 86 FC 20 A3 17 20 97 25
1430:13 A6 FC 4C 3F 4D 56 49 15
1438:45 57 20 51 16 20 86 03 33
1440:C9 23 D0 32 20 46 17 90 2C
1448:06 C9 2C F0 0E D0 C3 29 DE
1450:0F 20 27 17 20 80 03 D0 38
1458:EE F0 B7 E6 3D A5 FC F0 05
1460:AB 30 A9 20 59 FF 90 04 91
1468:A2 03 D0 C7 85 FE E0 04 27
1470:90 A3 A2 00 86 FC A6 98 2C
1478:E0 0A 90 04 A2 01 D0 B3 54
1480:20 CC FF 20 87 16 C9 2C 42
1488:F0 09 A2 3F 86 FB D0 1C 95
1490:4D 50 53 C8 A2 02 B1 3D 28
1498:DD 90 14 F0 06 CA 10 F6 D7
14A0:4C 12 14 E6 3D E6 3D 88 EC
14A8:85 FB B1 3D 20 B4 16 20 BC
14B0:90 03 F0 07 C9 22 D0 E8 D1
14B8:20 D4 16 A5 FC D0 04 A9 20
14C0:08 85 FC A0 02 98 20 59 04
14C8:FF B0 04 A8 C8 10 F6 85 22
14D0:FD 85 05 20 53 17 A5 FE 3D
14D8:F0 0A A9 17 8D 27 03 A9 5B
14E0:B0 8D 26 03 20 DB 17 A4 FC
14E8:FB C0 53 F0 0F C0 50 F0 C5
14F0:21 C0 4D D0 03 4C A8 15 40
14F8:C9 01 F0 16 20 D2 FF 8A 9C
1500:20 D2 FF 20 CF FF 24 90 48
1508:08 20 D2 FF 68 20 6E 16 4F
1510:D0 F1 A0 20 A9 6A 85 FA 32
1518:A9 51 85 FB B1 FA 99 00 88
1520:13 88 10 F8 A9 05 8D 1B 1F
1528:13 C8 B9 CF 17 99 21 13 B6
1530:C8 C0 06 90 F5 A9 0D 20 49
1538:D2 FF A0 03 20 CF FF 99 E9
1540:FA 00 88 10 F7 24 90 08 73
1548:68 20 6E 16 C8 B9 FA 00 01
1550:99 64 00 C0 01 90 F5 A2 45
1558:90 38 20 75 8C 20 44 8E 30
1560:20 E2 55 A9 20 20 D2 FF BF
1568:20 CF FF F0 C8 C9 CE F0 A1
1570:25 C9 FE F0 24 30 15 20 BA
1578:D2 FF C9 22 D0 EA 20 CF A9
1580:FF F0 B2 20 D2 FF C9 22 8B
1588:F0 DE D0 F2 AA A9 44 A0 51
1590:17 20 00 13 D0 D2 A0 C9 5C
1598:2C A0 09 20 CF FF 09 80 35
15A0:18 E9 01 AA A9 46 D0 E9 0E
15A8:48 A0 49 B9 0F B6 99 00 6A
15B0:13 88 10 F7 A9 FB 8D 3F 9F
15B8:13 A9 FA 8D 45 13 A9 13 EB
15C0:8D 34 13 A9 3E 8D 33 13 5D
15C8:68 85 FA 85 66 86 FB 86 0C
15D0:67 A9 0D 20 D2 FF 20 9B 2F
15D8:B8 A9 2D 20 D2 FF 20 CF 18
15E0:FF 24 90 08 48 20 A5 B8 6D
15E8:68 A0 00 84 67 84 68 99 6F
15F0:06 00 20 59 B6 48 AE AB 97
15F8:0A 86 FC C8 CA 30 0C 20 45
1600:CF FF 99 06 00 20 A5 B8 2C
1608:4C FB 15 A9 12 20 60 16 7E
1610:68 A2 03 20 A1 B6 A2 06 AE
1618:86 66 20 00 13 A9 20 20 C4
1620:60 16 A9 3B 20 D2 FF A0 D7
1628:00 E6 FC B9 06 00 48 29 33
1630:7F C9 20 68 B0 02 A9 20 1A
1638:20 D2 FF C8 C4 FC 90 EB DC
1640:68 20 6E 16 A6 FB A5 FA 43
1648:18 65 FC 90 01 E8 4C C9 90
1650:15 A9 00 A0 04 99 FA 00 F7
1658:88 10 FA 85 05 85 7A 60 18
1660:38 E5 EC A8 A9 20 20 D2 2B
1668:FF 88 D0 FA 18 60 48 28 7B
1670:70 12 20 E1 FF F0 01 60 A1
1678:A5 FE F0 05 A9 0D 20 D2 3A
1680:FF 4C 24 14 4C 27 14 20 CC
1688:86 03 C9 22 F0 02 D0 5B A0
1690:A9 30 8D 00 11 A9 3A 8D 80
1698:01 11 E6 3D B1 3D F0 13 B1
16A0:C9 22 F0 0F C9 2C F0 0B 35
16A8:99 02 11 C8 C0 11 D0 EC A9
16B0:4C 0F 14 60 48 C0 00 D0 65
16B8:03 4C 1E 14 98 48 A9 2C E3
16C0:99 02 11 C8 A5 FB 99 02 3B
16C8:11 C8 84 FA 18 68 65 3D 5A
16D0:85 3D 68 60 20 37 17 C9 F7
16D8:44 F0 06 C9 55 F0 1C D0 38
16E0:0A 20 46 17 90 0B 20 3A 7F
16E8:17 B0 10 4C 12 14 4C 0C 19
16F0:14 8D 00 11 C9 32 B0 F6 0B
16F8:20 37 17 C9 55 D0 EC 20 6A
1700:46 17 B0 E7 F0 E5 29 0F 2C
1708:85 FC 20 80 03 F0 0A B0 E4
1710:DA 29 0F 20 27 17 4C 0A 12
1718:17 A5 FC C9 08 B0 03 4C CC
1720:15 14 C9 1F B0 F9 60 48 7F
1728:A5 FC 0A 0A 65 FC 0A 85 03
1730:FC 68 65 FC 85 FC 60 20 74
1738:80 03 F0 14 C9 2C F0 03 AA
1740:4C 12 14 4C 80 03 20 80 31
1748:03 C9 30 F0 F9 4C 86 03 90
1750:68 68 60 A9 00 8D 00 FF A9
1758:AA 20 68 FF A5 FA A0 11 5C
1760:20 BD FF A5 05 A6 FC A4 C9
1768:FD 20 BA FF 20 0F F5 E6 05
```

```
1770:B7 E6 B7 20 C0 FF B0 28 BC
1778:A6 05 20 C6 FF 20 CF FF CB
1780:48 A9 02 24 90 D0 16 20 D3
1788:CF FF AA 68 20 D4 17 20 1D
1790:7D FF 0D 56 49 45 57 49 DB
1798:4E 47 0D 00 60 4C 18 14 D9
17A0:4C 21 14 20 CC FF A5 05 78
17A8:20 C3 FF A5 FE 4C C3 FF E2
17B0:20 D4 17 20 CC FF A6 FE BB
17B8:20 C9 FF 20 DB 17 20 79 60
17C0:EF 20 CC FF A6 05 20 C6 D8
17C8:FF 20 DB 17 4C 79 EF 29 3D
17D0:7F 4C D2 FF 85 41 86 42 AC
17D8:84 43 60 A5 41 A6 42 A4 4E
17E0:43 60 00 00 00 00 00 00 C8
```

Miami Ice

Article on page 17.

Program 1: Miami Ice For Commodore 128

```
EF 10 OPEN2,8,2,"HI-SCORE,S,W"
      :CLOSE2:OPEN15,8,15:INPU
      T#15,A$,B$:IFB$<>"FILE E
      XISTS"THENCLOSE15:GOSUB7
      50
EM 20 COLOR0,16:COLOR4,11
BR 30 PRINT"{CLR}{RED}{7 DOWN}
      {15 RIGHT}{RVS}MIAMI ICE
      "
SD 40 PRINT"{BLU}{DOWN}
      {11 SPACES}JOYSTICK IN P
      ORT 2"
JJ 50 PRINT"{BLK}{DOWN}{BLK}
      {11 SPACES}[LEFT]
      {2 SPACES}TURN LEFT":PRI
      NT"{11 SPACES}[RIGHT] TU
      RN RIGHT"
QC 60 PRINT"{11 SPACES}[FIRE]
      {2 SPACES}ACCELERATE":PR
      INT"{YEL}{DOWN}
      {13 SPACES}READING DATA.
      .."
BD 70 GOSUB1800:PRINT"{UP}
      {BLU}{10 SPACES}PRESS BU
      TTON TO PLAY"
DB 80 IFJOY(2)<>128THEN80
BE 90 HY=3:SC=0:SN=1
XD 100 FAST:ONSNGOSUB760,1020,
       1280,1550:SLOW:PRINT"
       {HOME}"TAB(32);"{BLK}LI
       VES";HY:COLOR0,16:TM=40
       0:T=0:XE=0
KC 110 GOSUB540:
AH 120 POKE2041,62:MOVSPR2,X,Y
       :SPRITE2,1,2,0,0,0,1:PO
       KE2040,57:XE=BUMP(2)
MX 130 MOVSPR1,30#0:SPRITE1,1,
       9,0,0,0,1:SPRCOLOR1,2:M
       OVSPR1,40,65:I=4:AN=180
       :HT=135:TH=0:XE=BUMP(2)
       +BUMP(1)
BQ 140 PRINT"{HOME}{RVS}";TM;"
       {LEFT} {OFF}"
JX 150 IFJOY(2)=0THEN150
PH 160 IFJOY(2)=3THEN280
GC 170 IFJOY(2)=7THEN310
PR 180 IFJOY(2)=128THENMOVSPR1
       ,AN#1:TH=1:SOUND1,5000,
       24,2,1000,3,3
XG 190 POKE2040,53+I:IFBUMP(1)
       =3THEN490:ELSEIFBUMP(2)
       AND1THEN420
PC 200 IFTH>1THENONABS(T-20)GO
       TO410
RC 210 T=T+1
KS 220 IFHT>180THENIFHT-180>AN
       THEN360
RS 230 IFHT>180THENIFHT-180<AN
       THEN370
KR 240 IFHT<180THENIFHT+180<AN
       THEN380
RD 250 IFHT<180THENIFHT+180>AN
       THEN390
MM 260 TM=TM-1
CC 261 IFTM<0THENTM=0
XP 262 PRINT"{HOME}{RVS}";TM;"
       {LEFT} {OFF}"
MH 270 IFJOY(2)<>3THEN300
AQ 280 AN=AN+45:IFAN>360THENAN
       =45
KD 290 I=I-1:IFI=0THENI=8:GOTO
       190
DS 300 IFJOY(2)<>7THEN340
CF 310 AN=AN-45:IFAN<0THENAN=3
       15
KE 320 IFAN=360THENAN=0
DD 330 I=I+1:IFI=9THENI=1:GOTO
       190
HH 340 IFJOY(2)=128THENSOUND1,
       5000,24,2,1000,3,3:TH=T
       H+1:T=0:IFTH>15THENTH=1
       5:GOTO190
CP 350 GOTO190
FG 360 HT=HT+((AN+(360-HT))/10
       ):MOVSPR1,HT#TH:IFHT<36
       0THEN260:ELSE:HT=0:GOTO
       390
RX 370 HT=HT-((HT-AN)/10):MOVS
       PR1,HT#TH:GOTO260
SM 380 HT=HT-((HT+(360-AN))/10
       ):MOVSPR1,HT#TH:IFHT>0T
       HEN260:ELSE:HT=360:GOTO
       370
JQ 390 HT=HT+((AN-HT)/10):MOVS
       PR1,HT#TH:GOTO260
QR 400 GOTO260
XD 410 TH=TH-1:T=0:IFTH<1THENT
       H=1:GOTO220:ELSE220
KF 420 POKE2040,63:FORDELAY=1T
       O150:NEXT:SPRITE 1,0
RJ 430 SOUND1,2000,100,0,1000,
       1,3,100
JB 440 T=0:HY=HY-1:PRINT"
       {HOME}"TAB(32)"LIVES";H
       Y:IFHY=0THEN460
CD 450 XE=BUMP(2):SLEEP2:POKE2
       040,57:MOVSPR1,33,55:XE
       =BUMP(2):GOTO130
HX 460 SLEEP2:PRINT"{9 DOWN}
       {15 RIGHT}{RVS}{BLK}GAM
       E OVER{OFF}"
PD 470 OPEN2,8,2,"HI-SCORE,S,R
       ":INPUT#2,A$,B$:CLOSE2:
       IFSC>VAL(A$)THENFORI=1T
       O8:SPRITEI,0:NEXT:GOTO5
       60
HK 480 IFJOY(2)<>128THEN480:EL
       SE:XE=BUMP(2):GOTO90
AS 490 POKE53280,6:MOVSPR1,40#
       0:PLAY"QGRGRG":IFTM=0TH
       EN520
GP 492 SD=INT(2000/TM)
XG 500 FORTY=1TOTMSTEP5:PRINT"
       {HOME}{RVS}";TM-TY:PRIN
       T"{HOME}{RVS}";TAB(13);
       SC+TY:IFTM-TY<99THENIFT
       M-TY>90THENPRINT"{HOME}
       {RVS}{4 SPACES}{OFF}"
HM 510 SOUND1,3000+(SD*TY),1:N
       EXT:SC=SC+TM:PRINT"
       {HOME}{RVS}{2 SPACES}0
       {SPACE}{OFF}";TAB(13);"
       {RVS}";SC
PC 520 SLEEP1:SN=SN+1:IFSN=5TH
       ENSN=1
CB 530 GOTO100
QK 540 PRINT"{HOME}{RVS}";TM;"
       {HOME}{RVS}"TAB(13);SC:
       RETURN
XB 550 REM ********** HI SCORE
        **********
GB 560 PLAY"O4SCCFGBBAR AB":PR
       INT"{CLR}{2 DOWN}
       {11 SPACES}YOUR SCORE:
       {SPACE}";SC:AB=65:OP=0
JG 570 PRINT"{10 SPACES}CCCCCC
       CCCCCCCCCCCCCC{UP}"
PB 580 OPEN2,8,2,"HI-SCORE,S,R
       ":FORI=1TO10:INPUT#2,B$
       (I):INPUT#2,A$(I):NEXT:
       CLOSE2:SCRATCH"HI-SCORE
       "
GB 590 FORU=1TO10:IFSC>VAL(B$(
       U))THENNEXT
FX 600 U=U-1:FORE=1TOU-1:A$(E)
       =A$(E+1):B$(E)=B$(E+1):
       NEXT:B$(U)=RIGHT$(STR$(
       SC),LEN(STR$(SC))-1):A$
       (U)="---"
PB 610 TE=LEN(B$(U)):FORP=1TO6
       -TE:B$(U)="0"+B$(U):NEX
       T
FG 620 PRINT"{2 DOWN}":FORE=10
       TO2STEP-1:PRINTTAB(11);
       11-E;"{2 SPACES}";A$(E)
       ;"{3 SPACES}";B$(E):NEX
       T
SF 630 PRINTTAB(10);10;"
       {2 SPACES}";A$(1);"
       {3 SPACES}";B$(1)
EQ 640 PRINT"{HOME}{5 DOWN}":F
       ORI=1TO11-U:PRINT:NEXT:
       NM$=""
CM 650 PRINT"{UP}"TAB(16+OP);C
       HR$(AB)
GF 660 IFJOY(2)=7THENAB=AB-1:I
       FAB<65THENAB=65:GOTO650
PK 670 IFJOY(2)=3THENAB=AB+1:I
       FAB>90THENAB=90:GOTO650
BD 680 IFJOY(2)=128THENNM$=NM$
       +CHR$(AB):AB=65:OP=OP+1
       :SLEEP1:IFOP=3THEN700
XA 690 GOTO650
ER 700 A$(U)=NM$:OPEN2,8,2,"HI
       -SCORE,S,W":FORI=1TO10:
       PRINT#2,B$(I):PRINT#2,A
       $(I):NEXT:CLOSE2
GF 710 PRINT"{HOME}":FORI=1TO1
       8:PRINT:NEXT
MG 720 PRINT"{7 SPACES}PRESS B
       UTTON TO PLAY AGAIN":GO
       TO480
FM 740 REM ******* CLEAR HI-SC
       ORES *******
GH 750 SCRATCH"HI-SCORE":PRINT
       "{CLR}MAKING HI-SCORE":
       OPEN2,8,2,"HI-SCORE,S,W
       ":FORI=1TO10:PRINT#2,"0
       00000":PRINT#2,"---":NE
       XT:CLOSE2:RETURN
DB 760 X=62:Y=135:COLOR4,16
KG 770 PRINT"{CLR}{RVS}{RED}
       {28 SPACES}{C3}"
JX 780 PRINT"{RVS} {OFF}
       {24 SPACES}{WHT}++{RED}
       {C3}{RVS}{9 SPACES}{C3}
       {OFF}"
KG 790 PRINT"{RVS} {OFF}
       {25 SPACES}{WHT}{Q3}
       {5 SPACES}{Q3}+{W3}{RED}
       {C3}{RVS}{3 SPACES}{C3}
       {OFF}"
AH 800 PRINT"{RVS} {OFF}
       {31 SPACES}{53}{RVS}
       {3 SPACES}{OFF}
       {2 SPACES}{RED}{C3}{RVS}
       {2 SPACES}{OFF}"
CD 810 PRINT"{RVS} {OFF}
       {37 SPACES}{RVS}
       {2 SPACES}{OFF}"
FC 820 PRINT"{RVS} {C3}{OFF}
       {36 SPACES}{C3}{RVS}
       {OFF}"
```

```
EX 830 PRINT"{RVS}{9 SPACES}
       {C}{OFF}{29 SPACES}
       {RVS}{2 SPACES}{OFF}
       {WHT}++++{RED}
       {3 SPACES}{C}{RVS}
       {4 SPACES}{C}{OFF}
       {24 SPACES}{BLK}{Q}
       {RED}{RVS} {OFF}"
HS 840 PRINT"{RVS} {OFF}{WHT}+
       ++{RED}{5 SPACES}{WHT}+
       {RED}{C}{RVS}
       {15 SPACES}{C}{OFF}
       {11 SPACES}{BLK}{Q}
       {RED}{RVS} {OFF}"
CM 850 PRINT"{RVS} {OFF}
       {2 SPACES}{WHT}+
       {5 SPACES}{Z}++{RED}
       {RVS}{F}{3 SPACES}{OFF}
       {WHT}{W}{4 SPACES}+++
       {RED}{C}{RVS}{2 SPACES}
       {OFF}{11 SPACES}{BLK}
       {Q}{RED}{RVS} {OFF}"
HE 860 PRINT"{RVS} {OFF}
       {9 SPACES}{WHT}{Q}++
       {RED}{RVS}{2 SPACES}
       {OFF}{WHT}+{W}
       {5 SPACES}{Q}++{RED}{C}
       {RVS} {C}{OFF}
       {10 SPACES}{D}{RVS}
       {OFF}"
GB 870 PRINT"{RVS} {OFF}
       {10 SPACES}{WHT}{Z}{X}
       {RED}{RVS} {OFF}{WHT}+
       {W}{8 SPACES}{Q}+{RED}
       {RVS}{2 SPACES}{OFF}
       {9 SPACES}{BLK}{Q}{RED}
       {RVS}{2 SPACES}{OFF}"
DK 880 PRINT"{RVS} {OFF}
       {12 SPACES}{RVS} {OFF}
       {WHT}{W}{10 SPACES}{Q}
       {RED}{C}{RVS} {OFF}
       {9 SPACES}{BLK}{Q}{RED}
       {RVS}{2 SPACES}{OFF}"
AJ 890 PRINT"{RVS} {OFF}
       {12 SPACES}{T}
       {13 SPACES}{RVS} {OFF}
       {9 SPACES}{BLK}{Q}{RED}
       {RVS}{2 SPACES}{OFF}"
PQ 900 PRINT"{RVS} {OFF}
       {26 SPACES}{RVS} {OFF}
       {9 SPACES}{BLK}{Q}{RED}
       {RVS}{2 SPACES}{OFF}"
GE 910 PRINT"{RVS} {OFF}
       {37 SPACES}{C}{RVS}
       {OFF}"
CM 920 PRINT"{RVS} {OFF}
       {37 SPACES}{BLK}{Q}
       {RED}{RVS} {OFF}"
GR 930 PRINT"{RVS} {OFF}
       {37 SPACES}{BLK}{Q}
       {RED}{RVS} {OFF}"
MQ 940 PRINT"{RVS} {OFF}
       {37 SPACES}{BLK}{Q}
       {RED}{RVS} {OFF}"
EK 950 PRINT"{RVS} {OFF}{F}
       {11 SPACES}{@}
       {24 SPACES}{BLK}{Q}
       {RED}{RVS} {OFF}"
DQ 960 PRINT"{RVS}{2 SPACES}
       {OFF}{F}{10 SPACES}
       {RVS} {OFF} {WHT}{Q}
       {22 SPACES}{BLK}{Q}
       {RED}{RVS} {OFF}"
MK 970 PRINT"{RVS}{3 SPACES}
       {OFF}{10 SPACES}{RVS}
       {OFF}{WHT}+{W}
       {15 SPACES}{A}
       {6 SPACES}{RED}{D}{RVS}
        {OFF}"
SD 980 PRINT"{RVS}{3 SPACES}
       {OFF}{F}{WHT}{2 R}+{S}
       {4 SPACES}{RED}{D}{RVS}
       {OFF}{WHT}+++
       {12 SPACES}{A}+{W}
       {5 SPACES}{RED}{RVS}{V}
       {2 SPACES}{OFF}"
SJ 990 PRINT"{RVS}{4 SPACES}
       {OFF}{F}{WHT}+++{S}
       {RED} {D}{RVS}
       {3 SPACES}{C}{OFF}{WHT}
       {Q}++{2 R}+{6 SPACES}
       {Q}++++{S}{2 SPACES}
       {RED}{RVS}{V}{4 SPACES}
       {OFF}"
PQ 1000 PRINT"{RVS}{39 SPACES}
        {OFF}";:POKE2023,224:P
        OKE56295,2
DS 1010 RETURN
CC 1020 X=262:Y=142:COLOR4,3
KS 1030 PRINT"{CLR}{13}{RVS}
        {40 SPACES}{OFF}"
KQ 1040 PRINT"{13}{RVS} {OFF}
        {8 SPACES}{RVS} {OFF}
        {WHT}{Q}{13}{RVS} {OFF}
        {WHT}C{E}{X}
        {17 SPACES}{13}{RVS}{F}
        {7 SPACES}{OFF}"
XM 1050 PRINT"{RVS} {OFF}
        {8 SPACES}{RVS} {OFF}
        {WHT}{Q}{13}{RVS} {OFF}
        {27 SPACES}{RVS}{J}
        {OFF}{7 SPACES}{RED}
        {D}{13}{RVS}{3 SPACES}
        {OFF}{27 SPACES}{RVS}
        {J}{OFF}"
AB 1060 PRINT"{RVS} {OFF}
        {8 SPACES}{WHT}{M} {13}
        {RVS} {OFF}{BLK}{S}
        {26 SPACES}{13}{RVS}{J}
        {OFF}"
CG 1070 PRINT"{RVS} {OFF}
        {8 SPACES}{WHT}{M}
        {BLK}C+{X}{5 SPACES}
        {WHT}{Q}{RED}{RVS}{U}
        {OFF}{WHT}{S} {RED}
        {12 P}{5 SPACES}{13}
        {RVS}{J}{OFF}"
XC 1080 PRINT"{RVS} {OFF}
        {8 SPACES}{WHT}{M}
        {RED}{RVS} {OFF}
        {6 SPACES}{WHT}{Q}
        {RED}{RVS} {13}
        {14 SPACES}{OFF}{RED}
        {F}{4 SPACES}{13}{RVS}
        {J}{OFF}"
DB 1090 PRINT"{RVS} {OFF}
        {8 SPACES}{WHT}{M}
        {RED} {RVS} {OFF}
        {6 SPACES}{WHT}{Q}
        {RED}{RVS} {OFF} {WHT}
        {Q}{13}{RVS} {OFF}{WHT}
        {X} {Q}{13}{RVS} {OFF}
        {WHT}{X}{2 SPACES}{Q}
        {13}{RVS} {OFF}{WHT}{X}
        {13}{RVS} {BLK}{L}{OFF}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
CS 1100 PRINT"{RVS} {OFF}
        {8 SPACES}{WHT}{M}
        {RED} {RVS} {OFF}
        {6 SPACES}{WHT}{Q}
        {RED}{RVS} {OFF} {WHT}
        {Q}{13}{RVS} {OFF}
        {2 SPACES}{WHT}{Q}{13}
        {RVS} {OFF}{3 SPACES}
        {WHT}{Q}{13}{2 SPACES}
        {RVS} {OFF}{BLK}{W}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
KX 1110 PRINT"{RVS}{2 SPACES}
        {OFF}{8 SPACES}{WHT}
        {Y}{RED}{RVS} {OFF}
        {6 SPACES}{WHT}{Q}
        {RED}{RVS} {OFF}
        {2 SPACES}{RED}{C}
        {2 SPACES}{WHT}{Q}
        {4 SPACES}{RED}{C}
        {2 SPACES}{13}{RVS}
        {OFF}{BLK}{W}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
XR 1120 PRINT"{RVS} {OFF}
        {9 SPACES}{RED}{RVS}
        {2 SPACES}{OFF}
        {6 SPACES}{WHT}{Q}
        {RED}{RVS} {OFF}
        {5 SPACES}{C}
        {7 SPACES}{13}{RVS}
        {OFF}{BLK}{W}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
JS 1130 PRINT"{RVS} {OFF}
        {9 SPACES}{RED}{RVS}
        {OFF}{5 SPACES}{RED}
        {D}C{RVS} {D}{OFF}
        {13 SPACES}{13}{RVS}
        {OFF}{BLK}{W}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
RQ 1140 PRINT"{RVS} {OFF}
        {9 SPACES}{RED}{RVS}
        {OFF}{6 SPACES}{WHT}
        {Q}{RED}{RVS} {OFF}{H}
        {13 SPACES}{13}{RVS}
        {OFF}{BLK}{W}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
AR 1150 PRINT"{RVS} {OFF}
        {9 SPACES}{RED}{RVS}
        {OFF}{6 SPACES}{WHT}
        {Q}{RED}{RVS} {OFF}{H}
        {13 SPACES}{13}{RVS}
        {OFF}{BLK}{W}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
RJ 1160 PRINT"{RVS} {OFF}
        {9 SPACES}{RED}{RVS}
        {OFF}{6 SPACES}{WHT}
        {Q}{RED}{RVS} {OFF}{H}
        {6 SPACES}{13}{RVS}{F}
        {7 SPACES}{OFF}{BLK}
        {W}{4 SPACES}{13}{RVS}
        {J}{OFF}"
FR 1170 PRINT"{RVS} {BLK}{D}
        {F}{OFF}{7 SPACES}
        {RED}{RVS} {OFF}
        {6 SPACES}{WHT}{Q}
        {RED}{RVS} {OFF}
        {9 SPACES}{13}{RVS}
        {BLU}{D}{F}{D}{OFF}
        {13}{RVS} {OFF}{BLK}{W}
        {4 SPACES}{13}{RVS}{J}
        {OFF}"
XJ 1180 PRINT"{RVS} {8}
        {2 SPACES}{OFF}{BLK}
        {G}{4 SPACES}{RED}{D}
        {RVS}{2 SPACES}{OFF}
        {6 SPACES}{WHT}{Q}
        {RED}{RVS} {OFF}
        {9 SPACES}{13}{RVS}{D}
        {BLU}{3 SPACES}{OFF}
        {BLK}{S}{13}{RVS} {OFF}
        {BLK}{W}{4 SPACES}{13}
        {RVS}{J}{OFF}"
BC 1190 PRINT"{RVS} {8}{D}{F}
        {OFF}{BLK}{G}
        {13 SPACES}{WHT}{Q}
        {RED}{RVS} {OFF}
        {10 SPACES}{WHT}{3 T}
        {BLK}{Z}{13}{RVS} {OFF}
        {BLK}{X}{4 SPACES}{13}
        {RVS}{J}{OFF}"
QM 1200 PRINT"{RVS} {8}
        {2 SPACES}{OFF}{BLK}
        {G}{12 SPACES}{RED}
        {RVS}{3 SPACES}{OFF}
```

```
        {14 SPACES}{13}{RVS}{D}
        {OFF}{5 SPACES}{RVS}
        {J}{OFF}"
HS 1210 PRINT"{RVS} {8}{D}{F}
        {OFF}{BLK}{G}
        {12 SPACES}{RED}{RVS}
        {3 SPACES}{OFF}
        {20 SPACES}{13}{RVS}{J}
        {OFF}"
ER 1220 PRINT"{RVS} {8}{D}{F}
        {OFF}{WHT}{S}
        {12 SPACES}{RED}{RVS}
        {SPACE}{3}Q{RED} {OFF}
        {20 SPACES}{13}{RVS}{J}
        {OFF}"
DG 1230 PRINT"{RVS} {8}{D}{F}
        {OFF}{WHT}++
        {11 SPACES}T{RED}{RVS}
        {3 SPACES}{OFF}
        {20 SPACES}{13}{RVS}{J}
        {OFF}"
ER 1240 PRINT"{RVS} {8}
        {2 SPACES}{OFF}{WHT}++
        {S}{9 SPACES}{A}{RED}
        {RVS} {3}Q{RED} {OFF}
        {SPACE}{WHT} {H}
        {17 SPACES}{13}{RVS}
        {OFF}"
PK 1250 PRINT"{RVS} {8}
        {5 SPACES}{OFF}{WHT}CC
        CCCC{13}{RVS}{7 SPACES}
        {RED}{2 Y}{C}{OFF}
        {17 SPACES}{13}{RVS}
        {OFF}"
MR 1260 PRINT"{RVS}{39 SPACES}
        {OFF}";:POKE2023,224:P
        OKE56295,8
XB 1270 RETURN
BA 1280 X=280:Y=200:COLOR4,16
DK 1290 PRINT"{CLR}{GRN}{RVS}
        {40 SPACES}{OFF}"
DA 1300 PRINT"{RVS} {OFF}
        {19 SPACES}{*}{RVS}
        {4 SPACES}{OFF}{WHT}-
        {BLK}{13 E}{GRN}{RVS}
        {SPACE}{OFF}"
SG 1310 PRINT"{RVS} {OFF}
        {21 SPACES}{*}{RVS}
        {2 SPACES}{OFF}{WHT}-
        {13 SPACES}{GRN}{RVS}
        {SPACE}{OFF}"
JD 1320 PRINT"{RVS} {OFF}
        {22 SPACES}{RVS}
        {2 SPACES}{OFF}{WHT}-
        {13 SPACES}{GRN}{RVS}
        {SPACE}{OFF}"
MD 1330 PRINT"{RVS} {OFF}
        {13 SPACES}{RVS}£
        {OFF}{7 SPACES}{*}
        {RVS} {OFF}{WHT}-
        {13 SPACES}{GRN}{RVS}
        {SPACE}{OFF}"
XQ 1340 PRINT"{RVS}{15 SPACES}
        {OFF}£{6 SPACES}{WHT}
        UC{GRN}{RVS} {OFF}
        {WHT}K{13 SPACES}{GRN}
        {RVS} {OFF}"
CD 1350 PRINT"{RVS}{4 SPACES}
        {OFF}£{17 SPACES}
        {WHT}-{GRN}{RVS}£
        {OFF}T{7 SPACES}{RVS}
        {OFF}{6 SPACES}{RVS}
        {OFF}"
CH 1360 PRINT"{RVS}{3 SPACES}
        {OFF}£{17 SPACES}
        {RVS}£{3 SPACES}{OFF}
        {7 SPACES}{RVS} {OFF}
        {6 SPACES}{RVS} {OFF}"
DG 1370 PRINT"{RVS}{2 SPACES}
        {OFF}£{17 SPACES}
        {RVS}£{3 SPACES}{OFF}
        £{7 SPACES}{RVS}
        {OFF}{6 SPACES}{RVS}
        {OFF}"
DE 1380 PRINT"{RVS}{2 SPACES}
        {OFF}{8 SPACES}{5}
        {RVS}£{GRN}
        {11 SPACES}{OFF}£
        {9 SPACES}{RVS} {OFF}
        {6 SPACES}{RVS} {OFF}
RE 1390 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{5}
        {RVS}£{4 SPACES}{GRN}
        {2 SPACES}{OFF}£
        {14 SPACES}{RVS}£
        {OFF}{6 SPACES}{RVS}
        {OFF}"
JS 1400 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{RVS}
        {SPACE}{5}{4 SPACES}
        {GRN} {OFF}£
        {14 SPACES}{RVS}£
        {2 SPACES}{OFF}
        {6 SPACES}{RVS} {OFF}"
HH 1410 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{RVS}
        {SPACE}{5}{2 SPACES}
        {GRN}{2 SPACES}{OFF}£
        {14 SPACES}{RVS}£
        {3 SPACES}{OFF}
        {6 SPACES}{RVS} {OFF}"
QA 1420 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{RVS}
        {SPACE}{5}{2 SPACES}
        {GRN} {OFF}£
        {14 SPACES}{RVS}£
        {2 SPACES}M {OFF}
        {6 SPACES}{RVS} {OFF}
BS 1430 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{RVS}
        {3 SPACES}{OFF}£
        {14 SPACES}{RVS}£
        {2 SPACES}M {OFF}£
        {6 SPACES}{RVS} {OFF}"
RM 1440 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{RVS}
        {2 SPACES}{OFF}£
        {14 SPACES}{RVS}£
        {2 SPACES}M {OFF}£
        {7 SPACES}{RVS} {OFF}"
RP 1450 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{RVS}
        {SPACE}{OFF}£
        {14 SPACES}{RVS}£
        {2 SPACES}M {OFF}£
        {8 SPACES}{RVS} {OFF}"
CE 1460 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{7 SPACES}{RVS}
        {SPACE}{OFF}
        {14 SPACES}{RVS}£
        {4 SPACES}{OFF}£
        {9 SPACES}{RVS} {OFF}"
FJ 1470 PRINT"{RVS} {OFF}{WHT}
        -{GRN}{20 SPACES}{5}
        {RVS}£ {GRN}
        {5 SPACES}{OFF}
        {10 SPACES}{RVS} {OFF}
        "
KX 1480 PRINT"{RVS} {OFF}{WHT}
        JI{GRN}{18 SPACES}{5}
        {RVS}£ {GRN}
        {3 SPACES}{OFF}{WHT}-
        {SPACE}{GRN}{RVS}
        {OFF}{10 SPACES}{RVS}
        {SPACE}{OFF}"
SH 1490 PRINT"{RVS} {5}
        {3 SPACES}{OFF}
        {17 SPACES}{RVS} {GRN}
        {3 SPACES}{OFF} {WHT}
        {Q}{C}{GRN}{RVS} {OFF}
        {10 SPACES}{RVS} {OFF}
        "
MX 1500 PRINT"{RVS} {5}
        {3 SPACES}{OFF}
        {16 SPACES}{GRN}{RVS}
        £{4 SPACES}{OFF}{WHT}
        C{W} {GRN}{RVS} {OFF}
        {10 SPACES}{RVS} {OFF}
        "
FF 1510 PRINT"{RVS} {5}
        {4 SPACES}{OFF}{WHT}CC
        CI{10 SPACES}{GRN}
        {RVS}£{2 SPACES}{OFF}
        {WHT} - U{W} {GRN}
        {RVS} {OFF}{10 SPACES}
        {RVS} {OFF}"
QR 1520 PRINT"{RVS} {5}
        {7 SPACES}{OFF}{WHT}-
        {8 SPACES}{GRN}{D}
        {RVS}{4 SPACES}{OFF}
        {WHT}CK {RVS} {OFF}JC
        {GRN}{RVS} {OFF}{WHT}C
        CCCCCCCCC{GRN}{RVS}
        {OFF}"
KX 1530 PRINT"{RVS}{39 SPACES}
        {OFF}";:POKE2023,224:P
        OKE56295,5
DA 1540 RETURN
CS 1550 X=110:Y=165:COLOR4,16
HK 1560 PRINT"{CLR}{5}{RVS}
        {17 SPACES}CCCCCCCC
        {3 SPACES}{OFF}{11 U}
        {RVS}{F}{OFF}"
SD 1570 PRINT"{J}{6 SPACES}{L}
        {H}{15 SPACES}{*}{RVS}
        {3 SPACES}{OFF}
        {11 SPACES}{L}"
AD 1580 PRINT"{J}{6 SPACES}
        {WHT}{C}{V}{5}
        {16 SPACES}{*}{RVS}
        {OFF}{11 SPACES}{L}
KB 1590 PRINT"{J}{25 SPACES}
        {*}{RVS}-{OFF}
        {11 SPACES}{L}"
HK 1600 PRINT"{J}{26 SPACES}
        {RVS}-{OFF}{11 SPACES}
        {L}"
AE 1610 PRINT"{RVS}{16 SPACES}
        {OFF}{F}{22 SPACES}{L}
        "
EQ 1620 PRINT"{RVS}{17 SPACES}
        {OFF}{F}{15 SPACES}
        {RVS}-{OFF}{5 SPACES}
        {L}"
XF 1630 PRINT"{BLK}{6 E}+{6 E}
        +{2 E}{5}{RVS}
        {3 SPACES}{OFF}{F}
        {12 SPACES}{D}{RVS}
        {OFF}{5 SPACES}{L}"
HA 1640 PRINT"{J}{5 SPACES}
        {WHT}{RVS}{I}{OFF}
        {6 SPACES}{RVS}{I}
        {OFF}{2 SPACES}{5}
        {RVS}{17 SPACES}{D}
        {OFF}{5 SPACES}{L}"
MJ 1650 PRINT"{J}{15 SPACES}
        {RVS}{3 SPACES}{OFF}
        {WHT}+++{W}{9 SPACES}
        {5}{RVS}{F}{C}{OFF}
        {5 SPACES}{L}"
XA 1660 PRINT"{J}{15 SPACES}
        {WHT}{3 T}{W}{Z}
        {10 SPACES}{5}{RVS}{V}
        {OFF}{WHT}UI
        {3 SPACES}{5}{L}"
GQ 1670 PRINT"{J}{4 SPACES}
        {RED}{9 @}{5}
        {18 SPACES}{RVS} {OFF}
        {BLU}{D}{RVS} {OFF}
        {WHT}-{3 SPACES}{5}{L}
        {J}{4 SPACES}{3}{RVS}
        {9 SPACES}{OFF}{WHT}
        {O}{17 SPACES}{5}{RVS}
        {3 SPACES}{OFF}{WHT}-
        {5}{3 SPACES}{L}{J}
        {4 SPACES}{3}{RVS} {C}
        {V}{C}{V}{C}{V}{C}{V}
```

```
        {OFF}{21 SPACES}{WHT}-
        {5}{3 SPACES}{L}"
KJ 1680 PRINT"{J}{5 SPACES}
        {RVS} {OFF}{7}-{5}
        {RVS} {OFF}{5 SPACES}
        {3}{RVS} C C {OFF}
        {WHT}{U}{15 SPACES}-
        {3 SPACES}{5}{L}"
DX 1690 PRINT"{J}{5 SPACES}
        {RVS} {OFF}{7}-{5}
        {RVS} {OFF}{5 SPACES}
        {RVS}{14 SPACES}{*}
        {OFF}{6 SPACES}{WHT}-
        {3 SPACES}{5}{L}"
SK 1700 PRINT"{J}{5 SPACES}
        {RVS} {OFF}{7}-{5}
        {RVS} {OFF}{18 SPACES}
        {*}{RVS} {*}{OFF}
        {5 SPACES}{WHT}-
        {3 SPACES}{5}{L}"
SJ 1710 PRINT"{J}{5 SPACES}
        {RVS} {OFF}{7}-{5}
        {RVS} {OFF}{19 SPACES}
        {*}{RVS} {*}{OFF}
        {4 SPACES}{WHT}-
        {3 SPACES}{5}{L}{J}
        {5 SPACES}{RVS} {7}
        {5} {OFF}{20 SPACES}
        {*}{RVS} {OFF}
        {4 SPACES}{WHT}-
        {3 SPACES}{5}{L}"
KM 1720 PRINT"{J}{5 SPACES}
        {RVS} {OFF} {RVS}
        {OFF}{21 SPACES}{RVS}
        {SPACE}{OFF}{4 SPACES}
        {BLK}{U}{3 SPACES}{5}
        {L}"
BG 1730 PRINT"{J}{5 SPACES}
        {RVS}{F}{2 SPACES}{C}
        {OFF}{20 SPACES}{RVS}
        {SPACE}{OFF}{8 SPACES}
        {L}"
FK 1740 PRINT"{J}{6 SPACES}
        {RVS}{F}{2 SPACES}
        {11 P} {OFF}{8 SPACES}
        {RVS} {OFF}{8 SPACES}
        {L}"
XQ 1750 PRINT"{J}{29 SPACES}
        {RVS} {OFF}{8 SPACES}
        {L}"
CP 1760 PRINT"{J}{29 SPACES}
        {RVS} {OFF}{8 SPACES}
        {L}"
PX 1770 PRINT"{RVS} {29 U}
        {9 SPACES}{OFF}";:POKE
        2023,224:POKE56295,12
CA 1780 RETURN
AQ 1800 I=3456
JS 1810 READ A:IF A=256 THEN R
        ETURN{7 SPACES}
JB 1820 POKE I,A:I=I+1:GOTO181
        0
HP 1830 DATA000,000,000,000,00
        0,000,000,000
KK 1840 DATA000,004,000,000,00
        9,000,000,040
DX 1850 DATA000,000,106,000,00
        0,018,128,016
KE 1860 DATA000,162,020,000,04
        3,133,000,011
PB 1870 DATA225,000,010,232,00
        0,018,168,000
EA 1880 DATA020,160,000,005,12
        8,000,001,000
FS 1890 DATA000,000,000,000,00
        0,000,000,000
EX 1900 DATA000,000,000,000,00
        0,000,000,000
DS 1910 DATA000,000,000,000,00
        0,000,000,000
SB 1920 DATA000,000,000,000,00
        0,000,000,000
AA 1930 DATA000,000,000,000,08
        4,000,000,084
MC 1940 DATA064,000,016,064,01
        0,168,170,171
MS 1950 DATA232,170,171,232,06
        4,010,168,064
GP 1960 DATA000,016,000,000,08
        4,000,000,084
BF 1970 DATA000,000,000,000,00
        0,000,000,000
QE 1980 DATA000,000,000,000,00
        0,000,000,000
PH 1990 DATA000,000,000,000,00
        0,000,000,000
KF 2000 DATA000,000,000,000,00
        0,000,000,000
AM 2010 DATA000,000,000,001,00
        0,000,005,128
JB 2020 DATA000,020,160,000,01
        8,232,000,011
ED 2030 DATA232,000,011,161,00
        0,042,133,000
QD 2040 DATA162,020,018,128,01
        6,106,000,000
XB 2050 DATA040,000,000,009,00
        0,000,004,000
EK 2060 DATA000,000,000,000,00
        0,000,000,000
BJ 2070 DATA000,000,000,000,00
        0,000,000,000
CP 2080 DATA000,000,000,000,00
        0,000,000,000
QP 2090 DATA000,000,004,168,06
        4,005,169,064
XG 2100 DATA005,169,064,004,18
        4,064,000,184
FH 2110 DATA000,000,184,000,00
        0,168,000,000
CG 2120 DATA168,000,000,032,00
        0,000,032,000
XP 2130 DATA000,032,000,000,03
        2,000,001,033
XX 2140 DATA000,001,169,000,00
        1,033,000,000
RX 2150 DATA000,000,000,000,00
        0,000,000,000
SS 2160 DATA000,000,000,000,00
        0,000,000,000
DD 2170 DATA000,000,000,064,00
        0,002,080,000
XJ 2180 DATA010,020,000,043,13
        2,000,043,224
JH 2190 DATA000,074,224,000,08
        2,168,000,020
EB 2200 DATA138,000,004,002,13
        2,000,000,169
FD 2210 DATA000,000,040,000,00
        0,096,000,000
JG 2220 DATA016,000,000,000,00
        0,000,000,000
KC 2230 DATA000,000,000,000,00
        0,000,000,000
MF 2240 DATA000,000,000,000,00
        0,000,000,000
EQ 2250 DATA000,000,021,000,00
        0,021,000,000
CC 2260 DATA004,000,001,042,16
        0,001,043,234
DJ 2270 DATA170,043,234,170,04
        2,160,001,004
SA 2280 DATA000,001,021,000,00
        0,021,000,000
MK 2290 DATA000,000,000,000,00
        0,000,000,000
JG 2300 DATA000,000,000,000,00
        0,000,000,000
FG 2310 DATA000,000,000,000,00
        0,000,000,000
RC 2320 DATA000,000,000,016,00
        0,000,096,000
SD 2330 DATA000,040,000,000,16
        9,004,002,132
CJ 2340 DATA020,138,000,082,16
        8,000,075,224
HF 2350 DATA000,047,160,000,04
        2,132,000,010
XB 2360 DATA020,000,002,080,00
        0,000,064,000
GR 2370 DATA000,000,000,000,00
        0,000,000,000
BQ 2380 DATA000,000,000,000,00
        0,000,000,000
GD 2390 DATA000,000,000,001,03
        3,000,001,169
CX 2400 DATA000,001,033,000,00
        0,032,000,000
KC 2410 DATA032,000,000,032,00
        0,000,032,000
HM 2420 DATA000,032,000,000,16
        8,000,000,184
QR 2430 DATA000,000,184,000,00
        4,184,064,005
HD 2440 DATA169,064,005,169,06
        4,004,168,064
CA 2450 DATA000,000,000,000,00
        0,000,000,000
RD 2460 DATA000,000,000,000,00
        0,000,000,000
JC 2470 DATA000,000,000,000,00
        0,000,000,000
HG 2480 DATA000,000,000,000,00
        0,020,000,000
SP 2490 DATA085,000,001,085,06
        4,005,085,080
QS 2500 DATA021,085,084,085,08
        5,085,106,150
JB 2510 DATA169,101,150,089,10
        1,150,089,106
AB 2520 DATA150,169,106,150,16
        9,106,150,169
BA 2530 DATA106,150,169,106,15
        0,169,106,150
JQ 2540 DATA169,106,150,169,10
        6,150,169,000
RB 2550 DATA000,000,000,000,00
        1,000,064,007
ED 2560 DATA064,064,007,064,00
        0,049,000,004
JB 2570 DATA000,010,032,000,01
        4,000,128,000
PJ 2580 DATA002,136,192,066,04
        2,131,000,168
MS 2590 DATA000,048,043,032,00
        2,168,012,016
BM 2600 DATA032,128,000,000,04
        0,003,000,064
BH 2610 DATA048,000,000,000,01
        6,016,131,000
QX 2620 DATA116,160,000,116,00
        0,000,016,000,256
```

Program 2: Miami Ice For Commodore 64

Version by Kevin Mykytyn, Editorial Programmer

Please refer to the "MLX" article in this issue before entering the following listing.

```
0801:0C 08 0A 00 9E 20 32 30 64
0809:36 32 00 00 00 20 E0 0E 11
0811:20 BC 0D A9 00 A0 18 B9 09
0819:A2 08 99 00 D4 88 10 F7 90
0821:20 C4 08 20 75 0A AD 01 A5
0829:DC 29 10 F0 F9 A2 00 A0 FD
0831:10 18 20 F0 FF A9 E2 A0 6F
0839:0B 20 1E AB A6 B4 E8 A9 D8
0841:00 20 CD BD A9 C8 85 F8 63
0849:A9 00 85 F9 20 D6 0D 20 15
0851:52 0A 20 95 0E A2 64 88 B6
0859:D0 FD CA D0 FA AD 1F D0 55
0861:AD 1E D0 AD 01 DC 29 10 A2
0869:D0 E8 A9 40 8D 04 D4 A9 25
0871:41 8D 04 D4 20 1D 09 20 FA
0879:48 09 CE 21 13 D0 09 AD 77
```

```
0881:22 13 8D 21 13 20 95 0E 7D
0889:CE 25 13 D0 0C AD 26 13 30
0891:8D 25 13 20 16 0E 20 4F 8E
0899:0D A0 00 88 D0 FD 4C 75 6D
08A1:08 00 05 00 01 00 19 F0 81
08A9:00 1E 00 00 00 89 00 00 67
08B1:0A 00 00 00 2B 00 00 05 25
08B9:F1 4F 20 CD BD A9 20 20 6C
08C1:D2 FF 60 A9 00 85 C3 85 05
08C9:C4 A9 00 85 B4 A9 03 85 D6
08D1:BD A9 93 20 D2 FF A2 03 7E
08D9:8E 21 D0 E8 8E 20 D0 A2 5B
08E1:07 A0 0B 18 20 F0 FF A9 EE
08E9:EC A0 0B 20 1E AB A5 B4 9B
08F1:18 69 31 8D A7 05 A9 0A 16
08F9:20 45 0A AD 01 DC 4A B0 48
0901:0A A5 B4 C9 06 F0 E7 E6 5F
0909:B4 10 E3 4A B0 08 A5 B4 40
0911:F0 DC C6 B4 10 D8 4A 4A B9
0919:4A B0 D3 60 C6 F7 D0 26 DA
0921:A9 C8 85 F7 A5 F8 05 F9 7F
0929:F0 1C A5 F8 38 E9 01 85 EF
0931:F8 A5 F9 E9 00 85 F9 A2 B3
0939:18 A0 07 18 20 F0 FF A6 4D
0941:F8 A5 F9 20 BB 08 60 AD E6
0949:23 13 10 05 49 FF 18 69 E7
0951:01 85 02 AD 24 13 10 05 F2
0959:49 FF 18 69 01 18 65 02 DE
0961:8D 00 D4 AD 1F D0 29 08 46
0969:D0 14 AD 1E D0 85 02 29 4A
0971:0A C9 0A F0 09 A5 02 29 57
0979:0C C9 0C F0 3A 60 A9 40 7B
0981:8D 04 D4 A9 80 8D 12 D4 C3
0989:A9 81 8D 12 D4 A9 08 8D 8E
0991:20 13 A9 05 20 45 0A AD D5
0999:20 13 C9 0D F0 06 EE 20 28
09A1:13 4C 93 09 A9 07 8D 15 EC
09A9:D0 A9 64 20 45 0A C6 BD BA
09B1:F0 5A 68 68 4C 4D 08 A9 B7
09B9:40 8D 04 D4 A9 00 85 02 77
09C1:A5 F8 05 F9 F0 32 A5 F8 B9
09C9:38 E9 01 85 F8 A5 F9 E9 27
09D1:00 85 F9 A5 B4 85 03 E6 87
09D9:C3 D0 02 E6 C4 20 52 0A 06
09E1:E6 02 A5 02 8D 01 D4 A9 80
09E9:40 8D 04 D4 A9 41 8D 04 BE
09F1:D4 C6 03 10 E2 4C C1 09 56
09F9:A9 40 8D 04 D4 E6 B4 A9 38
0A01:64 20 45 0A A9 05 8D 01 16
0A09:D4 4C 24 08 20 52 0A A9 A7
0A11:00 8D 15 D0 A2 0C A0 05 C3
0A19:18 20 F0 FF A9 9E A0 0B 73
0A21:20 1E AB A2 0B 20 38 0A BF
0A29:A2 0D 20 38 0A AD 01 DC 3F
0A31:29 10 D0 F9 4C 21 08 A0 2F
0A39:05 18 20 F0 FF A9 C1 A0 B3
0A41:0B 4C 1E AB 85 02 A9 00 F3
0A49:85 A2 A5 A2 C5 02 D0 FA 7A
0A51:60 A9 9F 20 D2 FF A2 18 E9
0A59:A0 17 18 20 F0 FF A6 C3 21
0A61:A5 C4 20 BB 08 A2 18 A0 D4
0A69:26 18 20 F0 FF A6 BD A9 69
0A71:00 4C CD BD A9 93 20 D2 DC
0A79:FF A9 01 8D 21 D0 A9 03 93
0A81:8D 20 D0 A2 18 A0 00 18 04
0A89:20 F0 FF A9 71 A0 0B 20 C8
0A91:1E AB A0 27 A9 A0 99 00 29
0A99:04 99 98 07 A9 04 99 00 2A
0AA1:D8 99 98 DB 88 10 ED A9 63
0AA9:00 85 FB 85 FD A9 04 85 1B
0AB1:FC A9 D8 85 FE A2 18 A0 75
0AB9:00 A9 A0 91 FB A0 27 91 A7
0AC1:FB A9 04 91 FD A0 00 91 DB
0AC9:FD A5 FB 18 69 28 85 FB 3A
0AD1:A5 FC 69 00 85 FC A5 FD 8E
0AD9:18 69 28 85 FD A5 FE 69 9F
0AE1:00 85 FE CA D0 D1 A9 00 05
0AE9:85 FB A9 04 85 FC A6 B4 57
0AF1:E0 07 90 02 A2 06 BD 09 1C
0AF9:0D 0A AA BD FB 0C 85 03 66
0B01:BD FC 0C 85 04 A0 FF C8 7A
0B09:B1 03 D0 19 C8 B1 03 8D 05
0B11:02 D0 8D 04 D0 8D 00 D0 DB
0B19:C8 B1 03 8D 03 D0 8D 05 B4
0B21:D0 8D 01 D0 60 84 02 48 91
0B29:48 29 1F 8D 28 13 68 4A 13
0B31:4A 4A 4A 29 06 AA A0 00 F6
0B39:68 10 02 A0 A0 8C 29 13 6E
0B41:A5 FB 18 7D 37 0C 85 FB F4
0B49:A5 FC 7D 38 0C 85 FC A0 B5
0B51:00 AD 29 13 F0 11 91 FB 14
0B59:A5 FB 85 FD A5 FC 18 69 8C
0B61:D4 85 FE A9 04 91 FD A4 C4
0B69:02 CE 28 13 D0 D2 F0 97 B5
0B71:12 9F 54 49 4D 45 52 3A F5
0B79:20 20 20 20 20 20 20 20 8F
0B81:20 20 53 43 4F 52 45 3A D6
0B89:20 20 20 20 20 20 20 20 9F
0B91:20 20 43 41 52 53 3A 20 B0
0B99:20 9D 94 20 00 12 9F 20 63
0BA1:92 47 41 4D 45 20 4F 56 6F
0BA9:45 52 20 2D 20 50 52 45 F9
0BB1:53 53 20 46 49 52 45 42 0F
0BB9:55 54 54 4F 4E 12 20 00 0A
0BC1:12 9F 20 20 20 20 20 20 B0
0BC9:20 20 20 20 20 20 20 20 DF
0BD1:20 20 20 20 20 20 20 20 E7
0BD9:20 20 20 20 20 20 20 20 EF
0BE1:00 9C 12 53 43 52 45 45 C9
0BE9:4E 20 00 92 9C 4D 20 49 FB
0BF1:20 41 20 4D 20 49 20 20 C7
0BF9:20 49 20 43 20 45 11 11 F3
0C01:11 9D 9D 9D 9D 9D 9D 9D D2
0C09:9D 9D 9D 9D 9D 1F 53 43 38
0C11:52 45 45 4E 20 31 11 11 2A
0C19:11 9D 9D 9D 9D 9D 9D 9D EA
0C21:9D 9D 9D 9D 9D 50 52 45 15
0C29:53 53 20 46 49 52 45 42 88
0C31:55 54 54 4F 4E 00 D8 FF AC
0C39:FF FF 28 00 01 00 4B FF F4
0C41:86 CD 00 37 B4 6A D1 A3 1A
0C49:E6 66 C6 90 A3 E6 65 86 5A
0C51:C5 E3 CC A3 E6 00 FB 42 8A
0C59:45 EC 47 25 EE 8B 6A 43 48
0C61:CD AA C2 02 B4 45 65 00 09
0C69:2D C8 68 C3 45 A7 EB C3 F7
0C71:83 E4 83 C3 E6 C3 83 E2 61
0C79:A1 04 83 47 E6 83 C3 E3 F8
0C81:C3 45 E5 AB 83 C6 83 AA 2D
0C89:83 C6 83 AA 83 C6 45 65 57
0C91:83 00 FA BC 66 C6 A1 E3 0C
0C99:A2 C8 E1 27 E2 A1 65 C4 11
0CA1:45 24 E1 63 05 EA C2 82 B3
0CA9:EA 81 C1 E5 46 81 05 E3 54
0CB1:06 E4 A1 C2 82 AD 81 C1 F5
0CB9:A6 C2 82 A4 84 E1 A1 82 E1
0CC1:A1 E1 81 68 05 C3 A1 E1 36
0CC9:C3 EA C1 81 E2 06 C1 00 81
0CD1:DC 46 65 D3 E9 64 E2 81 FB
0CD9:C1 EE 8E A2 24 AA 24 A4 43
0CE1:C9 E8 64 EC 00 73 89 46 9B
0CE9:F2 82 E1 C2 EC C5 E2 C1 8A
0CF1:A2 C4 AC C2 A1 82 B1 00 C8
0CF9:2D C8 3F 0C 46 0C 59 0C A4
0D01:6B 0C 95 0C D3 0C E8 0C F3
0D09:00 06 01 05 02 04 03 CE 0A
0D11:27 13 D0 39 A9 07 8D 27 DC
0D19:13 CE 2A 13 D0 0F A9 4B 48
0D21:8D 2A 13 AD 26 13 C9 07 E1
0D29:F0 03 EE 26 13 AD 23 13 65
0D31:F0 0B 10 06 EE 23 13 4C 5F
0D39:3E 0D CE 23 13 AD 24 13 6C
0D41:F0 0B 10 06 EE 24 13 4C 73
0D49:4E 0D CE 24 13 60 AE 20 81
0D51:13 AD 23 13 18 7D 77 0D A8
0D59:C9 46 90 04 C9 B9 90 03 95
0D61:8D 23 13 AD 24 13 18 7D 63
0D69:7F 0D C9 46 90 04 C9 B9 06
0D71:90 03 8D 24 13 60 01 00 A4
0D79:FF FF FF 00 01 01 FF FF 9F
0D81:FF 00 01 01 01 00 A9 01 28
0D89:8D 19 D0 AD 1B 13 8D 06 EB
0D91:D0 AD 1E 13 8D 07 D0 AD 4C
0D99:1C 13 0A 0A 0A 8D 10 D0 DF
0DA1:AD 20 13 18 69 D4 8D FB 34
0DA9:07 A9 FA 8D 12 D0 AD 0D 26
0DB1:DC 29 01 F0 03 4C 31 EA 4A
0DB9:4C BC FE A9 1B 8D 11 D0 A5
0DC1:A9 7F 8D 0D DC A9 87 8D 3D
0DC9:14 03 A9 0D 8D 15 03 A9 25
0DD1:81 8D 1A D0 60 A9 00 8D 97
0DD9:1C 13 8D 23 13 8D 24 13 D4
0DE1:A9 26 8D 1B 13 A9 3C 8D 03
0DE9:1E 13 A9 07 8D 20 13 A5 36
0DF1:A2 C5 A2 F0 FC A9 0F 8D 6C
0DF9:15 D0 A9 64 8D 21 13 8D F2
0E01:22 13 A9 07 8D 25 13 8D 4D
0E09:26 13 A9 07 8D 27 13 A9 7B
0E11:37 8D 2A 13 60 20 10 0D 53
0E19:AD 23 13 30 1A 18 6D 1A 60
0E21:13 8D 1A 13 AD 1B 13 69 08
0E29:00 8D 1B 13 AD 1C 13 69 AA
0E31:00 8D 1C 13 4C 57 0E 49 8A
0E39:FF 69 01 85 02 AD 1A 13 36
0E41:38 E5 02 8D 1A 13 AD 1B 9F
0E49:13 E9 00 8D 1B 13 AD 1C DE
0E51:13 E9 00 8D 1C 13 AD 24 F6
0E59:13 30 18 18 6D 1D 13 8D 23
0E61:1D 13 AD 1E 13 69 00 8D 34
0E69:1E 13 AD 1F 13 69 00 8D CC
0E71:1F 13 60 49 FF 18 69 01 B6
0E79:85 02 AD 1D 13 38 E5 02 A7
0E81:8D 1D 13 AD 1E 13 E9 00 F9
0E89:8D 1E 13 AD 1F 13 E9 00 4A
0E91:8D 1F 13 60 AD 01 DC 4A 1A
0E99:4A 4A B0 12 20 D3 0E EE FF
0EA1:20 13 AE 20 13 E0 08 D0 67
0EA9:05 A2 00 8E 20 13 4A B0 6C
0EB1:0D 20 D3 0E CE 20 13 10 E4
0EB9:05 A2 07 8E 20 13 4A B0 5D
0EC1:10 AD 26 13 C9 02 F0 09 88
0EC9:CE 26 13 AD 26 13 8D 25 D1
0ED1:13 60 48 A9 80 8D 0B D4 58
0ED9:A9 81 8D 0B D4 68 60 A0 37
0EE1:00 B9 5A 0F 99 00 35 B9 99
0EE9:5A 10 99 00 36 B9 5A 11 C8
0EF1:99 00 37 B9 5A 12 99 00 AB
0EF9:38 A9 00 99 00 39 88 D0 FC
0F01:E0 A0 3F B9 DA 12 99 80 0E
0F09:39 88 10 F7 A0 02 A9 FF C7
0F11:99 00 39 99 3C 39 88 10 A4
0F19:F7 A0 36 A9 80 99 03 39 66
0F21:88 88 88 10 F8 A9 0C 8D CB
0F29:5C 39 8D 62 39 A2 E4 8E 48
0F31:F9 07 E8 8E FA 07 E8 8E 68
0F39:F8 07 A0 03 B9 56 0F 99 B8
0F41:27 D0 88 10 F7 A9 08 8D 3D
0F49:1C D0 A9 00 8D 25 D0 A9 2B
0F51:07 8D 26 D0 60 02 00 02 35
0F59:02 00 00 00 00 00 00 00 78
0F61:00 00 00 00 10 00 00 60 60
0F69:00 00 28 00 00 A9 04 02 3D
0F71:84 14 8A 00 52 A8 00 4B A8
0F79:E0 00 2F A0 00 2A 84 00 A9
0F81:0A 14 00 02 50 00 00 40 8C
0F89:00 00 00 00 00 00 00 00 A7
0F91:00 00 00 00 00 00 00 00 AF
0F99:00 00 00 00 01 21 00 01 45
0FA1:A9 00 01 21 00 00 20 00 07
0FA9:00 20 00 00 20 00 00 20 F0
0FB1:00 00 20 00 00 A8 00 00 76
0FB9:B8 00 00 B8 00 04 B8 40 81
0FC1:05 A9 40 05 A9 40 04 A8 24
0FC9:40 00 00 00 00 00 00 00 08
0FD1:00 00 00 00 00 00 00 00 EF
0FD9:00 00 00 00 00 00 00 00 F7
0FE1:00 00 04 00 00 09 00 00 A4
0FE9:28 00 00 6A 00 00 12 80 67
0FF1:10 00 A2 14 00 2B 85 00 65
0FF9:0B E1 00 0A E8 00 12 A8 CA
1001:00 14 A0 00 05 80 00 01 65
1009:00 00 00 00 00 00 00 00 29
1011:00 00 00 00 00 00 00 00 31
1019:00 00 00 00 00 00 00 00 39
1021:00 00 00 00 00 00 00 00 41
1029:00 00 00 00 00 54 00 00 9A
1031:54 40 00 10 40 0A A8 AA B2
1039:AB E8 AA AB E8 40 0A A8 7E
1041:40 00 10 00 00 54 00 00 D4
1049:54 00 00 00 00 00 00 00 93
1051:00 00 00 00 00 00 00 00 71
1059:00 00 00 00 00 00 00 00 79
1061:00 00 00 00 00 00 00 00 81
1069:00 00 00 00 01 00 00 05 96
1071:80 00 14 A0 00 12 E8 00 78
```

```
1079:0B E8 00 0B A1 00 2A 85 F0
1081:00 A2 14 12 80 10 6A 00 07
1089:00 28 00 00 09 00 00 04 FF
1091:00 00 00 00 00 00 00 00 B1
1099:00 00 00 00 00 00 00 00 B9
10A1:00 00 00 00 00 00 00 00 C1
10A9:00 00 00 04 A8 40 05 A9 04
10B1:40 05 A9 40 04 B8 40 00 EF
10B9:B8 00 00 B8 00 00 A8 00 13
10C1:00 A8 00 00 20 00 00 20 2D
10C9:00 00 20 00 00 20 00 01 6F
10D1:21 00 01 A9 00 01 21 00 83
10D9:00 00 00 00 00 00 00 00 F9
10E1:00 00 00 00 00 00 00 00 02
10E9:00 00 00 00 40 00 02 50 60
10F1:00 0A 14 00 2B 84 00 2B AD
10F9:E0 00 4A E0 00 52 A8 00 7C
1101:14 8A 00 04 02 84 00 00 32
1109:A9 00 00 28 00 00 60 00 43
1111:00 10 00 00 00 00 00 00 37
1119:00 00 00 00 00 00 00 00 3B
1121:00 00 00 00 00 00 00 00 43
1129:00 00 00 15 00 00 15 00 C6
1131:00 04 00 01 2A A0 01 2B 65
1139:EA AA 2B EA AA 2A A0 01 CF
1141:04 00 01 15 00 00 15 00 01
1149:00 00 00 00 00 00 00 00 6B
1151:00 00 00 00 00 00 00 00 73
1159:00 00 00 00 00 00 0C 00 7B
1161:01 00 00 07 40 10 07 40 05
1169:10 01 00 01 30 28 08 80 96
1171:38 02 80 C0 04 A2 03 00 FC
1179:98 01 0C AB 01 02 A8 80 46
1181:04 22 B0 03 00 60 0C 10 1E
1189:00 28 C0 40 20 01 D0 C0 39
1191:01 D0 00 00 40 00 00 00 6A
1199:00 00 00 00 00 01 00 40 FF
11A1:07 40 40 07 40 00 31 00 34
11A9:04 00 0A 20 00 0E 00 80 C9
11B1:00 02 88 C0 42 2A 83 00 33
11B9:A8 00 30 2B 20 02 A8 0C 4F
11C1:10 20 80 00 00 28 03 00 AA
11C9:40 30 00 00 00 10 10 83 FB
11D1:00 74 A0 00 74 00 00 10 D8
11D9:00 00 07 40 40 01 00 00 E6
11E1:00 00 04 30 00 00 00 00 87
11E9:80 00 00 02 80 00 00 8A FA
11F1:80 00 2A 0C 42 08 03 02 94
11F9:8B 00 C0 20 E0 CA 28 03 81
1201:D0 A8 00 02 00 28 30 20 F8
1209:02 0C 00 40 C0 00 00 00 3B
1211:10 01 03 00 07 80 00 07 1F
1219:00 00 01 00 10 30 00 00 9E
1221:00 00 00 00 00 02 88 00 5E
1229:02 20 00 00 32 A0 00 0A 74
1231:03 00 00 80 00 82 00 02 EA
1239:A8 00 20 08 E0 3A 00 00 26
1241:28 02 08 40 02 02 00 20 37
1249:00 00 00 00 30 00 10 00 0F
1251:00 00 00 10 00 03 00 01 83
1259:00 40 30 00 00 00 00 08 9B
1261:00 00 0A 00 00 00 08 00 D6
1269:00 30 2A 00 00 A0 00 00 61
1271:00 0A 00 20 02 00 80 00 2B
1279:08 A0 20 00 B8 CC 00 00 C6
1281:28 00 00 20 00 02 00 02 C5
1289:00 00 22 00 00 00 00 C0 B2
1291:00 00 00 00 01 00 10 00 DD
1299:00 08 30 00 28 00 00 00 07
12A1:00 00 00 02 00 00 0C 00 FD
12A9:00 00 0A A0 00 00 80 00 1A
12B1:00 00 00 02 00 00 08 00 06
12B9:00 0A 80 00 0B 30 00 20 A9
12C1:00 00 00 00 00 00 00 00 E5
12C9:00 00 00 00 00 00 00 00 ED
12D1:00 00 08 02 01 00 12 00 43
12D9:00 00 00 00 3F FF FC 5F 51
12E1:FF FA 5F FF FA 6F FF F6 3D
12E9:6F FF F6 77 FF EE 77 FF C6
12F1:EE 7B FF DE 7B F3 DE 7C 40
12F9:00 3E 7B F3 DE 7B FF DE 20
1301:77 FF EE 77 FF EE 6F FF D2
1309:F6 6F FF F6 5F FF FA 5F 46
1311:FF FA 3F FF FC 00 00 00 C5
1319:FF 00 00 00 00 00 00 00 3F
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Euchre

Article on page 16.

```
SK 10 GOSUB1100:GOSUB1000:GOSU
      B2700:GOSUB1300:GOSUB100
      00
SG 25 GOSUB2000
MS 30 IFTP<>4THEN70
QJ 35 POKEX,12:POKEY,30:SYSPLT
      :PRINT"{BLK}NO{2 SPACES}
      BIDS";
XQ 40 POKEX,13:POKEY,28:SYSPLT
      :PRINT"HAND{2 SPACES}DUM
      PED";
XG 50 GOSUB1400:FORI=0TO4:POKE
      X,19:POKEY,I*5+2:SYSPLT:
      PRINTEC$;:NEXT
XP 51 DL=FNNP(DL):POKEX,7:POKE
      Y,12:SYSPLT:PRINTEC$;
 DH 55 GOSUB1900:GOTO25
FR 70 POKEX,12:POKEY,30:SYSPLT
      :PRINT"{BLK}TRUMP: "CO$(
      TP)S$(TP);
JJ 75 POKEX,13:POKEY,29:SYSPLT
      :PRINT"{BLK}BIDDER:";:IF
      BD=0THENPRINT"YOU";:GOTO
      90
EF 80 PRINT" P"RIGHT$(STR$(BD)
      ,1);
ES 90 PRINT"{BLU}";:IFKU=0THEN
      135
AX 100 IFDL>0THEN125
KM 115 POKEX,15:POKEY,28:SYSPL
      T:PRINT"{BLK}PICK DISCA
      RD";
FD 121 GOSUB2200:POKEX,15:POKE
      Y,28:SYSPLT:PRINT"{BLU}
      {12 SPACES}";:GOTO130
XM 125 GOSUB2300
JK 130 C(DL,F)=KC:S(DL,F)=KS:G
      OSUB1220
QJ 135 POKEX,7:POKEY,12:SYSPLT
      :PRINTEC$;
MK 140 GOSUB2500:GOSUB3000:IF(
      PW(0)>9)OR(PW(1)>9)THEN
      300
GS 150 POKEX,21:POKEY,31:SYSPL
      T:N=0:GOSUB1465
MP 151 POKEX,21:POKEY,35:SYSPL
      T:N=0:GOSUB1465
AG 200 POKEX,7:POKEY,31:SYSPLT
      :N=PW(0):GOSUB1465
BS 205 POKEX,7:POKEY,35:SYSPLT
      :N=PW(1):GOSUB1465
CD 210 DL=FNNP(DL):GOSUB1355:G
      OTO25
MP 300 WT=0:IFPW(1)>=10THENWT=
      1
PE 305 POKEX,7:POKEY,29+WT*6:S
      YSPLT:N=1:GOSUB1465
EQ 310 POKEX,7:POKEY,31+WT*6:S
      YSPLT:N=PW(WT)-10:GOSUB
      1465
KM 315 FORDE=1TO2000:NEXT
EM 320 GOSUB1450:POKEX,13:POKE
      Y,30:SYSPLT:PRINT"{RED}
      YOU";
DE 321 IFWT=0THENPRINT" WIN!";
PQ 322 IFWT=1THENPRINT" LOSE";
PJ 325 POKEX,15:POKEY,11:SYSPL
      T:PRINT"{RVS}{BLU}PLAY
      {SPACE}AGAIN?{OFF}";
BA 330 LO=11:HI=12:XP=15:YP=23
      :GOSUB1910
RG 340 IFAN=11THENRUN
AB 350 PRINT"{CLR}";
GG 999 END
JH 1000 POKE53281,15:POKE53280
      ,6:REM SETSCR
AR 1005 PRINT"{CLR}{BLU}";
CE 1010 FORI=1TO24:PRINT"{RVS}
      {28 SPACES}{OFF}
      {12 SPACES}";:NEXT
FB 1015 PRINT"{RVS}{28 SPACES}
      {OFF}{HOME}";
AC 1020 PRINTSPC(28)"{BLU}{A}C
      CCCCCCCCC{S}";
SG 1025 PRINTSPC(28)"B{RED}
      {2 SPACES}EUCHRE
      {2 SPACES}{BLU}B";
      {32 SPACES}
DD 1030 PRINTSPC(28)"{Z}CCCCCC
      CCCC{X}";
XF 1031 PRINTSPC(28)"{RVS}
      {12 SPACES}";
XB 1032 PRINTSPC(28)"
      {3 SPACES}POINTS
      {3 SPACES}{OFF}";
CD 1035 PRINTSPC(28)"{RED} YOU
       {BLU}{M}{G}{RED}COMP
      {SPACE}";
SB 1040 PRINTSPC(28)"{BLU}
      {5 Y}PO{5 Y}";
GR 1045 FORI=1TO4:PRINTSPC(28)
      "{5 SPACES}{M}{G}
      {5 SPACES}";:NEXT
KQ 1050 PRINTSPC(28)"{12 Y}
      {5 DOWN}";
AJ 1051 PRINTSPC(28)"{12 @}";
CP 1055 PRINTSPC(28)"{RVS}
      {3 SPACES}TRICKS
      {3 SPACES}{OFF}";
AC 1060 PRINTSPC(28)"{RED} YOU
       {BLU}{M}{G}{RED}COMP
      {SPACE}";
KE 1065 PRINTSPC(28)"{BLU}
      {5 Y}PO{5 Y}";
HR 1070 FORI=1TO3:PRINTSPC(28)
      "{5 SPACES}{M}{G}
      {5 SPACES}";:NEXT
DM 1075 PRINTSPC(28)"
      {5 SPACES}{M}{G}
      {4 SPACES}";
ER 1080 POKE2023,32:POKE56295,
      6
XS 1081 POKEX,7:POKEY,31:SYSPL
      T:GOSUB1465:POKEX,7:PO
      KEY,35:SYSPLT:GOSUB146
      5
SB 1082 POKEX,21:POKEY,31:SYSP
      LT:GOSUB1465:POKEX,21:
      POKEY,35:SYSPLT:GOSUB1
      465
QE 1090 RETURN
SS 1100 DIM C1$(7),C2$(7),S$(3
      ),DC(23),DS(23),C(3,4)
      ,S(3,4),CO$(3),CX(3),C
      Y(3)
MS 1101 DIM OB(6),OU(6),PU(6),
      MS(6,3),GA(6),ME$(15),
      PX(3),PY(3),NM$(9,2),N
      F(6)
SS 1102 DIM CL(7,3)
QS 1108 X=RND(-TI)
GQ 1109 FORI=0TO3:READS$(I),CO
      $(I):NEXT
EF 1110 DATA"Z","{RED}","X","
      {BLK}","S","{RED}","A"
      ,"{BLK}"
PE 1111 FORI=0TO6:READNF(I):NE
      XT:DATA4,0,1,2,3,4,0
JH 1115 FORI=0TO7:READC1$(I),C
      2$(I):NEXT
```

```
GG 1120 DATA"9 "," 9","1O","1O
        ","J "," J","Q "," Q",
        "K "," K","A "," A"
XJ 1125 DATA"J "," J","J "," J
        "
JS 1130 FORI=251TO254:READQ:PO
        KEI,Q:NEXT
RK 1135 DATA24,76,240,255
EM 1140 PLT=251:X=781:Y=782:SX
        =53248:BL$="
        {12 SPACES}"
GQ 1145 EC$="{4 SPACES}{DOWN}
        {4 LEFT}":EC$="{RVS}"+
        EC$+EC$+EC$+EC$+EC$+"
        {OFF}"
KS 1150 FORJ=ØTO3:FORI=ØTO5
AP 1155 DC(J*6+I)=I:DS(J*6+I)=
        J
RF 1160 NEXTI:NEXTJ
EK 1161 FORI=ØTO3:READPX(I),PY
        (I):NEXT:DATA13,12,7,7
        ,1,12,7,17
PD 1165 FORI=ØTO3:READCX(I),CY
        (I):NEXT
CF 1170 DATA17,11,10,4,3,11,10
        ,21
SQ 1171 FORI=ØTO2:FORJ=ØTO9:RE
        ADNM$(J,I):NEXT:NEXT
BR 1172 DATA"{RVS} {F3}","{RVS}
        {K3}{OFF}{K3}","{RVS}{I3
        ","{RVS}{I3 ","{2 K3}
        ","{RVS} {I3}","{RVS}
        {I3}","{RVS}{I3 ","
        {RVS} {F3}","{RVS}{D3} "
BX 1173 DATA" {K3}","{RVS}{K3}
        {OFF}{K3}"," {I3}","{I3
        {SPACE}","{RVS}{2 C3}",
        "{I3 "," {F3}","{OFF}
        {RVS} "," {F3}","{C3} "
DH 1174 DATA" {RVS}{V3}","{RVS}
        {K3}{OFF}{K3}"," {OFF}
        {I3}","{OFF}{I3{RVS} ",
        "{OFF} {K3}","{OFF}{I3
        {RVS} "," {V3}","{OFF}
        {SPACE}{RVS} "," {V3}",
        "{OFF}{I3{RVS} "
EA 1175 DEF FNNP(X)=((X+1)/4-I
        NT((X+1)/4))*4
HH 1176 FORI=ØTO5:READCP(I):NE
        XT:DATA1,1,8,1,2,-1
QA 1177 FORI=ØTO13:README$(I):
        NEXT
XX 1178 DATA"PASS{4 SPACES}","
        ORDER UP","PASS
        {3 SPACES}","PICK UP",
        "{2 SPACES}PASS
        {4 SPACES}"
EM 1179 DATA"Z DIAMONDS","X CL
        UBS{3 SPACES}","S HEAR
        TS{2 SPACES}","A SPADE
        S{2 SPACES}","NORMAL
        {4 SPACES}"
HH 1180 DATA"AGGRESSIVE","YES"
        ,"NO ","YES"
JD 1185 FORI=1TO3:READMX(I),MY
        (I):NEXT:DATA7,2,1,10,
        7,19
BQ 1186 FORI=ØTO6:READOB(I),OU
        (I),PU(I),MS(I,Ø),MS(I
        ,1),MS(I,2),MS(I,3),GA
        (I):NEXT
FF 1187 DATA99,99,99,99,99,99,
        99,99
AF 1188 DATA99,99,99,99,99,99,
        99,99
JX 1189 DATA99,99,14,14,14,13,
        13,99
FQ 1190 DATA20,12,08,08,08,08,
        07,19
CK 1191 DATA14,00,00,00,00,00,
        00,16
ES 1192 DATA00,00,00,00,00,00,
        00,14
AK 1193 DATA00,00,00,00,00,00,
        00,00
XX 1199 RETURN
ER 1200 IF(S=TP)AND(C=6)THENS=
        S+2:S=(S/4-INT(S/4))*4
CH 1205 PRINTCO$(S)C1$(C)"
        {2 SPACES}{DOWN}
        {4 LEFT}"S$(S)"
        {3 SPACES}{DOWN}
        {4 LEFT}{4 SPACES}
        {DOWN}{4 LEFT}
        {3 SPACES}"S$(S)"
        {DOWN}{4 LEFT}
        {2 SPACES}"C2$(C);
XQ 1207 PRINT"{BLU}";
JM 1210 RETURN
AK 1220 FORU=ØTO4:POKEX,19:POK
        EY,U*5+2:SYSPLT
AB 1225 C=C(Ø,U):S=S(Ø,U):GOSU
        B1200:NEXTU
CQ 1230 RETURN
XH 1250 FORI=ØTO23:J=INT(RND(1
        )*24)
HA 1255 T=DC(I):DC(I)=DC(J):DC
        (J)=T
FH 160 T=DS(I):DS(I)=DS(J):DS(
       J)=T
HC 1265 NEXT
DE 1270 FORJ=ØTO3:FORI=ØTO4
AG 1275 C(J,I)=DC(J*5+I):S(J,I
        )=DS(J*5+I):NEXT:NEXT:
        KC=DC(20):KS=DS(20)
CA 1280 RETURN
SD 1300 GOSUB1450:POKEX,12:POK
        EY,28:SYSPLT
DF 1305 PRINT"{BLK}FIRST
        {2 SPACES}BLACK";
EX 1310 POKEX,13:POKEY,29:SYSP
        LT
GB 1315 PRINT"JACK DEALS";
XD 1320 GOSUB1250:DL=Ø:CC=Ø
SD 1325 POKEX,CX(DL):POKEY,CY(
        DL):SYSPLT
EP 1330 C=DC(CC):S=DS(CC):GOSU
        B1200
PQ 1331 FORDE=1TO500:NEXT
MH 1335 IF(DC(CC)=2)AND((DS(CC
        )AND253)=1)THENGOTO135
        5
JC 1336 POKEX,CX(DL):POKEY,CY(
        DL):SYSPLT
RC 1337 FORDE=1TO100:NEXT
MS 1340 PRINTEC$;:CC=CC+1:DL=F
        NNP(DL):GOTO1325
JK 1355 GOSUB1450:PRINT"{BLK}"
        ;:POKEX,12:POKEY,29:SY
        SPLT
RM 1356 IFDL=ØTHEN1370
HH 1360 PRINT" PLAYER"DL;
RE 1362 POKEX,13:POKEY,31:SYSP
        LT
AG 1365 PRINT"DEALS";:GOTO1375
KX 1370 PRINT"YOUR{2 SPACES}DE
        AL";
PR 1375 GOSUB1400
EB 1376 POKEX,CX(DL):POKEY,CY(
        DL):SYSPLT:PRINTEC$;
JG 1380 RETURN
JM 1400 POKEX,15:POKEY,28:SYSP
        LT
EE 1405 GOSUB6000:PRINT"{BLK}
        {RVS} HIT BUTTON {OFF}
        ";:POKE198,Ø
RR 1410 GETWT$:IF(WT$<>CHR$(13
        ))AND(PEEK(56320)<>111
        )THEN1410
DA 1415 PRINT"{BLU}";:GOSUB145
        Ø
HP 1430 RETURN
EM 1450 FORR=12TO16:POKEX,R:PO
        KEY,28:SYSPLT
JB 1455 PRINT"{12 SPACES}";:NE
        XT
QR 1460 RETURN
AS 1465 PRINT"{RED}";:FORI=ØTO
        2:PRINTNM$(N,I)"{DOWN}
        {2 LEFT}";:NEXT:PRINT"
        {OFF}{BLU}";
DQ 1470 RETURN
KE 1500 FORI=ØTO3:SP(P,I)=FC(P
        AND253):NS(P,I)=Ø:NEXT
RM 1505 FORI=ØTO4:S=S(P,I):C=C
        (P,I):SP(P,S)=SP(P,S)+
        CP(C):NS(P,S)=NS(P,S)+
        1
JH 1510 IFC=2THENS=S+2:S=(S/4-
        INT(S/4))*4:SP(P,S)=SP
        (P,S)+6:NS(P,S)=NS(P,S
        )+1
XA 1515 IFC=5THENFORJ=ØTO3:SP(
        P,J)=SP(P,J)+4:NEXTJ
QJ 1520 NEXTI
SE 1525 SS=Ø:FORI=ØTO4:IFNS(P,
        I)=ØTHENSP(P,I)=Ø:SS=S
        S+1
BJ 1530 NEXT:FORI=ØTO3:SP(P,I)
        =SP(P,I)+SS:NEXT
EC 1535 IFP<>DLTHEN1550
PX 1540 IFKC=5THENSP(P,KS)=SP(
        P,KS)+4
PP 1545 SP(P,KS)=SP(P,KS)+CP(K
        C):NS(P,KS)=NS(P,KS)+1
ED 1550 RETURN
BM 1600 LO=Ø:HI=1:XP=13:YP=10:
        GOSUB1910
RX 1605 IF AN=1THENTP=KS
CE 1610 RETURN
AB 1615 POKEX,13:POKEY,10:SYSP
        LT:PRINT"{RVS}LONEHAND
        ?{OFF}";
PJ 1616 LO=12:HI=13:XP=13:YP=2
        Ø:GOSUB1910:LH=Ø
SA 1618 IFAN=13THENLH=1:POKEX,
        MX(BD)+2:POKEY,MY(BD):
        SYSPLT:PRINT"{RVS}LONE
        HAND{OFF}";
CB 1619 POKEX,13:POKEY,10:SYSP
        LT:PRINT"{RVS}
        {9 SPACES}{OFF}";
HH 1620 RETURN
SP 1625 LH=Ø:IFSP(P,TP)>GA(NS(
        P,TP))THENLH=1
JH 1630 RETURN
EX 1650 IFFNNP(FNNP(P))=DLTHEN
        GOSUB1625:F=LH:GOTO168
        5
GE 1654 F=Ø:IFKC=2THENGOTO1660
HF 1655 IFSP(P,KS)>OU(NS(P,KS)
        )THENF=1
HS 1660 IFSP(P,KS)>OB(NS(P,KS)
        )THENF=1
AP 1665 IF(F=Ø)OR(P<>FNNP(DL))
        THEN1685
XJ 1670 SB=CP(KC):IFKC=5THENSB
        =3
CX 1675 FORI=ØTO3:IFI<>KSTHENI
        FSP(P,I)>=(SP(P,KS)-SB
        )THENF=Ø
BR 1680 NEXT
DX 1685 IFF=1THENTP=KS
XX 1699 RETURN
SA 1700 LO=2:HI=3:XP=13:YP=11:
        GOSUB1910
QH 1705 IF AN=3THENTP=KS
QM 1710 RETURN
GG 1750 IFSP(P,KS)>PU(NS(P,KS)
        )THENTP=KS
AB 1755 RETURN
KE 1800 LO=4:HI=8:XP=13:YP=9:G
        OSUB1910
AS 1801 IFAN-5=KSTHEN1800
AQ 1805 IFAN>4THENTP=AN-5
HD 1810 RETURN
```

```
KA 1850 DF=0:FORI=0TO3:IFI=KST
        HEN1865
GQ 1855 IFSP(P,I)-MS(NS(P,I),P
        S)<DFTHEN1865
CS 1860 DF=SP(P,I)-MS(NS(P,I),
        PS):TP=I
RK 1865 NEXT
KH 1870 RETURN
QH 1900 FORI=1TO3:FORJ=0TO2:PO
        KEX,MX(I)+J:POKEY,MY(I
        ):SYSPLT:PRINT"{RVS}
        {8 SPACES}";
PC 1905 NEXT:NEXT:PRINT"{OFF}"
        ;
PJ 1907 RETURN
ED 1910 POKE198,0:AN=LO
BD 1915 X1=XP:Y1=YP:FORI=LOTOH
        I:PRINT"{RVS}{BLU}";
BJ 1920 IFI<>ANTHEN1925
JC 1921 IF(AN=LO)OR(HI-LO=1)TH
        ENPRINT"{RED}";:GOTO19
        25
JM 1922 PRINTCO$(I-LO-1);
HJ 1925 POKEX,X1:POKEY,Y1:SYSP
        LT:PRINTME$(I);:X1=X1+
        1:NEXT
BH 1930 GETR$:DR=PEEK(56320):I
        FNOT((R$="{UP}")OR(DR=
        126))THENGOTO1940
RH 1935 AN=AN-1:IFAN<LOTHENAN=
        HI
PJ 1940 IFNOT((R$="{DOWN}")OR(
        DR=125))THENGOTO1950
GX 1945 AN=AN+1:IFAN>HITHENAN=
        LO
SB 1950 IFNOT((R$=CHR$(13))OR(
        DR=111))THENGOTO1915
SG 1955 X1=XP:Y1=YP:PRINT"
        {RVS}{BLU}";:FORI=LOTO
        HI
HG 1960 POKEX,X1:POKEY,Y1:SYSP
        LT:PRINTLEFT$(BL$,LEN(
        ME$(LO)));:X1=X1+1:NEX
        T
JK 1965 PRINT"{OFF}";:RETURN
FQ 2000 GOSUB1250:GOSUB1220:P=
        FNNP(DL):TP=4:BD=0:KU=
        0
MK 2005 POKEX,7:POKEY,12:SYSPL
        T:C=KC:S=KS:GOSUB1200
HX 2007 IFDL<>0THENPOKEX,MX(DL
        ):POKEY,MY(DL)+1:SYSPL
        T:PRINT"{RVS}DEALER"
BA 2010 GOSUB1500
XS 2015 IFP=0THENGOSUB1600:GOT
        O2040
BH 2016 IFABS(P-DL)<>2THEN2020
RS 2017 GOSUB1625:IFLH=1THENTP
        =KS:GOTO2025
RG 2020 GOSUB1650
XA 2025 POKEX,MX(P):POKEY,MY(P
        ):SYSPLT:PRINT"{RVS}";
RH 2030 IFTP=4THENPRINT"
        {2 SPACES}PASS":GOTO20
        40
FD 2035 BD=P:PRINT"ORDER UP";
AF 2040 P=FNNP(P):IF(P<>DL)AND
        (TP=4)THEN2010
KM 2045 P=DL:GOSUB1500:IFTP<>4
        THEN2105
RM 2050 IFDL=0THENGOSUB1700:GO
        TO2070
PS 2055 GOSUB1750:POKEX,MX(DL)
        :POKEY,MY(DL):SYSPLT:P
        RINT"{RVS}{BLU}";
CQ 2060 IFTP=4THENPRINT" TURNE
        D{DOWN}{5 LEFT}DOWN";:
        GOTO2070
AJ 2065 BD=P:PRINT" PICKED
        {DOWN}{4 LEFT}UP";
HM 2070 FORDE=1TO2000:NEXT
HR 2071 IF(BD=0)AND(TP<>4)THEN
        2105
QE 2072 POKEX,7:POKEY,12:SYSPL
        T:PRINTEC$;:IFTP<>4THE
        N2105
QC 2073 GOSUB1900:PS=0
HX 2075 P=FNNP(P)
JS 2080 IFP=0THENGOSUB1800:GOT
        O2090
PK 2085 GOSUB1850:POKEX,MX(P):
        POKEY,MY(P):SYSPLT:PRI
        NT"{RVS}";
RS 2086 FORDE=1TO600:NEXT
FC 2088 IFTP=4THENPRINT"
        {2 SPACES}PASS";:GOTO2
        090
RS 2089 BD=P:PRINTRIGHT$(ME$(T
        P+5),8);
HD 2090 IF(P<>DL)AND(TP=4)THEN
        PS=PS+1:GOTO2075
PK 2100 GOTO2109
KK 2105 KU=1:IF(BD=0)AND(DL=2)
        THENLH=1:GOTO2120
DR 2109 IFTP=4THEN2140
RA 2110 IF(LH=1)AND(BD<>0)THEN
        2120
DP 2111 IFBD=0THENGOSUB1615:GO
        TO2140
RS 2112 GOSUB1625
MA 2115 IFLH=0THEN2140
EX 2120 POKEX,MX(BD)+2:POKEY,M
        Y(BD):SYSPLT:PRINT"
        {RVS}LONEHAND";
CF 2140 PRINT"{OFF}";:FORDE=1T
        O2000:NEXT
DX 2145 GOSUB1900
PG 2150 RETURN
DF 2200 POKE198,0:F=-1
GE 2205 F=F+1:IFC(0,F)=-1THEN2
        205
PQ 2210 G=F:POKESX+1,214:POKES
        X,(F*5+3)*8+22:POKE532
        69,1
AG 2215 POKESX+1,214:POKESX,(F
        *5+3)*8+22
QS 2220 GETR$:DR=PEEK(56320):I
        F(R$="")AND(DR=127)THE
        N2220
ES 2225 G=F:IFNOT((R$="{LEFT}"
        )OR(DR=123))THEN2250
XF 2230 F=NF(F):IFC(0,F)<0THEN
        2230
CA 2245 GOTO2215
BP 2250 IFNOT((R$="{RIGHT}")OR
        (DR=119))THEN2275
DA 2255 F=NF(F+2):IFC(0,F)<0TH
        EN2255
QD 2270 GOTO2215
FH 2275 IFNOT((R$=CHR$(13))OR(
        DR=111))THEN2215
KB 2280 RETURN
DG 2300 FORI=0TO4:IF(S(P,I)=TP
        )AND(C(P,I)=2)THENC(P,
        I)=7:GOTO2310
HK 2305 IF((S(P,I)AND253)=(TPA
        ND253))AND(C(P,I)=2)TH
        ENC(P,I)=6:S(P,I)=TP
SA 2310 NEXT
AH 2315 FORI=0TO4:FORJ=0TO3:IF
        S(P,J)>S(P,J+1)THENGOT
        O2331
QE 2320 IFS(P,J)=S(P,J+1)THENI
        FC(P,J)>C(P,J+1)THENGO
        TO2331
GC 2325 T=C(P,J):C(P,J)=C(P,J+
        1):C(P,J+1)=T
KP 2330 T=S(P,J):S(P,J)=S(P,J+
        1):S(P,J+1)=T
XC 2331 NEXT
SX 2335 FORI=0TO4:PT(I)=0:IFS(
        P,I)=TPTHENPT(I)=C(P,I
        )+10:GOTO2350
SF 2340 IFC(P,I)=5THENPT(I)=9:
        GOTO2350
QC 2345 IF(S(P,I)<>S(P,NF(I)))
        AND(S(P,I)<>S(P,NF(I+2
        )))THENPT(I)=-1
CE 2350 NEXT
XM 2355 L=99:FORI=0TO4:IFPT(I)
        <LTHENF=I:L=PT(I)
SQ 2360 NEXT:RETURN
MJ 2500 FORI=0TO3:FORJ=0TO3:NS
        (I,J)=0:NEXT:FORJ=0TO4
        :IFC(I,J)<>2THEN2515
QH 2505 IFS(I,J)=TPTHENC(I,J)=
        7:GOTO2515
RD 2510 IFABS(S(I,J)-TP)=2THEN
        C(I,J)=6:S(I,J)=TP
GQ 2515 NS(I,S(I,J))=NS(I,S(I,
        J))+1:NEXT:NEXT
RX 2520 RETURN
KJ 2700 PRINT"{HOME}{2 DOWN}
        {2 RIGHT}{RVS}PARTNER?
        {OFF}";:LO=9:HI=10:XP=
        2:YP=12:GOSUB1910
FK 2705 FC(0)=0:IFAN=10THENFC(
        0)=2
XM 2710 PRINT"{HOME}{2 DOWN}
        {2 RIGHT}{RVS}OPPONENT
        S?{OFF}";:LO=9:HI=10:X
        P=2:YP=14:GOSUB1910
EG 2715 FC(1)=0:IFAN=10THENFC(
        1)=2
GG 2720 PRINT"{HOME}{2 DOWN}
        {2 RIGHT}{RVS}
        {10 SPACES}{OFF}";:RET
        URN
FS 3000 FORI=0TO7:FORJ=0TO3:CL
        (I,J)=0:NEXT:NEXT:CL(2
        ,TPAND253)=1
FF 3001 FORI=0TO3:SL(I)=0:NEXT
XX 3002 LD=FNNP(DL):DM=4:TR(0)
        =0:TR(1)=0:IFLH=0THEN3
        015
QR 3005 IFBD=2THENFORI=0TO4:PO
        KEX,19:POKEY,I*5+2:SYS
        PLT:PRINTEC$;:NEXT
JE 3010 DM=FNNP(FNNP(BD))
BB 3011 IFLH=1THENIFLD=DMTHENL
        D=FNNP(LD)
EJ 3015 FORTK=0TO4:P=LD:PS=0:T
        L=0:IFDM=PTHENP=FNNP(P
        )
SP 3020 GOSUB3500:WP=P:IFLH=1T
        HENPS=PS+1
DX 3021 SL(S(P,PC(P)))=1
PP 3025 IFS(P,PC(P))=TPTHENTL=
        1
FH 3030 FORI=1TO3:P=FNNP(P):IF
        P=DMTHEN3060
RS 3035 PS=PS+1:GOSUB3500:IFTL
        =0THEN3050
XF 3040 IFS(P,PC(P))=TPTHENIFC
        (P,PC(P))>C(WP,PC(WP))
        THENWP=P
JD 3045 GOTO3060
GS 3050 IFS(P,PC(P))=TPTHENWP=
        P:TL=1:GOTO3060
CK 3055 IFS(P,PC(P))=S(WP,PC(W
        P))THENIFC(P,PC(P))>C(
        WP,PC(WP))THENWP=P
DF 3060 NEXT:FORDE=1TO400:NEXT
KQ 3065 POKESX+3,(PX(WP)+1)*8+
        52:POKESX+2,(PY(WP)+1)
        *8+19:POKE53269,2
GX 3071 FORDE=1TO3000:NEXT:LD=
        WP:WT=WPAND253:TR(WT)=
        TR(WT)+1
HG 3072 POKEX,21:POKEY,31+4*WT
        :SYSPLT:N=TR(WT):GOSUB
        1465:POKE53269,0
QF 3075 FORI=0TO3:POKEX,PX(I):
        POKEY,PY(I):SYSPLT:PRI
        NTEC$;:C(I,PC(I))=-1:N
        EXT:NEXT
KJ 3078 POKEX,MX(BD)+2:POKEY,M
```

```
      Y(BD):SYSPLT:PRINT"
      {RVS}{9 SPACES}{OFF}";
DR 3080 BT=BDAND253:POKEX,15:P
      OKEY,28:SYSPLT
XA 3085 PRINT"{BLK}";:IFTR(BT)
      <3THEN3108
JP 3086 IFTR(BT)<5THEN3097
AB 3087 PW(BT)=PW(BT)+2+LH*2
CF 3090 IFBT=0THENPRINT"
      {2 SPACES}YOU
      {2 SPACES}WON{DOWN}
      {9 LEFT}ALL TRICKS";:G
      OTO3200
RB 3095 IFBT=1THENPRINT"COMPUT
      ER WON{DOWN}{11 LEFT}A
      LL TRICKS";:GOTO3200
FQ 3097 PW(BT)=PW(BT)+1
EA 3100 IFTR(0)>2THENPRINT"YOU
       WON HAND";:GOTO3200
FP 3105 IFTR(1)>2THENPRINT"
      {2 SPACES}COMPUTER
      {DOWN}{8 LEFT}WON HAND
      ";:GOTO3200
SR 3108 PW(1-BT)=PW(1-BT)+2
GE 3110 IFTR(0)<3THENPRINT"YOU
      'VE{2 SPACES}BEEN
      {DOWN}{10 LEFT}EUCHRED
      !":GOTO3200
QM 3115 IFTR(1)<3THENPRINT"
      {2 SPACES}COMPUTER
      {DOWN}{8 LEFT}EUCHRED!
      ";:
GH 3119 FORDE=1TO4000:NEXT
EG 3200 FORDE=1TO4000:NEXT:PRI
      NT"{BLU}";
GJ 3205 RETURN
JJ 3500 IFP>0THEN3509
EG 3501 POKEX,15:POKEY,28:SYSP
      LT
GK 3502 PRINT"{BLK}{RVS} YOUR
      {2 SPACES}PLAY {OFF}
      {BLU}";:GOSUB2200
MA 3503 LS=S(LD,PC(LD)):IF(PS=
      0)OR(S(P,F)=LS)OR(NS(P
      ,LS)=0)THEN3507
QM 3504 GOSUB2215:GOTO3503
AF 3505 POKE53269,0:POKEX,19:P
      OKEY,F*5+2:SYSPLT:PRIN
      TEC$;:GOTO3530
PE 3507 POKEX,15:POKEY,28:SYSP
      LT:PRINT"{12 SPACES}"
BG 3508 POKE53269,0:POKEX,19:P
      OKEY,F*5+2:SYSPLT:PRIN
      TEC$;:GOTO3530
FJ 3509 IFTK<5THEN3515
ER 3510 FORK=0TO4:IFC(P,J)>-1T
      HENF=I
RH 3512 NEXT:GOTO3530
MH 3515 ON(PS+1)GOSUB4000,4100
      ,4200,4200
QF 3530 PC(P)=F:POKEX,PX(P):PO
      KEY,PY(P):SYSPLT:C=C(P
      ,F):S=S(P,F):GOSUB1200
RG 3535 NS(P,S(P,F))=NS(P,S(P,
      F))-1:CL(C(P,F),S(P,F)
      )=1
MX 3540 RETURN
SH 4000 IFNOT(NS(P,TP)=5-TK)TH
      EN4015
GR 4005 SP=TP:GOSUB5200:IFF=1T
      HENGOTO5150
MQ 4010 GOTO5160
BA 4015 IFNOT((LH=1)AND(BD=P))
      THENGOTO4030
SF 4020 IFNS(P,TP)>0THENSP=TP:
      GOTO5150
RR 4025 GOTO5050
QF 4030 GOSUB5000:IF(F=1)AND(A
      BS(BD-P)=2)THENSP=TP:G
      OTO5150
CH 4035 GOSUB5250:IFNOT((F=1)A
      ND(P=BD))THENGOTO5050
CF 4040 GOSUB5200:IFF=1THENSP=
      TP:GOTO5150
FQ 4045 IFNS(P,TP)>2THENSP=TP:
      GOTO5160
JC 4050 GOTO5050
HF 4100 IFNS(P,S(LD,PC(LD)))=0
      THEN4115
RJ 4105 GOSUB5300:SP=S(LD,PC(L
      D)):IFF=1THEN5150
DH 4110 GOTO5160
XR 4115 IFNS(P,TP)=5-TKTHENSP=
      TP:GOTO5160
SB 4120 IFNS(P,TP)=0THEN5100
PD 4125 IFC(LD,PC(LD))=5THENSP
      =TP:GOTO5160
QC 4130 IFBD<>PTHENSP=TP:GOTO5
      160
XM 4135 GOSUB5250:IFF=1THENSP=
      TP:GOTO5160
EB 4140 GOTO5100
SA 4200 IFNS(P,S(LD,PC(LD)))=0
      THEN4235
MJ 4201 SP=S(LD,PC(LD))
KH 4205 IF(SP<>TP)AND(TL=1)THE
      N5160
RS 4210 IFABS(WP-P)<>2THEN4225
PR 4215 GOSUB5300:IFF=1THENGOS
      UB5350:IFF=0THEN5150
GS 4220 GOTO5160
JK 4225 GOSUB5300:IFF=1THEN515
      0
DX 4230 GOTO5160
BH 4235 IFNS(P,TP)<5-TKTHEN427
      0
XG 4245 SP=TP:IFABS(WP-P)=2THE
      N5160
XA 4250 IFTL=0THEN5160
RQ 4255 GOSUB5300:IFF=1THENGOT
      O5400
MR 4260 GOTO5160
BS 4270 IFNS(P,TP)=0THENGOTO51
      00
KA 4275 IFABS(WP-P)<>2THEN4310
HD 4280 IF(TL=1)OR(PS=3)THEN51
      00
XP 4285 IFC(WP,PC(WP))=5THEN51
      00
HG 4290 IFC(WP,PC(WP))<4THENSP
      =TP:GOTO5160
PD 4300 GOSUB5250:IFF=1THENSP=
      TP:GOTO5160
FQ 4305 GOTO5100
GE 4310 IFTL=0THENSP=TP:GOTO51
      60
BP 4315 GOSUB5300:IFF=1THENSP=
      TP:GOTO5400
MQ 4320 GOTO5100
BQ 5000 F=0:FORA=0TO4:IFC(P,A)
      >5THENF=1
BA 5005 NEXT:RETURN
QR 5050 F=-1:FORA=0TO4:IF(SL(S
      (P,A))=0)AND(S(P,A)<>T
      P)THENIFC(P,A)=5THENF=
      A
BD 5055 NEXT:IFF>-1THEN5070
CS 5060 LC=-1:FORA=0TO4:IFS(P,
      A)<>TPTHENIFC(P,A)>LCT
      HENLC=C(P,A):F=A
SB 5065 NEXT
BQ 5070 RETURN
GK 5100 IFNS(P,TP)>0THEN5125
SD 5105 SP=-1:FORA=0TO4
HM 5110 IFS(P,A)<>TPTHENIF(C(P
      ,A)=5)AND(NS(P,S(P,A))
      >1)THENSP=S(P,A)
EJ 5115 NEXT:IFSP>-1THEN5160
PH 5120 GOTO5180
FX 5125 V=4:F=-1:FORA=0TO4
DG 5126 IFS(P,A)=TPTHEN5135
XP 5130 IF(NS(P,S(P,A))<>1)OR(
      SL(S(P,A))=1)THEN5135
FH 5131 IF(C(P,A)>=0)AND(C(P,A
      )<V)THENV=C(P,A):F=A
HJ 5135 NEXT:IFF=-1THEN5180
FA 5140 RETURN
BJ 5150 IFPS=3THEN5400
MQ 5151 V=-1:FORA=0TO4:IFS(P,A
      )=SPTHENIFC(P,A)>VTHEN
      V=C(P,A):F=A
MS 5155 NEXT:RETURN
SJ 5160 V=10:FORA=0TO4
EH 5161 IFS(P,A)=SPTHENIF(C(P,
      A)>=0)AND(C(P,A)<V)THE
      NV=C(P,A):F=A
RR 5165 NEXT:RETURN
SM 5180 V=10:FORA=0TO4:IFS(P,A
      )<>TPTHENIFC(P,A)>-1TH
      ENIFC(P,A)<VTHENV=C(P,
      A):F=A
QD 5185 NEXT:RETURN
XJ 5200 HT=8:F=0
QM 5205 HT=HT-1:IFHT>0THENIFCL
      (HT,TP)=1THEN5205
GJ 5210 IFHT<0THEN5240
QS 5215 FORA=0TO4:IFS(P,A)=TPT
      HENIFC(P,A)=HTTHENF=1
BG 5220 NEXT
DM 5240 RETURN
XG 5250 F=1:FORA=0TO4:IFC(P,A)
      >-1THENIF(S(P,A)<>TP)A
      ND(C(P,A)<5)THENF=0
PH 5255 NEXT:RETURN
DD 5300 F=0:FORA=0TO4:IFS(P,A)
      =S(WP,PC(WP))THENIFC(P
      ,A)>C(WP,PC(WP))THENF=
      1
AF 5305 NEXT:RETURN
ES 5350 F=0:FORA=0TO4:IFS(P,A)
      =S(WP,PC(WP))THENIFC(P
      ,A)-C(WP,PC(WP))=1THEN
      F=1
BK 5355 NEXT:RETURN
PH 5400 D=10:FORA=0TO4
BC 5405 IFS(P,A)=S(WP,PC(WP))T
      HENE=C(P,A)-C(WP,PC(WP
      )):IF(E<D)AND(E>0)THEN
      D=E:F=A
MX 5410 NEXT:RETURN
QE 6000 FORR=54272TO54296:POKE
      R,0:NEXT:POKE54275,1
QC 6010 POKE54277,21:POKE54278
      ,135:POKE54273,150:POK
      E54276,17
HX 6020 FORR=15TO0STEP-.2:POKE
      54296,R:NEXT
FR 6030 POKE54276,16:POKE54296
      ,0:RETURN
QG 10000 POKE53285,13:POKE5328
      7,11:POKE53288,5:POKE
      53276,2:POKE2040,13:P
      OKE2041,14
SX 10010 SA=832:FORJ=0TO1:SA=S
      A+J*64:FORI=0TO63:REA
      DA:POKESA+I,A:NEXT:NE
      XT:RETURN
PJ 10050 DATA 0,96,0,6,108,0,6
      ,108
SX 10051 DATA 0,6,109,128,3,10
      9,128,3
DF 10052 DATA 253,128,27,255,0
      ,13,255,0
PF 10053 DATA 15,255,0,7,254,0
      ,3,254
AX 10054 DATA 0,1,252,0,0,252,
      0,0
GF 10055 DATA 252,0,0,0,0,0,0,
      0
HS 10056 DATA 0,0,0,0,0,0,0,0
RH 10057 DATA 0,0,0,0,0,0,0,19
      0
PP 10060 DATA 10,170,168,10,14
      9,168,10,85
SR 10061 DATA 104,9,89,88,9,10
      6,152,9
```

```
SG 10062 DATA 153,88,9,153,88,
         9,153,88
BB 10063 DATA 9,153,88,9,106,8
         8,9,89
KH 10064 DATA 152,9,89,152,9,8
         9,152,9
RX 10065 DATA 89,152,9,170,88,
         9,89,88
DF 10066 DATA 10,85,104,10,149
         ,168,10,170
FX 10067 DATA 168,0,0,0,0,0,0,
         0
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Sprite Designer

See instructions in article on page 35 before typing in.

```
84D0:4C 35 85 20 95 88 C9 30 92
84D8:90 F9 C9 39 B0 F5 8D 21 0F
84E0:85 A9 03 20 6B 87 38 E9 4D
84E8:30 0A AA BD 23 85 85 FD F5
84F0:E8 BD 23 85 85 FE 6C FD 99
84F8:00 20 F0 85 4C D3 84 20 5B
8500:E4 FF F0 FB 60 00 A5 FB A5
8508:18 6D 05 85 85 FB 90 02 B2
8510:E6 FC 60 00 A5 FD 18 6D 9C
8518:13 85 85 FD 90 F4 E6 FE C3
8520:60 20 00 07 94 E3 93 94 C3
8528:8B CE 97 BF 98 6D 98 3E 85
8530:9B C7 95 88 96 A9 EF 8D FE
8538:28 03 A9 C1 8D 18 03 A9 E5
8540:8E 20 D2 FF A9 08 20 D2 75
8548:FF AD 86 02 8D FE 9F AD 05
8550:20 D0 8D FD 9F AD 21 D0 F7
8558:8D FC 9F AD 8A 02 8D FF AF
8560:9F A9 01 8D 86 02 8D 20 16
8568:D0 A9 15 8D 18 D0 20 F0 F6
8570:85 A9 00 85 F9 85 FA 8D 6A
8578:23 9F 20 90 FF A9 33 8D A4
8580:21 85 20 68 8E A9 0D 85 C2
8588:F7 85 F8 A9 30 8D 79 9F F4
8590:8D 7A 9F A9 80 8D 8A 02 E0
8598:A9 31 8D 21 85 A9 03 20 81
85A0:6B 87 4C E3 93 20 44 E5 96
85A8:A9 04 8D 5B 86 A9 42 85 D5
85B0:22 A9 70 8D 00 04 A9 6E EF
85B8:8D 27 04 A9 7D 8D E7 07 68
85C0:A9 6D 8D C0 07 A2 25 A9 70
85C8:43 9D 01 04 9D 51 04 9D 15
85D0:C1 07 CA 10 F4 20 75 88 74
85D8:A9 6B 8D 50 04 A9 73 8D 85
85E0:77 04 A5 22 85 02 A9 00 07
85E8:20 6B 87 A5 02 85 22 60 F4
85F0:20 48 86 A9 6B 8D 30 06 81
85F8:A9 72 8D 62 04 A9 7D 8D 9C
8600:42 06 A2 10 A9 43 9D 31 CB
8608:06 CA 10 FA A2 0A A0 12 0D
8610:A9 78 20 58 86 A9 01 4C C2
8618:6B 87 A9 FF 8D 15 D0 20 74
8620:A5 85 A9 72 8D 69 04 A9 81
8628:71 8D D9 07 20 87 88 A9 D6
8630:6B 8D E9 06 A9 73 8D F7 22
8638:06 A2 0C A9 43 9D EA 06 79
8640:CA 10 FA A9 02 4C 6B 87 50
8648:A9 00 8D 15 D0 8D 21 D0 FC
8650:A9 01 8D 86 02 4C A5 85 9E
8658:85 FB A9 FF 85 FC A9 28 F7
8660:8D 05 85 A5 22 91 FB 20 EF
8668:06 85 CA 10 F6 A5 22 60 27
8670:53 50 52 49 54 45 53 20 98
8678:44 45 53 49 47 4E 45 52 48
8680:00 31 2E 20 49 4E 49 54 0C
8688:49 41 4C 49 5A 45 0D 1D C7
8690:1D 32 2E 20 45 44 49 54 A2
8698:0D 1D 1D 33 2E 20 41 4E 0D
86A0:49 4D 41 54 45 0D 1D 1D C8
86A8:34 2E 20 53 41 56 45 0D 8F
86B0:1D 1D 35 2E 20 4C 4F 41 2F
86B8:44 0D 1D 1D 36 2E 20 44 8F
86C0:41 54 41 20 42 41 53 49 B4
86C8:43 0D 1D 1D 37 2E 20 4D 30
86D0:4F 4E 4F 43 48 52 4F 4D AE
86D8:45 0D 1D 1D 38 2E 20 4D 49
86E0:55 4C 54 49 43 4F 4C 4F 0A
86E8:52 45 44 0D 1D 1D 30 2E B5
86F0:20 51 55 49 54 0D 0D 1D AF
86F8:1D 41 43 54 49 4F 4E 20 D6
8700:00 30 20 20 20 20 31 11 16
8708:11 11 11 9D 9D 9D 9D 9D 1C
8710:9D 32 20 20 20 20 33 11 79
8718:11 11 11 9D 9D 9D 9D 9D 2C
8720:9D 34 20 20 20 20 35 11 0E
8728:11 11 11 9D 9D 9D 9D 9D 3C
8730:9D 36 20 20 20 20 37 11 A2
8738:11 11 11 9D 9D 9D 9D 9D 4C
8740:9D 9D 53 45 52 49 45 53 D9
8748:3A 11 9D 9D 9D 9D 9D 9D 82
8750:9D 53 50 52 49 54 45 3A 92
8758:11 11 9D 9D 9D 9D 9D 9D FD
8760:9D 2B 20 C3 C3 20 48 45 BD
8768:4C 50 00 0A 0A A8 A2 03 8D
8770:B9 87 87 48 C8 CA 10 F8 3E
8778:68 A8 68 AA 18 20 F0 FF C0
8780:68 A8 68 4C 1E AB 00 70 CF
8788:86 01 0C 81 86 03 02 01 F9
8790:87 03 1D 21 85 0D 09 81 CD
8798:91 16 1C 61 87 16 1C CF 2C
87A0:93 10 02 1F 96 07 18 2A DA
87A8:96 07 1F 14 96 05 18 A7 8A
87B0:94 10 02 13 95 12 02 A4 1D
87B8:97 10 02 4D 98 10 02 FF B5
87C0:9E 14 02 E4 91 0C 19 1C BD
87C8:92 0C 19 58 98 12 02 86 64
87D0:87 14 02 F9 93 10 02 7F E8
87D8:95 07 13 A6 95 15 02 4A 90
87E0:96 15 02 8C 99 14 02 B6 61
87E8:88 04 02 AF 89 0E 02 78 79
87F0:8A 11 17 85 9A 10 02 2D 0B
87F8:9B 10 02 A9 D8 8D 5B 86 EE
8800:A5 22 8D 00 D8 8D 27 D8 42
8808:8D E7 DB 8D C0 DB A2 25 0E
8810:9D 01 D8 9D 51 D8 9D C1 10
8818:DB CA 10 F4 20 75 88 3D 90
8820:50 D8 8D 77 D8 A2 1F 9D E5
8828:2C D8 CA 10 FA 8D 69 D8 99
8830:8D D9 DB 20 87 88 8D E9 5F
8838:DA 8D F7 DA A2 0C 9D EA 32
8840:DA CA 10 FA 8D 95 D8 8D 25
8848:9A D8 8D 35 D9 8D 3A D9 35
8850:8D D5 D9 8D DA D9 8D 75 80
8858:DA 8D 7A DA A2 09 9D 14 BF
8860:DB CA 10 FA A2 08 9D 3C 70
8868:DB CA 10 FA A2 08 9D 8C C8
8870:DB CA 10 FA 60 A2 16 A0 2E
8878:00 A9 28 20 58 86 A2 16 33
8880:A0 27 A9 28 4C 58 86 A2 D6
8888:14 A0 19 A9 78 4C 58 86 B5
8890:12 20 92 9D 00 A9 90 A0 47
8898:88 18 20 1E AB A9 EC 85 3D
88A0:A2 20 E4 FF D0 0F A5 A2 58
88A8:D0 F7 AD 90 88 49 80 8D D6
88B0:90 88 4C 95 88 60 1F 44 57
88B8:49 53 50 4C 41 43 45 20 D3
88C0:54 48 45 20 43 55 52 53 20
88C8:4F 52 20 20 9E 43 52 53 16
88D0:52 20 1C 4B 45 59 53 0D 8E
88D8:0D 1D 1D 1F 4D 4F 44 49 C6
88E0:46 59 20 20 20 4D 4F 4E 94
88E8:4F 43 48 52 4F 4D 45 20 FA
88F0:20 9E 53 50 41 43 45 11 DB
88F8:9D 9D 9D 9D 9D 53 48 49 E1
8900:46 54 2D 53 50 41 43 45 79
8908:0D 1D 1D 1D 1D 1D 1D 1D 13
8910:1D 1D 1F 4D 55 4C 54 49 7F
8918:43 4F 4C 4F 52 45 44 20 6F
8920:20 9E 30 20 20 20 31 20 F6
8928:20 20 32 20 20 20 33 11 94
8930:9D 9D 9D 9D 9D 9D 9D 9D 43
8938:9D 9D 9D 9D 9D 9D 1C 28 D2
8940:9E 5A 1C 29 20 28 9E 58 86
8948:1C 29 20 28 9E 43 1C 29 9D
8950:20 28 9E 56 1C 29 0D 0D 63
8958:1D 1D 1F 4D 4F 56 45 20 78
8960:41 52 4F 55 4E 44 20 53 FE
8968:43 52 45 45 4E 20 20 20 02
8970:9E 51 20 1C 4C 45 46 54 45
8978:20 20 9E 45 20 1C 42 4F 11
8980:54 54 4F 4D 0D 1D 1D 1F C7
8988:43 4F 4D 4D 4F 44 4F 52 0C
8990:45 20 4B 45 59 20 41 4E 28
8998:44 2E 2E 20 20 9E 57 20 6B
89A0:1C 52 49 47 48 54 20 9E 66
89A8:52 20 1C 54 4F 50 00 1F 90
89B0:53 59 4D 4D 45 54 52 59 BB
89B8:20 20 20 20 20 20 20 20 CB
89C0:20 20 20 20 20 9E 4B 20 24
89C8:1C 48 4F 52 49 5A 4F 4E AB
89D0:54 41 4C 0D 1D 1D 1F 43 97
89D8:4F 4D 4D 4F 44 4F 52 45 CE
89E0:20 4B 45 59 20 41 4E 44 FB
89E8:2E 2E 20 20 9E 49 20 1C 1B
89F0:56 45 52 54 49 43 41 4C 36
89F8:0D 0D 1D 1D 9E 46 31 20 DB
8A00:05 C3 C3 20 1C 43 4F 50 DF
8A08:59 0D 1D 1D 9E 46 32 20 15
8A10:05 C3 C3 20 1C 52 45 56 1E
8A18:45 52 53 45 0D 1D 1D 9E D8
8A20:46 33 20 05 C3 C3 20 1C 03
8A28:53 45 52 49 45 53 20 28 F6
8A30:9E 46 34 1C 29 0D 1D 1D 43
8A38:9E 46 35 20 05 C3 C3 20 B5
8A40:1C 53 50 52 49 54 45 20 AD
8A48:28 9E 46 36 1C 29 0D 1D 02
8A50:1D 9E 46 37 20 05 C3 C3 38
8A58:20 1C 53 50 52 49 54 45 99
8A60:27 53 0D 1D 1D 1D 1D 1D 06
8A68:1D 1D 1D 43 4F 4C 4F 52 C7
8A70:20 28 9E 46 38 1C 29 00 5C
8A78:9E 43 4C 52 20 05 C3 20 19
8A80:1C 44 45 4C 45 54 45 11 39
8A88:9D 9D 9D 9D 9D 9D 53 50 BB
8A90:52 49 54 45 11 11 9D 9D A5
8A98:9D 9D 9D 9D 9D 9D 9D 9D AD
8AA0:9D 9D 9E 2A 20 05 C3 C3 C2
8AA8:20 1C 4D 45 4E 55 11 11 CD
8AB0:11 9D 9D 9D 9D 9D 9D 9D 7F
8AB8:9D 05 53 54 52 49 4B 45 21
8AC0:20 41 20 4B 45 59 2E 2E 09
8AC8:2E 00 20 48 86 A9 6B 8D BC
8AD0:C8 04 8D 90 05 8D 08 06 7A
8AD8:8D 80 06 A9 73 8D EF 04 E5
8AE0:8D B7 05 8D 2F 06 8D A7 78
8AE8:06 A9 71 8D D4 07 A9 43 CB
8AF0:A2 25 9D C9 04 9D 91 05 AF
8AF8:9D 09 06 9D 81 06 CA 10 83
8B00:F1 A9 72 8D 94 06 A9 06 B7
8B08:8D 5B 86 A2 06 A0 14 A9 3C
8B10:42 85 22 A9 A8 20 58 86 85
8B18:A9 18 20 6B 87 A9 19 20 F9
8B20:6B 87 A9 1A 20 6B 87 A9 0D
8B28:00 85 C6 A5 C6 F0 FC A9 71
8B30:00 85 C6 4C 94 8B 30 31 AA
8B38:32 33 5A 58 43 56 20 A0 5A
8B40:11 91 1D 9D 86 87 88 8A AF
8B48:8B 8C 89 85 AB B3 B2 B1 15
8B50:A1 A2 93 2B 2A EB 8B EB 0A
8B58:8B EB 8B EB 8B 1F 8C 24 76
8B60:8C 29 8C 2E 8C C1 8B C1 C0
8B68:8B 49 8C 52 8C 5B 8C 6C A5
8B70:8C 80 8C 8F 8C 9B 8C 86 EA
8B78:8C 95 8C A3 8C AE 8C C2 02
8B80:8C C8 8C F3 8C 2B 8D 5A 67
8B88:8D 87 8D 09 8E 38 8E CA C7
8B90:8A 47 8E 00 20 1A 86 20 27
8B98:4D 8E 20 FF 84 8D 93 8B 0B
8BA0:A0 1E A2 00 DD 36 8B F0 B3
8BA8:07 E8 88 10 F7 4C 9A 8B 41
8BB0:8A 0A AA BD 55 8B 85 FD A2
8BB8:E8 BD 55 8B 85 FE 6C FD 16
8BC0:00 AD 1C D0 F0 03 4C 9A 9A
8BC8:8B AD 93 8B C9 20 D0 05 B1
8BD0:A9 20 4C D7 8B A9 51 85 F6
```

```
8BD8:B5 20 92 90 20 F9 90 20 58
8BE0:49 91 20 72 90 20 20 90 02
8BE8:4C 9A 8B AD 1C D0 D0 03 E1
8BF0:4C 9A 8B AD 93 8B 29 0F 4D
8BF8:C9 02 D0 08 A6 FA BD 27 D3
8C00:D0 4C 08 8C AA BD 24 9F 92
8C08:85 B6 20 92 90 AD 93 8B AC
8C10:20 36 8C 20 72 90 AD 93 1F
8C18:8B 20 37 8C 4C E2 8B A9 5D
8C20:30 4C 30 8C A9 31 4C 30 0E
8C28:8C A9 32 4C 30 8C A9 33 37
8C30:8D 93 8B 4C EB 8B 4A 4A 97
8C38:B0 05 A9 20 4C 41 8C A9 4C
8C40:51 85 B5 20 F9 90 4C 49 10
8C48:91 20 92 90 20 58 90 4C 5D
8C50:7A 8C 20 92 90 20 65 90 57
8C58:4C 7A 8C 20 92 90 20 72 53
8C60:90 AD 1C D0 F0 14 20 72 48
8C68:90 4C 7A 8C 20 92 90 20 81
8C70:82 90 AD 1C D0 F0 03 20 D6
8C78:82 90 20 20 90 4C 9A 8B 73
8C80:20 B9 8E 4C 89 8C 20 D4 42
8C88:8E 20 68 8E 4C 9A 8B 20 EA
8C90:04 8F 4C 9A 8B 20 27 8F 7D
8C98:4C 9A 8B A6 FA FE 27 D0 4D
8CA0:4C A8 8C A6 FA DE 27 D0 78
8CA8:20 70 8F 4C 9A 8B 20 0B F2
8CB0:90 A0 3F B1 FB 49 FF 91 D3
8CB8:FB 88 10 F7 20 B1 8F 4C A6
8CC0:9A 8B 20 8B 91 4C 9A 8B 45
8CC8:20 1E 8D A0 3E B1 FB 0A EF
8CD0:91 FB 88 B1 FB 2A 91 FB 85
8CD8:88 B1 FB 2A 91 FB 08 C8 1A
8CE0:C8 B1 FB 4A 28 2A 91 FB F7
8CE8:88 88 88 10 E0 CA 10 DB A8
8CF0:4C BC 8C 20 1E 8D A0 3C 97
8CF8:B1 FB 4A 91 FB C8 B1 FB AE
8D00:6A 91 FB C8 B1 FB 6A 91 A4
8D08:FB 08 88 88 B1 FB 0A 28 76
8D10:6A 91 FB 88 88 88 10 E0 34
8D18:CA 10 DB 4C BC 8C 20 0B 40
8D20:90 A2 00 AD 1C D0 F0 02 0F
8D28:A2 01 60 20 0B 90 A2 02 C4
8D30:A0 02 B1 FB 95 22 88 CA 23
8D38:10 F8 A0 03 B1 FB 88 88 F4
8D40:88 91 FB C8 C8 C8 C8 C0 CB
8D48:3F D0 F1 A2 02 A0 3E B5 64
8D50:22 91 FB 88 CA 10 F8 4C BD
8D58:BC 8C 20 0B 90 A2 02 A0 5D
8D60:3E B1 FB 95 22 88 CA 10 B8
8D68:F8 B1 FB C8 C8 C8 91 FB 01
8D70:88 88 88 88 C0 FF D0 F1 25
8D78:A2 02 A0 02 B5 22 91 FB EE
8D80:88 CA 10 F8 4C BC 8C 20 B2
8D88:0B 90 AD 1C D0 D0 31 A0 91
8D90:3E A2 07 18 B1 FB 2A 91 39
8D98:FB 88 88 B1 FB 6A 91 FB A8
8DA0:C8 C8 CA 10 EF B1 FB 2A 15
8DA8:91 FB 88 A2 07 B1 FB 2A E7
8DB0:66 02 CA 10 FA A5 02 91 DD
8DB8:FB 88 88 10 D4 4C BC 8C E3
8DC0:A0 3E A2 03 18 B1 FB 2A E9
8DC8:08 2A 66 22 28 66 22 CA 4B
8DD0:10 F5 88 88 A2 03 18 B1 0E
8DD8:FB 2A 08 2A 66 23 28 66 96
8DE0:23 CA 10 F5 A5 22 91 FB 76
8DE8:C8 C8 A5 23 91 FB 88 A2 B1
8DF0:03 B1 FB 2A 08 2A 66 02 D3
8DF8:28 66 02 CA 10 F5 A5 02 54
8E00:91 FB 88 88 10 BC 4C BC 47
8E08:8C 20 0B 90 A0 3C A2 00 19
8E10:B1 FB 9D 3C 03 C8 E8 B1 3B
8E18:FB 9D 3C 03 C8 E8 B1 FB 9B
8E20:9D 3C 03 E8 88 88 88 88 0A
8E28:88 10 E5 A0 3F B9 3C 03 B0
8E30:91 FB 88 10 F8 4C BC 8C 26
8E38:20 0B 90 A0 3E A9 FF 91 6E
8E40:FB 88 10 FB 4C BC 8C 20 CD
8E48:48 8F 4C F9 84 20 85 8E D4
8E50:20 A5 8E 20 68 8E 20 56 CE
8E58:8F 20 4E 92 20 E6 8E 20 12
8E60:3C 8F 20 67 93 4C B1 8F BA
8E68:A5 F9 0A 0A 0A 69 87 A8 66
8E70:A2 07 9D F8 07 88 98 CA 3A
8E78:10 F8 AD 21 85 C9 33 F0 4E
8E80:03 20 B1 8F 60 A9 FF 8D 8D
8E88:10 D0 A9 0A 85 22 A9 32 F1
8E90:85 23 A2 0E A5 23 9D 00 63
8E98:D0 CA CA A5 22 9D 00 D0 DC
8EA0:CA CA 10 F0 60 A9 AA A2 88
8EA8:0E 38 9D 01 D0 CA CA 9D 83
8EB0:01 D0 E9 20 CA CA 10 F2 56
8EB8:60 20 48 8F E6 F9 A5 F9 74
8EC0:C9 10 D0 04 A9 00 85 F9 73
8EC8:20 56 8F 20 4E 92 20 67 E3
8ED0:93 4C E6 8E 20 48 8F C6 98
8ED8:F9 A5 F9 C9 FF D0 E9 A9 F8
8EE0:0F 85 F9 4C C8 8E A2 00 B0
8EE8:A5 F9 C9 0A 90 04 E8 E9 81
8EF0:0A 18 69 30 8D 7A 9F 8D 6C
8EF8:1D 07 8A 69 30 8D 79 9F 98
8F00:8D 1C 07 60 E6 FA A5 FA 3D
8F08:C9 08 D0 0A A9 00 85 FA 1C
8F10:20 B9 8E 20 68 8E AD 21 7B
8F18:85 C9 33 F0 06 20 B1 8F 85
8F20:20 67 93 20 3C 8F 60 C6 45
8F28:FA A5 FA C9 FF D0 E7 A9 E6
8F30:07 85 FA 20 D4 8E 20 68 1F
8F38:8E 4C 16 8F A5 FA 18 69 20
8F40:30 8D 7B 9F 8D 44 07 60 30
8F48:20 64 8F BD 27 D0 99 7C 8A
8F50:9F 88 CA 10 F6 60 20 64 99
8F58:8F B9 7C 9F 9D 27 D0 88 EA
8F60:CA 10 F6 60 A5 F9 0A 0A 01
8F68:0A 18 69 07 A8 A2 07 60 6E
8F70:AD 1C D0 F0 06 20 6C 93 B3
8F78:4C D4 92 A9 79 85 FB A9 63
8F80:D8 85 FC A9 28 8D 05 85 AE
8F88:A6 FA BD 27 D0 85 22 29 ED
8F90:0F D0 05 A9 0C 4C 9A 8F FC
8F98:A9 00 8D 21 D0 A2 14 A0 2A
8FA0:17 A5 22 91 FB 88 10 FB 30
8FA8:20 06 85 CA 10 F1 4C 20 B7
8FB0:90 AD 1C D0 F0 03 4C D4 15
8FB8:92 20 70 8F 20 0B 90 A9 28
8FC0:79 85 FD A9 04 85 FE A9 36
8FC8:28 8D 13 85 A9 00 85 25 97
8FD0:A9 02 85 22 A0 00 A9 07 77
8FD8:85 23 84 02 A4 25 B1 FB 4D
8FE0:85 24 A4 02 06 24 90 05 67
8FE8:A9 51 4C EF 8F A9 20 91 AE
8FF0:FD C8 C6 23 10 EE E6 25 7B
8FF8:A9 3F C5 25 F0 0A C6 22 27
9000:10 D4 20 14 85 4C D0 8F 32
9008:4C 20 90 A6 FA BD F8 07 9B
9010:85 FB A9 00 85 FC A2 05 92
9018:06 FB 26 FC CA 10 F9 60 BA
9020:AD 21 85 C9 33 D0 01 60 EC
9028:20 C7 90 A0 00 AD 1C D0 27
9030:F0 03 4C 82 93 B1 FB C9 61
9038:20 F0 07 C9 51 F0 03 4C C3
9040:44 90 85 B5 A5 B5 C9 20 6B
9048:F0 05 A9 D1 4C 51 90 A9 E7
9050:A0 91 FB A9 01 91 FD 60 EA
9058:A5 F7 C9 17 D0 04 A9 02 E0
9060:85 F7 E6 F7 60 A5 F7 C9 F1
9068:03 D0 04 A9 18 85 F7 C6 E7
9070:F7 60 A5 F8 C9 18 D0 07 41
9078:A9 00 85 F8 20 58 90 E6 19
9080:F8 60 A5 F8 C9 01 D0 07 75
9088:A9 19 85 F8 20 65 90 C6 83
9090:F8 60 20 C7 90 A0 00 AD 7B
9098:1C D0 F0 1F A9 51 91 FB BD
90A0:A5 B6 91 FD A9 01 8D 05 C5
90A8:85 8D 13 85 20 06 85 20 EE
90B0:14 85 A9 51 91 FB A5 B6 06
90B8:91 FD 60 A5 B5 91 FB A6 1B
90C0:FA BD 27 D0 91 FD 60 A9 AF
90C8:00 85 FB 85 FD A9 04 85 47
90D0:FC A9 D8 85 FE A9 28 8D CA
90D8:05 85 8D 13 85 A6 F7 F0 68
90E0:0A 20 06 85 20 14 85 CA 4F
90E8:4C DF 90 A5 F8 8D 05 85 22
90F0:8D 13 85 20 06 85 4C 14 43
90F8:85 A5 F8 85 29 C6 29 A9 1E
9100:07 25 29 85 29 A5 F7 85 C2
9108:2A C6 2A C6 2A C6 2A A5 0A
9110:2A 18 65 2A 65 2A 85 2A A6
9118:A5 F8 85 FB C6 FB A9 18 4E
9120:25 FB 4A 4A 4A 18 65 2A 6A
9128:85 2A 20 0B 90 A5 2A 8D 4A
9130:05 85 20 06 85 A0 00 B1 FB
9138:FB 85 22 A9 80 A6 29 F0 7B
9140:05 4A CA 4C 3F 91 85 23 05
9148:60 A5 23 48 A5 22 48 20 54
9150:75 93 68 85 22 68 85 23 59
9158:B0 0A A5 B5 C9 51 D0 0D A8
9160:20 6E 91 60 A5 B5 C9 20 1F
9168:D0 03 20 6E 91 60 A5 23 1C
9170:49 FF 85 23 A5 22 45 23 7E
9178:49 FF A0 00 91 FB 85 22 FD
9180:60 43 4F 50 59 20 2E 2E 69
9188:2E 2E 00 A9 04 20 6B 87 E7
9190:20 0B 90 A5 FB 85 AA A5 E3
9198:FC 85 AB 20 FF 84 C9 86 3F
91A0:D0 09 20 B9 8E 20 68 8E 62
91A8:4C 9B 91 C9 8A D0 09 20 71
91B0:D4 8E 20 68 8E 4C 9B 91 DA
91B8:C9 87 D0 06 20 04 8F 4C 99
91C0:9B 91 C9 8B D0 06 20 27 0E
91C8:8F 4C 9B 91 C9 85 D0 0F 68
91D0:20 0B 90 A0 3F B1 AA 91 8A
91D8:FB 88 10 F9 20 B1 8F A9 4E
91E0:05 4C 6B 87 AB C3 C3 AE 22
91E8:11 9D 9D 9D 9D C2 30 20 02
91F0:C2 11 9D 9D 9D 9D C2 31 61
91F8:12 20 92 C2 11 9D 9D 9D 83
9200:9D C2 32 12 20 92 C2 11 ED
9208:9D 9D 9D 9D C2 33 12 20 18
9210:92 C2 11 9D 9D 9D 9D AB 75
9218:C3 C3 BD 00 C2 20 20 20 BE
9220:11 9D 9D 9D 9D C2 20 20 1B
9228:20 11 9D 9D 9D 9D C2 20 38
9230:20 20 11 9D 9D 9D 9D C2 CA
9238:20 20 20 11 9D 9D 9D 9D C6
9240:C2 20 20 20 11 9D 9D 9D AC
9248:9D C2 20 20 20 00 A6 F9 3B
9250:BD 68 9F 8D 1C D0 D0 19 1A
9258:A9 01 8D 86 02 A9 10 20 A3
9260:6B 87 AD 00 D8 29 0F D0 2D
9268:07 A9 01 85 22 20 FB 87 05
9270:60 20 58 97 A2 03 B9 28 0F
9278:9F 9D 24 9F 88 CA 10 F6 D9
9280:AD 25 9F 8D 25 D0 AD 27 81
9288:9F 8D 26 D0 AD 24 9F 8D 7D
9290:21 D0 A5 F8 4A B0 03 20 F9
9298:82 90 20 C0 93 8D 86 02 15
92A0:85 02 A9 0F 20 6B 87 A5 92
92A8:02 85 22 A9 4B 85 FB A9 21
92B0:DA 85 FC A9 28 8D 05 85 E5
92B8:A0 00 A2 01 BD 24 9F 91 E1
92C0:FB 20 06 85 E8 E0 04 D0 A8
92C8:F3 AD 00 D8 C5 22 F0 03 7C
92D0:20 FB 87 60 20 0B 90 A5 EF
92D8:FB 85 22 A5 FC 85 23 A9 E9
92E0:79 85 FB 85 FD A9 04 85 20
92E8:FC A9 D8 85 FE A9 00 85 8E
92F0:24 85 26 A4 24 B1 22 85 4A
92F8:25 A9 02 8D 05 85 8D 13 A0
9300:85 A2 03 A9 00 85 02 06 AD
9308:25 26 02 06 25 26 02 A5 57
9310:02 86 02 20 46 93 20 06 E2
9318:85 20 14 85 A6 02 CA 10 C7
9320:E2 E6 26 A9 03 C5 26 D0 1E
9328:12 A9 00 85 26 A9 10 8D A0
9330:05 85 8D 13 85 20 06 85 5C
9338:20 14 85 E6 24 A9 3F C5 9F
9340:24 D0 B0 4C 20 90 A8 C9 E6
9348:02 D0 08 A6 FA BD 27 D0 FD
9350:4C 56 93 B9 24 9F A0 00 22
9358:91 FD C8 91 FD A0 00 A9 16
9360:51 91 FB C8 91 FB 60 AD 8B
9368:1C D0 F0 08 A6 FA BD 27 34
9370:D0 8D 73 DA 60 46 22 08 E7
9378:46 23 B0 04 28 4C 75 93 D2
9380:28 60 A2 03 B1 FB C9 51 BA
9388:D0 06 B1 FD 29 0F 85 B6 F6
9390:BD 24 9F C5 B6 F0 05 CA 3E
9398:10 F6 A2 02 8A 18 69 B0 32
93A0:91 FB 20 C0 93 91 FD 85 04
93A8:22 A9 01 8D 05 85 8D 13 B0
93B0:85 20 06 85 20 14 85 A9 C1
93B8:A0 91 FB A5 22 91 FD 60 22
93C0:AD 24 9F C9 01 D0 05 A9 57
93C8:00 4C CE 93 A9 01 60 49 71
```

```
93D0:4E 49 54 49 41 4C 49 5A B8
93D8:45 20 28 59 2F 4E 29 20 6A
93E0:3F 20 00 A9 06 20 6B 87 59
93E8:20 3C 94 F0 06 20 4D 94 B0
93F0:4C F9 84 20 21 9E 4C F9 65
93F8:84 51 55 49 54 20 28 59 C2
9400:2F 4E 29 20 3F 20 00 A9 9F
9408:13 20 6B 87 20 3C 94 F0 B4
9410:DF 20 34 9E AD FF 9F 8D DB
9418:8A 02 AD FE 9F 8D 86 02 EE
9420:AD FD 9F 8D 20 D0 AD FC 09
9428:9F 8D 21 D0 A9 93 20 D2 64
9430:FF A9 ED 8D 28 03 A9 47 42
9438:8D 18 03 00 20 95 88 C9 C0
9440:59 D0 03 A9 FF 60 C9 4E A8
9448:D0 F2 A9 00 60 A9 00 85 FA
9450:FB A9 20 85 FC A9 80 8D 5B
9458:05 85 A2 1F A0 00 A9 FF 04
9460:91 FB 88 D0 FB 20 06 85 61
9468:20 06 85 CA 10 EE A9 01 11
9470:A2 7F 9D 7C 9F CA 10 FA 89
9478:A2 07 9D 27 D0 CA 10 FA A7
9480:A2 3F 9D 28 9F CA 10 FA 44
9488:A9 00 A2 0F 9D 68 9F CA 64
9490:10 FA A9 00 85 F9 85 FA CF
9498:A9 33 8D 21 85 20 68 8E 33
94A0:A9 0D 85 F7 85 F8 60 41 24
94A8:20 05 12 20 20 20 20 92 BB
94B0:05 20 20 20 20 45 20 9C 5D
94B8:12 20 20 20 20 92 05 20 6E
94C0:20 20 20 49 20 81 12 20 E5
94C8:20 20 20 92 05 20 20 20 40
94D0:20 4D 20 98 12 20 20 20 5C
94D8:20 92 05 0D 1D 1D 42 20 2A
94E0:90 12 20 20 20 20 92 05 88
94E8:20 20 20 20 46 20 1E 12 31
94F0:20 20 20 20 92 05 20 20 41
94F8:20 20 4A 20 95 12 20 20 DA
9500:20 20 92 05 20 20 20 20 C7
9508:4E 20 99 12 20 20 20 20 98
9510:92 05 00 43 20 1C 12 20 AF
9518:20 20 20 92 05 20 20 20 91
9520:20 47 20 1F 12 20 20 20 94
9528:20 92 05 20 20 20 20 4B B7
9530:20 96 12 20 20 20 20 92 A9
9538:05 20 20 20 20 4F 20 9A 0D
9540:12 20 20 20 20 92 05 0D E4
9548:1D 1D 44 20 9F 12 20 20 79
9550:20 20 92 05 20 20 20 20 18
9558:48 20 9E 12 20 20 20 20 86
9560:92 05 20 20 20 20 4C 20 56
9568:97 12 20 20 20 20 92 05 95
9570:20 20 20 20 50 20 9B 12 06
9578:20 20 20 20 92 05 00 53 BD
9580:50 52 49 54 45 53 20 C3 52
9588:C3 20 30 31 32 33 34 35 B2
9590:36 37 11 9D 9D 9D 9D 9D DC
9598:9D 9D 9D 12 20 20 20 20 B0
95A0:20 20 20 20 92 00 9E 54 10
95A8:59 50 45 20 41 20 4C 45 A7
95B0:54 54 45 52 20 4F 52 20 EB
95B8:9F 53 54 4F 50 9E 20 54 99
95C0:4F 20 45 4E 44 05 00 20 7F
95C8:61 97 A9 15 20 6B 87 20 EE
95D0:6B 97 A9 14 20 6B 87 20 EB
95D8:64 8F B9 7C 9F 9D 5E D9 23
95E0:88 CA 10 F6 20 FF 84 C9 48
95E8:03 F0 26 C9 41 90 F5 C9 35
95F0:51 B0 F1 38 E9 41 20 96 DD
95F8:97 85 22 20 64 8F A5 22 66
9600:99 7C 9F 88 CA 10 F9 A9 C9
9608:00 A6 F9 9D 68 9F 4C C7 1A
9610:95 4C F9 84 53 45 52 49 40
9618:45 53 3A 20 20 20 00 30 B7
9620:11 9D 31 11 9D 32 11 9D E9
9628:33 00 12 20 20 11 9D 9D 51
9630:20 D1 11 9D 9D D1 20 11 63
9638:9D 9D D1 D1 92 00 12 20 CB
9640:20 20 20 92 00 12 20 92 CD
9648:00 00 9E 54 59 50 45 20 45
9650:41 20 4C 45 54 54 45 52 D4
9658:20 4F 52 20 9F 52 45 54 DA
9660:55 52 4E 9E 2C 0D 1D 1D 6D
9668:9F 53 54 4F 50 9E 20 54 4B
9670:4F 20 45 4E 44 2C 20 9F 8D
9678:44 45 4C 9E 20 54 4F 20 9D
9680:43 41 4E 43 45 4C 05 00 03
9688:20 61 97 A9 07 20 6B 87 C2
9690:A9 08 20 6B 87 A9 16 20 7E
9698:6B 87 20 6B 97 20 58 97 9D
96A0:A2 03 B9 28 9F 9D 24 9F F4
96A8:88 CA 10 F6 A2 03 8E 49 C5
96B0:96 BD 24 9F 20 87 97 AE 14
96B8:49 96 CA 10 F1 A9 01 8D 50
96C0:86 02 AD 49 96 18 69 30 14
96C8:8D 46 96 A0 18 20 4C 97 9C
96D0:A9 45 A0 96 20 1E AB 20 92
96D8:FF 84 C9 14 D0 03 4C F9 C6
96E0:84 C9 0D F0 23 C9 03 D0 8A
96E8:03 4C 26 97 C9 41 90 E7 45
96F0:C9 51 B0 E3 38 E9 41 20 B7
96F8:96 97 AE 49 96 9D 24 9F D4
9700:20 87 97 A9 01 8D 86 02 FB
9708:A0 18 20 4C 97 A9 46 A0 E6
9710:96 20 1E AB EE 49 96 AD 88
9718:49 96 C9 04 D0 05 A9 00 F8
9720:8D 49 96 4C C2 96 20 58 09
9728:97 A2 03 BD 24 9F 99 28 03
9730:9F 88 CA 10 F6 20 64 8F 3C
9738:AD 26 9F 99 7C 9F 88 CA 93
9740:10 F6 A9 FF A6 F9 9D 68 2B
9748:9F 4C F9 84 AD 49 96 18 B9
9750:69 07 AA 18 20 F0 FF 60 F1
9758:A5 F9 0A 0A 18 69 03 A8 CF
9760:60 A9 0A 20 6B 87 A9 0B 45
9768:4C 6B 87 A2 00 A5 F9 C9 08
9770:0A 90 04 E8 E9 0A 18 69 E8
9778:30 8D 1D 96 8A 69 30 8D 18
9780:1C 96 A9 09 4C 6B 87 8D D5
9788:86 02 A0 1A 20 4C 97 A9 3C
9790:3E A0 96 4C 1E AB C9 01 D2
9798:D0 03 A9 00 60 C9 00 D0 21
97A0:02 A9 01 60 49 4E 54 45 D2
97A8:52 56 41 4C 20 49 4E 20 66
97B0:36 30 54 48 0D 1D 1D 4F 7C
97B8:46 20 53 45 43 4F 4E 44 0A
97C0:20 28 31 20 54 4F 20 32 84
97C8:35 35 29 3A 20 00 A5 F9 EE
97D0:48 A5 FA 48 A5 F8 48 A9 BC
97D8:0C 20 6B 87 20 60 A5 A2 6C
97E0:00 BD 00 02 C9 20 F0 04 54
97E8:E8 4C E1 97 E0 04 B0 E7 B5
97F0:A9 00 85 7A A9 02 85 7B 29
97F8:20 79 00 20 6B A9 A5 15 FA
9800:D0 D5 A5 14 F0 3B C6 14 1B
9808:A9 FF 45 14 85 14 20 1A CE
9810:86 A9 00 85 F9 85 FA 20 43
9818:4D 8E A5 14 85 A2 20 9F 20
9820:FF 8A C9 A0 F0 1B C9 20 DE
9828:D0 0A A9 00 85 C6 20 FF 01
9830:84 4C 1A 98 A5 A2 D0 E6 C3
9838:20 04 8F 20 B1 8F 4C 1A EC
9840:98 68 85 F8 68 85 FA 68 CF
9848:85 F9 4C F9 84 46 49 4C FF
9850:45 4E 41 4D 45 3A 20 00 08
9858:54 41 50 45 20 4F 52 20 65
9860:44 49 53 4B 20 28 54 2F 9E
9868:44 29 3A 20 00 20 1D 99 A3
9870:A0 00 20 BA FF A2 00 A0 CC
9878:20 A9 ED 8D 28 03 A5 2D 80
9880:48 A5 2E 48 A9 00 20 D5 EC
9888:FF 68 85 2E 68 85 2D A9 C4
9890:EF 8D 28 03 20 9B 99 C9 BE
9898:14 B0 D2 A2 7F BD 00 40 B7
98A0:9D 7C 9F CA 10 F7 A2 3F 45
98A8:BD 80 40 9D 28 9F CA 10 20
98B0:F7 A2 0F BD C0 40 9D 68 EE
98B8:9F CA 10 F7 4C F9 84 A2 E3
98C0:7F BD 7C 9F 9D 00 40 CA E2
98C8:10 F7 A2 3F BD 28 9F 9D B3
98D0:80 40 CA 10 F7 A2 0F BD D2
98D8:68 9F 9D C0 40 CA 10 F7 2B
98E0:20 1D 99 A0 FF 20 BA FF 9C
98E8:A5 2B 48 A5 2C 48 A9 00 F0
98F0:85 2B A9 20 85 2C A2 D0 D9
98F8:A0 40 20 0D 99 68 85 2C 05
9900:68 85 2B 20 9B 99 C9 14 1B
9908:B0 D6 4C F9 84 A9 ED 8D A6
9910:28 03 A9 2B 20 D8 FF A9 0E
9918:EF 8D 28 03 60 A9 0D 20 BF
9920:6B 87 20 60 A5 A2 00 BD 6A
9928:00 02 C9 20 F0 04 E8 4C CC
9930:27 99 E0 00 D0 0F 68 68 75
9938:AD 21 85 C9 36 D0 03 20 F2
9940:21 9E 4C F9 84 8A A2 00 68
9948:A0 02 20 BD FF A9 20 A2 B5
9950:23 9D FF 9E CA 10 FA A9 9C
9958:0E 20 6B 87 A9 11 20 6B BD
9960:87 20 95 88 C9 54 F0 07 23
9968:C9 44 F0 03 4C 36 99 85 D3
9970:02 20 D2 FF A5 02 C9 54 24
9978:D0 0A A9 12 20 6B 87 A9 54
9980:01 4C 86 99 A9 08 85 02 2C
9988:AA A9 04 60 55 4E 49 54 CC
9990:20 4E 4F 54 20 52 45 41 AC
9998:44 59 00 C9 05 D0 08 A9 06
99A0:17 20 6B 87 A9 FF 60 A5 01
99A8:02 C9 08 F0 03 A9 00 60 7E
99B0:A9 08 20 B4 FF A9 6F 20 AF
99B8:96 FF A9 00 85 22 20 A5 07
99C0:FF C9 0D F0 0F 85 23 A5 91
99C8:22 AA A5 23 9D FF 9E E6 AF
99D0:22 4C BE 99 20 AB FF A9 F2
99D8:0E 20 6B 87 A9 FF 85 7A D3
99E0:A9 9E 85 7B 20 79 00 20 FF
99E8:6B A9 A5 14 60 64 0A 01 DB
99F0:20 53 45 52 49 45 53 20 FC
99F8:20 20 00 20 53 50 52 49 10
9A00:54 45 20 30 00 20 53 50 2F
9A08:52 49 54 45 20 31 00 20 7D
9A10:53 50 52 49 54 45 20 32 0C
9A18:00 20 53 50 52 49 54 45 6A
9A20:20 33 00 20 53 50 52 49 FD
9A28:54 45 20 34 00 20 53 50 97
9A30:52 49 54 45 20 35 00 20 B5
9A38:53 50 52 49 54 45 20 36 38
9A40:00 20 53 50 52 49 54 45 92
9A48:20 37 00 20 35 33 32 38 70
9A50:37 2D 39 34 00 20 35 33 F4
9A58:32 38 31 2C 35 33 32 38 B0
9A60:35 2C 43 4F 4C 4F 52 53 30
9A68:2C 35 33 32 38 36 00 F0 16
9A70:99 FB 99 05 9A 0F 9A 19 54
9A78:9A 23 9A 2D 9A 37 9A 41 12
9A80:9A 4B 9A 55 9A 9E 54 59 CF
9A88:50 45 20 9F 2A 9E 20 54 95
9A90:4F 20 53 45 4C 45 43 54 86
9A98:05 20 20 20 30 30 20 20 01
9AA0:20 30 31 20 20 20 30 32 2E
9AA8:20 20 20 30 33 0D 1D 1D 22
9AB0:9E 53 45 52 49 45 53 20 FD
9AB8:4F 52 20 9F 52 45 54 55 CD
9AC0:52 4E 9E 2C 05 20 20 30 62
9AC8:34 20 20 20 30 35 20 20 DC
9AD0:20 30 36 20 20 20 30 37 04
9AD8:0D 1D 1D 9F 53 54 4F 50 54
9AE0:20 9E 54 4F 20 42 55 49 4B
9AE8:4C 44 2C 05 20 20 20 20 0D
9AF0:20 30 38 20 20 20 30 39 66
9AF8:20 20 20 31 30 20 20 20 BF
9B00:31 31 0D 1D 1D 9F 44 45 C4
9B08:4C 20 9E 54 4F 20 43 41 49
9B10:4E 43 45 4C 05 20 20 20 B5
9B18:20 20 20 31 32 20 20 20 F0
9B20:31 33 20 20 20 31 34 20 11
9B28:20 20 31 35 00 52 45 4D 12
9B30:41 52 4B 53 20 28 59 2F BE
9B38:4E 29 20 3F 20 00 A9 1B 48
9B40:20 6B 87 A9 00 85 69 85 5C
9B48:6A 85 6B A9 01 85 6C A0 B5
9B50:00 A9 03 85 6D A2 04 A9 52
9B58:51 99 94 06 98 48 8A 48 D4
9B60:20 FF 84 85 6E 68 AA 68 63
9B68:A8 A5 6E C9 14 D0 03 4C FD
9B70:F9 84 C9 0D D0 03 4C A0 9B
9B78:9B C9 03 D0 03 4C BD 9B BD
9B80:C9 2A D0 D8 B9 95 06 49 48
9B88:80 99 95 06 B9 96 06 49 F6
9B90:80 99 96 06 A5 6A 45 6C 6F
9B98:85 6A A5 69 45 6B 85 69 C4
9BA0:06 6C 26 6B A9 20 99 94 07
9BA8:06 C8 C8 C8 C8 CA D0 8A
9BB0:A6 98 18 69 14 A8 C6 6D 39
9BB8:10 9B 4C 4B 9B A9 00 C5 66
9BC0:69 D0 07 C5 6A D0 03 4C 07
```

```
9BC8:F9 84 20 F0 85 A9 1C 20 5C
9BD0:6B 87 20 3C 94 85 70 A4 A7
9BD8:2E A5 2D 18 69 FE 85 FD 08
9BE0:85 65 B0 01 88 84 FE 84 33
9BE8:66 20 34 9E A9 30 85 63 48
9BF0:A9 F2 85 64 A9 2C 8D F8 C2
9BF8:9E A9 00 85 6D 46 69 66 FF
9C00:6A B0 0B E6 6D A5 6D C9 11
9C08:10 D0 F2 4C B8 9C A2 00 1E
9C10:A5 6D C9 0A 90 04 E8 38 EF
9C18:E9 0A 09 30 8D F9 99 8A FE
9C20:09 30 8D F8 99 A5 6D 0A 73
9C28:0A 0A 69 80 20 10 90 A5 26
9B30:41 52 4B 53 20 28 59 2F BE
9B38:4E 29 20 3F 20 00 A9 1B 48
9B40:20 6B 87 A9 00 85 69 85 5C
9B48:6A 85 6B A9 01 85 6C A0 B5
9B50:00 A9 03 85 6D A2 04 A9 52
9B58:51 99 94 06 98 48 8A 48 D4
9B60:20 FF 84 85 6E 68 AA 68 63
9B68:A8 A5 6E C9 14 D0 03 4C FD
9B70:F9 84 C9 0D D0 03 4C A0 9B
9B78:9B C9 03 D0 03 4C BD 9B BD
9B80:C9 2A D0 D8 B9 95 06 49 48
9B88:80 99 95 06 B9 96 06 49 F6
9B90:80 99 96 06 A5 6A 45 6C 6F
9B98:85 6A A5 69 45 6B 85 69 C4
9BA0:06 6C 26 6B A9 20 99 94 07
9BA8:06 C8 C8 C8 C8 CA D0 8A
9BB0:A6 98 18 69 14 A8 C6 6D 39
9BB8:10 9B 4C 4B 9B A9 00 C5 66
9BC0:69 D0 07 C5 6A D0 03 4C 07
9BC8:F9 84 20 F0 85 A9 1C 20 5C
9BD0:6B 87 20 3C 94 85 70 A4 A7
9BD8:2E A5 2D 18 69 FE 85 FD 08
9BE0:85 65 B0 01 88 84 FE 84 33
9BE8:66 20 34 9E A9 30 85 63 48
9BF0:A9 F2 85 64 A9 2C 8D F8 C2
9BF8:9E A9 00 85 6D 46 69 66 FF
9C00:6A B0 0B E6 6D A5 6D C9 11
9C08:10 D0 F2 4C B8 9C A2 00 1E
9C10:A5 6D C9 0A 90 04 E8 38 EF
9C18:E9 0A 09 30 8D F9 99 8A FE
9C20:09 30 8D F8 99 A5 6D 0A 73
9C28:0A 0A 69 80 20 10 90 A5 26
9C30:FC 18 69 80 85 FC A5 70 FE
9C38:D0 06 20 39 9D 4C 4A 9C 42
9C40:A9 00 85 62 20 06 9D 20 99
9C48:0A 9D 20 03 9E A9 01 8D 4D
9C50:05 85 A9 20 85 29 A9 10 D8
9C58:85 2A A2 05 20 72 9D 20 A9
9C60:06 9D A9 7C 85 FB A9 9F 10
9C68:85 FC A5 6D 0A 0A 0A 8D 49
9C70:05 85 20 06 85 A0 07 B1 60
9C78:FB 29 0F 91 FB 88 10 F7 0F
9C80:A9 01 8D 05 85 85 29 A9 0F
9C88:08 85 2A A2 05 20 72 9D C1
9C90:20 06 9D A9 28 85 FB A9 A2
9C98:9F 85 FC A5 6D 0A 0A 8D 32
9CA0:05 85 20 06 85 A9 01 8D 84
9CA8:05 85 85 29 A9 04 85 2A 9B
9CB0:A2 05 20 72 9D 4C 03 9C 68
9CB8:A9 00 A8 91 FD C8 91 FD 29
9CC0:A9 02 8D 13 85 20 14 85 8C
9CC8:A5 FD 85 2D A5 FE 85 2E 3A
9CD0:20 12 9E 20 F0 85 20 1D 6F
9CD8:99 A0 FF 20 BA FF A6 FD 2A
9CE0:A4 FE 20 0D 99 20 9B 99 1F
9CE8:C9 14 B0 EA 20 21 9E 4C DF
9CF0:F9 84 A5 64 8D 03 02 A5 65
9CF8:63 8D 02 02 18 69 01 85 95
9D00:63 90 02 E6 64 60 A5 70 20
9D08:F0 2F 20 F2 9C A9 8F 8D F2
9D10:04 02 A4 62 B9 6F 9A 8D D6
9D18:26 9D C8 B9 6F 9A 8D 27 AA
9D20:9D A2 05 A0 00 B9 FF FF 64
9D28:F0 08 9D 00 02 E8 C8 4C 23
9D30:25 9D 20 43 9D E6 62 E6 D1
9D38:62 20 F2 9C A9 83 8D 04 4F
9D40:02 60 CA A9 00 9D 00 02 01
9D48:86 61 E8 8A 18 65 65 85 8B
9D50:65 90 02 E6 66 A5 65 8D 33
9D58:00 02 A5 66 8D 01 02 A4 48
9D60:61 B9 00 02 91 FD 88 10 80
9D68:F8 A5 65 85 FD A5 66 85 67
9D70:FE 60 86 61 A0 00 20 A0 10
9D78:9D 20 06 85 C6 2A D0 F2 17
9D80:20 42 9D C6 29 F0 18 A5 5F
9D88:29 29 03 D0 06 20 06 9D 6A
9D90:4C 96 9D 20 39 9D A9 10 F0
9D98:85 2A A2 05 4C 72 9D 60 8D
9DA0:A2 00 A9 30 8D F5 9E 8D 74
9DA8:F6 9E 8D F7 9E B1 FB DD C9
9DB0:ED 99 90 0A 38 FD ED 99 2B
9DB8:FE F5 9E 4C AF 9D E8 E0 30
9DC0:03 D0 EC A6 61 B1 FB C9 4D
9DC8:64 B0 06 C8 C9 0A B0 01 88
9DD0:C8 B9 F5 9E 9D 00 02 E8 61
9DD8:C8 C0 04 D0 F4 60 A9 00 B2
9DE0:8D EB 9D 8D EE 9D A0 20 B9
9DE8:A2 00 AD FF FF 8D FF FF 61
9DF0:EE EB 9D EE EE 9D CA D0 95
9DF8:F1 EE EC 9D EE EF 9D 88 5B
9E00:D0 E6 60 AD 0E DC 29 FE 7B
9E08:8D 0E DC A5 01 29 FE 85 B5
9E10:01 60 A5 01 09 01 85 01 03
9E18:AD 0E DC 09 01 8D 0E DC 13
9E20:60 20 03 9E A9 A0 8D EC B7
9E28:9D A9 20 8D EF 9D 20 DE 90
9E30:9D 4C 12 9E A9 20 8D EC 51
9E38:9D A9 A0 8D EF 9D 4C DE 09
9E40:9D 00 00 00 00 00 00 00 4C
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Mozart Magic

Article on page 45.

```
MF 5 PRINT CHR$(144):VOL 15
RM 10 SCNCLR:PRINT"{9 DOWN}
      {RVS}{14 RIGHT}MOZART MA
      GIC"
GH 20 TEMPO8:PLAY"O4QCICCCC.CS
      FQCRO3$BI$B$B$B$BSO4CO3$
      BA$BIARBQBIBBBBO4.CSDQES
      RE.FSDQCO3BO4C"
DB 30 DIMM$(7,11),M1$(8,11),M2
      $(2),R(7),R1(8)
RK 40 FORI=1TO7:FORJ=1TO11:REA
      DM$(I,J):NEXT:NEXT
EH 50 FORI=1TO8:FORJ=1TO11:REA
      DM1$(I,J):NEXT:NEXT
FR 60 M2$(1)="V2O4QDV3GV1O1IGO
      2SGFEDM":M2$(2)="V2O4QDV
      3GV1O1IGO2SBG#FEM"
GG 70 SCNCLR:FORV=1TO3
CB 80 PRINT"{HOME}{DOWN} CHOOS
      E AN INSTRUMENT FOR VOIC
      E"V
PJ 90 PRINT"{DOWN} {RVS}0{OFF}
      PIANO
ME 100 PRINT"{DOWN} {RVS}1
      {OFF} ACCORDION
EF 110 PRINT"{DOWN} {RVS}2
      {OFF} CALLIOPE
KS 120 PRINT"{DOWN} {RVS}3
      {OFF} DRUM
DM 130 PRINT"{DOWN} {RVS}4
      {OFF} FLUTE
FR 140 PRINT"{DOWN} {RVS}5
      {OFF} GUITAR
EB 150 PRINT"{DOWN} {RVS}6
      {OFF} HARPSICHORD
DB 160 PRINT"{DOWN} {RVS}7
      {OFF} ORGAN
CD 170 PRINT"{DOWN} {RVS}8
      {OFF} TRUMPET
FE 180 PRINT"{DOWN} {RVS}9
      {OFF} XYLOPHONE
RD 190 GETKEYI$:IFI$<"0"ORI$>"
      9"THEN190
BE 200 INS=VAL(I$)
KM 210 IFV=1THENPLAY"V1
CX 220 IFV=2THENPLAY"V2
PR 230 IFV=3THENPLAY"V3
FS 240 IFINS=0THENPLAY"T0
KD 250 IFINS=1THENPLAY"T1
SA 260 IFINS=2THENPLAY"T2
DG 270 IFINS=3THENPLAY"T3
JP 280 IFINS=4THENPLAY"T4
RJ 290 IFINS=5THENPLAY"T5
FX 300 IFINS=6THENPLAY"T6
MQ 310 IFINS=7THENPLAY"T7
AG 320 IFINS=8THENPLAY"T8
PA 330 IFINS=9THENPLAY"T9
EJ 340 NEXT:SCNCLR
RQ 350 N=8:DO
AR 360 PRINT"{HOME}{DOWN} TEMP
      O{4 RIGHT}{3 SPACES}
      {4 LEFT}"N
DC 370 PRINT"{DOWN} {RVS}F
      {OFF}ASTER
FF 380 PRINT"{DOWN} {RVS}S
      {OFF}LOWER
AQ 390 PRINT"{DOWN} {RVS}E
      {OFF}XIT
JF 400 GETKEYT$
XS 410 IFT$="F"THENN=N+1:IFN=>
      255THENN=255
JQ 420 IFT$="S"THENN=N-1:IFN=<
      0THENN=1
DD 430 IFT$="E"THENEXIT
FF 440 LOOP:TEMPON
MF 450 FORI=1TO7:R(I)=INT(RND(
      1)*11+1):NEXT
JR 460 FORI=1TO8:R1(I)=INT(RND
      (1)*11+1):NEXT:SCNCLR
MP 470 FORK=1TO2:FORI=1TO7:PLA
      YM$(I,R(I)):NEXT:PLAYM2
      $(K):NEXT
KF 480 FORK=1TO2:FORI=1TO8:PLA
      YM1$(I,R1(I)):NEXT:NEXT
RF 490 GOTO70
AP 500 REM FIRST THROW
EX 510 DATA V1O2QCV3O4IECO3GM,
      V1O2QCV2EV3O3IGO4CEM,V1
      O2QCV2EV3O4IGECM,V1O2QC
      V2EV3O4SCO3BO4CEO3GO4CM
      ,V1O2QCV2EV3O5SCO4BO5CO
      4GECM,V1O2QCV3O4SEDEGO5
      CO4GM
BC 520 DATA V1O2QCV2EV3O4IGSFE
      DCM,V1O2QCV2EV3O4SECGEO
      5CO4GM,V3O4ICV1O2SCV2EG
      MV3O3IGV1O2SCV2EGMV3O4I
      EV1O2SCV2EGM
XS 530 DATA V1O2QCV2EV3O4IGCEM
      ,V1O2ICV2O3EV3O4CV1O2CV
      2O3EV3O4CV1O2CV2O3EV3O4
      CM
GX 540 REM SECOND THROW
MM 550 DATA V1O2QCV3O4IECO3GM,
      V1O2QCV2EV3O3IGO4CEM,V1
      O2QCV2EV3O4IGECM,V1O2QE
      V2GV3O4SCO3GO4CEO3GO4CM
XP 560 DATA V1O2QCV2EV3O5SCO4B
      O5CO4GECM,V1O2QCV3O4SED
      EGO5CO4GM,V1O2QCV2EV3O4
      IGSFEDCM,V1O2QCV2EV3O4S
      CO3GO4ECGEM
SK 570 DATA V1O2QCV2EV3O4ICO3G
      O4EM,V1O2QCV2EV3O4IGCMV
      1O2CV2GV3O4EM,V1O2ICV2O
      3EV3O4CV1O2CV2O3EV3O4CV
      1O2CV2O3EV3O4CM
BG 580 REM THIRD THROW
RH 590 DATA V1O1QBV2O2GV3O4SDE
      FDMV1O1IGV3O4CO3BM,V1O1
      QGV3O3IBO4DGM,V1O1QGV3O
      3IBO4SDO3BAGM,V1O2QGV2B
      V3O4IFDO3BM
```

```
SJ 600 DATA V1O1QBV2O2DV3O4SG#
       FGDO3BGM,V1O2QGV2BV3O4S
       FEFDCO3BM,V1O1QGV2O2GV3
       O3SBO4CDEMV1O1IBV2O2GV3
       O4SFDM
CC 610 DATA V1O2IGV2O3BV3O4DV1
       O2GV2O3BV3O4DV1O2GV2O3B
       V3O4DM,V1O1QGV3O3SBO4CD
       O3BAGM,V1O1QBV3O4IDO3BG
       M,V1O2QGV3O3SBABO4CDO3B
       M
QH 620 REM FOURTH THROW
JS 630 DATA V1O2QCV2EV3O4SCO3B
       O4CEO3IGM,V1O2QCV3O4SEC
       O3BO4CO3IGM,V1O2QEV2GV3
       O4ICO3GEM,V1O2QEV2GV3O4
       ICEO3GM
PC 640 DATA V1O2QEV2GV3O4SCO3B
       O4CO3GECM,V1O2QCV2EV3O4
       ICSCDIEM,V1O2QCV2O4ICV3
       EV2SCV3EV2DV3FV2IEV3GM
XP 650 DATA V1O2QEV2GV3O4ICSEC
       O3IGM,V1O2QEV2GV3O4SCO3
       GO4ECIGM,V1O2QEV2GV3O4I
       CSECIGM,V1O2QEV2GV3O4SC
       ECO3GIEM
KE 660 REM FIFTH THROW
HS 670 DATA V1O2QCV3O4I#FSA#FD
       #FM,V1O2ICV2O3#FV3O4DV1
       O2CV2O4DV3#FV1O2CV2O4#F
       V3AM,V1O2QCV3O4SDO3AO4#
       FDA#FM
HB 680 DATA V1O2ICV2O3#FV3O4DV
       1O2CV2O3#FV3O4DV1O2CV2O
       3#FV3O4DM,V1O2QCV3O4IDO
       3SABAO4I#FM,V1O2QCV3O4S
       D#CD#FA#FM
HM 690 DATA V1O2QCV2AV3O4I#FAM
       V1O2CV2AV3O4DM,V1O2ICV2
       #FV3O3AV1O2CV2#FV3O3SAO
       4DMV1O2ICV2AV3O4#FM
QX 700 DATA V1O2ICV2O4DV3#FV1O
       2CV2O4DV3#FV1O2CV2O4DV3
       #FM,V1O2ICV2DV3O4#FV1O2
       CV2DV3O4S#FDMV1O2ICV2DV
       3O4AM,V1O2QCV2AV3O4S#FD
       O3AO4A#FDM
EM 710 REM SIXTH THROW
QH 720 DATA V1O2IBV2O2DV3O4SG#
       FMV1O1IBV2O2DV3O4SGBMV1
       O1IBV2O2DV3O4DM,V1O1QBV
       2O2DV3O4IGSBGDO3BM,V1O1
       QBV2O2DV3O4IGBDM
FR 730 DATA V1O1QBV2O2GV3O3IAS
       #FGBO4GM,V1O1QBV2O2DV3O
       4SG#FGDMV1O1IBV2O2GV3O3
       SBGM,V1O1QBV3O4IGSBGDGM
       ,V1O1QBV2O2GV3O4IDSGDO3
       BO4DM,V1O1QBV2O2GV3O4ID
       SDGIBM
AA 740 DATA V1O1IBV2O2DV3O4SAG
       MV1O1IBV2O2DV3O4S#FGMV1
       O1IBV2O2GV3O4DM,V1O1QBV
       2O2DV3O4IGSGDIBM,V1O1QB
       V2O2DV3O4SGBGDO3IBM
EQ 750 REM SEVENTH THROW
XQ 760 DATA V1O2ICV3O4SECMV1O2
       IDV3O3SBAMV1O1IDV3O3SG#
       FM,V1O2ICV3O3SAO4EMV1O2
       IDV2O3SBV3O4DV2O3AV3O4C
       MV1O1IDV2O3SGV3BV2#FV3A
       M
BR 770 DATA V1O2ICV2O3SBV3O4DV
       2O3AV3O4CMV1O2IDV2O3AV3
       O4CV2O3GV3BMV1O1IDV2O3S
       GV3BV2#FV3AM,V1O2ICV3O4
       SEGMV1O2IDV3O4SDCMV1O1I
       DV3O3SBAM
AJ 780 DATA V1O2ICV3O3SAO4EMV1
       O2IDV3O4SDGMV1O1IDV3O4S
       #FAM,V1O2ICV3O4SEAMV1O2
       IDV3O4SGBMV1O1IDV3O4S#F
       AM,V1O2ICV3O4SCEMV1O2ID
       V3O4SGDMV1O1IDV3O3SAO4#
       FM
RP 790 DATA V1O2ICV3O4SEGMV1O2
       IDV3O4SDGMV1O1IDV3O3SBO
       4#FM,V1O2ICV3O4SECMV1O2
       IDV3O3SBGMV1O1IDV3O3SA#
       FM,V1O2ICV3O4SEO5CMV1O2
       IDV3O4SBGMV1O1IDV3O4SA#
       FM
DQ 800 DATA V1O2ICV3O3AV1O2DV3
       O4SDCMV1O1IDV3O3SBAM
HX 810 REM PART TWO FIRST THRO
       W
XG 820 DATA V1O2QDV3O4I#FSA#FM
       V1O2ICV3O4SD#FM,V1O2QDV
       2#FV3O4SDO3AO4D#FA#FM,V
       1O2IDV2AV3O4#FV1O2DV2#F
       V3O4AV1O2CV2DV3O4#FM
DG 830 DATA V1O2QCV2AV3O4S#FAO
       5DO4AMV1O2ICV2AV3O4#FAM
       ,V1O2QDV3O3SD#FAO4DMV1O
       2ICV3O4S#FAM
PP 840 DATA V2O4IDV3#FV1O1SDO2
       DMV3O4Q#FV1O2S#CDCDM,V1
       O2QDV2#FV3O4IA#FMV1O2CV
       2#FV3DM,V1O2QDV2#FV3O5I
       DO4SA#FMV1O2ICV2#FV3O4S
       DO3AM
GC 850 DATA V1O2QDV2#FV3O4SDO3
       AO4ID#FM,V1O2QCV2AV3O4S
       #FDO3IAMV1O2CV2AV3O4#FM
       ,V1O2QDV2#FV3O3IAO4DMV1
       O2CV2AV3O4#FM
PD 860 REM PART TWO SECOND THR
       OW
AG 870 DATA V1O1QBV2O2GV3O4IGS
       BGIDM,V3O4IGV1O1SBO2DMV
       3O3IGV1O2SGDMV3O3IGV1O1
       SBGM,V1O1QBV3O4SGBGBIDM
CC 880 DATA V1O1QBV2O2DV3O4SAG
       BGMV1O1IBV2O2DV3O4SDGM,
       V1O1QBV2O2DV3O4IGSDO3BM
       V1O1IBV2O2DV3O3GM
JK 890 DATA V1O1QBV2O2DV3O4SGB
       O5DO4BMV1O1IBV2O2DV3O4G
       M,V1O1QBV2O2DV3O4SGBGDO
       3BGM,V1O1QBV2O2DV3O4SGD
       GBMV1O1IBV2O2DV3O4SGDM
DP 900 DATA V1O1QBV2O2DV3O4SGB
       IGMV1O1IBV2O2GV3O4DM,V3
       O4IGV1O1SGBMV3O4QDV1O2I
       GO1BM,V1O1QBV3O4IGSBO5D
       O4IDM
AF 910 REM PART TWO THIRD THRO
       W
XJ 920 DATA V2O4ICV3EV1O2SCEMV
       2O4ICV3EV1O2SGEMV2O4ICV
       3EV1O3SCO2CM,V1O2QEV3O4
       SCO3GO4CEMV1O2EV3O4GV1O
       2CV2O4CV3EM,V1O2QCV2GV3
       O4IESGEMV1O2ICV2DV3O4CM
XE 930 DATA V1O2QCV2GV3O4SECEG
       O5CO4GM,V1O2QCV2GV3O4SE
       GO5CO4GMV1O2ICV2GV3O4SE
       CM,V2O4ICV3EV1O2SCO1BMV
       3O4QEV1O2SCDE#FM
PM 940 DATA V3O4IEV1O2SCV2EGMV
       3O4ICV1O2SCV2EGMV3O3IBV
       1O2SCV2EGM,V1O2QCV2GV3O
       4IESCEMV1O2CV2EV3O4GO5C
       M
RF 950 DATA V1O2QCV2GV3O4SECIE
       MV1O2CV2EV3O4GM,V1O2QCV
       2GV3O4SECO3IGMV1O2CV2GV
       3O4EM,V1O2QCV2GV3O4IEGM
       V1O2CV2EV3O5CM
FR 960 REM PART TWO FOURTH THR
       OW
EP 970 DATA V1O2QGV2O4ICV3EV2O
       3BV3O4DMV1O1GM,V1O1QGV2
       O2GV3O4SDO3BIGMV1O2GM,V
       1O2IGV2O4CV3EMV1O1GV2O3
       SBV3O4DV2O3GV3BV2IGM
JE 980 DATA V1O2QGV3O4SECDO3BI
       GM,V1O2IGV3O4SGEMV1O1IG
       V3O4SDO3BIGM,V1O2QGV3O3
       SBO4DGDMV1O1IGV3O3BM,V1
       O2IGV3O4SECMV1O1IGV3O3S
       BO4DIGM
RF 990 DATA V1O2QGV2BV3O4SDBGD
       O3IBM,V1O1QBV2O2GV3O4SD
       O3BIGMV1O1BV2O2DV3O4GM,
       V3O4IDV1O2SG#FMV3O3QBV1
       O2SGDO1BGM
GA 1000 DATA V1O2QGV2O3IBV3O4D
        SGBMV1O2IGV3O4DM
MK 1010 REM PART TWO FIFTH THR
        OW
MJ 1020 DATA V3O4IEV1O2SCV2EGM
        V3O4ICV1O2SCV2EGMV3O3I
        GV1O2SCV2EGM,V3O3IGV1O
        2SCV2EGMV3O4ICV1O2SCV2
        EGMV3O4IEV1O2SCV2EGM
AK 1030 DATA V3O4IGV1O2SCV2EGM
        V3O4EV1O2SCV2EGMV3O4IC
        V1O2SCV2EGM,V1O2QCV2EV
        3O4SCO3BO4CO4EMV1O2IEV
        2GV3O3SGO4CM
PX 1040 DATA V1O2QCV2EV3O5SCO4
        BO5CO4GMV1O2ICV2GV3O4S
        ECM,V1O2QCV2GV3O4SEDEG
        MV1O2ICV2EV3O5SCO4GM,V
        1O2QCV2EV3O4IGSFEDCM
SQ 1050 DATA V1O2QCV2EV3O4SCO3
        GO4ECGEM,V3O4ICV1O2SCV
        2EGMV3O3IGV1O2SCV2EGMV
        3O4IEV1O2SCV2EGM
XQ 1060 DATA V3O4IGV1O2SCV2EGM
        V3O4ICV1O2SCV2EGMV3O4I
        EV1O2SCV2EGM,V1O2ICV2O
        3EV3O4CV1O2CV3O3EV3O4C
        V1O2CV2O3EV3O4CM
GG 1070 REM PART TWO SIXTH THR
        OW
RX 1080 DATA V3O4IEV1O2SCV2EGM
        V3O4ICV1O2SCV2EGMV3O3I
        GV1O2SCV2EGM,V1O2QCV2E
        V3O3IBO4CMV1O2CV2GV3O4
        EM,V3O4IGV1O2SCV2EGMV3
        O4IEV1O2SCV2EGMV3O4ICV
        1O2SCV2EGM
BQ 1090 DATA V1O2QCV2EV3O4SCO3
        BO4EMV1O2ICV2EV3O3SGO4
        CM,V1O2QCV2EV3O5SCO4BO
        5CO4GECM,V1O2QCV2GV3O4
        SEDECMV1O2ICV2EV3O5SCO
        4GM
QH 1100 DATA V1O2QCV2EV3O4IGSF
        EMV1O2IEV2GV3O4SDCM,V1
        O2QCV2EV3O4SCO3GO4ECGE
        M,V3O4ICV1O2SCV2EGMV3O
        3IGV1O2SCV2EGMV3O4IEV1
        O2SCV2EGM
RX 1110 DATA V3O4IGV1O2SCV2EGM
        V3O4ICV1O2SCV2EGMV3O4I
        EV1O2SCV2EGM,V1O2ICV2O
        3EV3O4CV1O2CV2O3EV3O4C
        V1O2CV2O3EV3O4CM
BM 1120 REM PART TWO SEVENTH T
        HROW
XK 1130 DATA V1O2QFV2AV3O4SDFD
        FMV1O2IGV2O3DV3SBO4DM,
        V1O2QFV3O4SDFAFMV1O2IG
        V3O4SDO3BM,V1O2QDV3O4S
        DFO3AO4DMV1O2IGV3O3SBO
        4DM
BB 1140 DATA V1O2QFV3O4SD#CDFM
        V1O2IGV3O3SGBM,V1O2IFV
        3O4FV1O2DV3O4DV1O2GV3O
        4GM,V1O2SFV3O4FV1O2EV3
        O4EV1O2DV3O4DV1O2EV3O4
        EV1O2FV3O4FV1O2GV3O2GM
BQ 1150 DATA V1O2SFV3O4FV1O2EV
        3O4EV1O2IDV3O4DV1O2GV3
```

```
        O4GM,V1O2QFV3O4SFEDCMV
        1O2IGV3O3SBO4DM,V1O2QF
        V3O4SFDO3IAMV1O2GV3O3B
        M
PF 1160 DATA V1O2QFV3O4SFAO3IA
        MV1O2GV3O3SBO4DM,V1O2Q
        FV3O3IAO4SFDMV1O2IGV3O
        3SABM
XB 1170 REM SECOND PART EIGHTH
         THROW
CG 1180 DATA V3O4QCV1O2ICO1GCM
        ,V3O4QCV1O2ICO1GCM,V3O
        4QCV1O2ICO1GCM,V3O4QCV
        1O2ICO1GCM,V3O4QCV1O2I
        CO1GCM,V3O4QCV1O2ICO1G
        CM
CG 1190 DATA V3O4QCV1O2ICO1GCM
        ,V3O4QCV1O2ICO1GCM,V3O
        4QCV1O2ICO1GCM,V1O2QCV
        3O4ICO3CV1O1CM,V3O4QCV
        1O2ICO1GCM
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

The Construction Set

Article on page 39.

Program 1: Main Program

```
DD 100 POKE53280,1:POKE53281,1
CQ 110 PRINT"{CLR}{10 DOWN}
       {9 SPACES}{20 @}"
DJ 120 PRINT"{9 SPACES}{RVS}TH
       E CONSTRUCTION SET":A$=
       "{2 SPACES}{RED}{8 @}"
HP 130 PRINT"{7 DOWN}
       {12 SPACES}PLEASE WAIT.
       ."
JX 140 POKE55,0:POKE56,64:CLR:
       BA=16384:READA:FORI=1TO
       A:READX$:NEXT:FORI=1TO3
       52
HX 150 READX:S=S+X:NEXT:IFS<>3
       4546THENPRINT"ERROR IN
       {SPACE}DATA STATEMENTS.
       ":STOP
XF 160 RESTORE:DI=8:READNS:REM
        SET DI=1 FOR TAPE
PP 170 DATA 3,"HOUSE","CREATUR
       E","SHAPES"
FJ 180 PRINT"{UP}{14 SPACES}LO
       ADING...{2 SPACES}":FOR
       N=0TONS-1:CHSET=BA+6144
       +N*2048:READX$
JE 190 GOSUB1200:NEXT:PRINT"
       {CLR}{RVS}{GRN}":FORN=1
       TO40:PRINT"{Y}";:NEXT
GE 200 PRINTA$:PRINT"
       {2 SPACES}{RVS}{RED}
       {3 SPACES}F1{3 SPACES}
       {OFF}{BLU}{9 SPACES}FOR
        DIFFERENT PICTURE";
QJ 210 PRINTA$:PRINT"
       {2 SPACES}{RVS}{RED}
       {3 SPACES}F7{3 SPACES}
       {OFF}{BLU}{9 SPACES}FOR
        INSTRUCTIONS"
MK 220 PRINTA$:PRINT"
       {2 SPACES}{RVS}{RED}+ A
       ND - {OFF}{BLU}
       {9 SPACES}FOR EXTRA PIE
       CES"
BR 230 PRINTA$:PRINT"
       {2 SPACES}{RVS}{RED} CU
       RSOR {BLU}{OFF}
       {9 SPACES}SELECTS PIECE
       "
KG 240 PRINT"{2 SPACES}{RED}
       {10 @}":PRINT"
       {2 SPACES}{RVS}{RED} JO
       YSTICK {BLU}{OFF}
       {7 SPACES}POSITIONS PIE
       CE"
AC 250 PRINT"{2 SPACES}{RED}
       {13 @}":PRINT"
       {2 SPACES}{RVS}{RED} FI
       RE BUTTON {OFF}{BLU}
       {4 SPACES}STAMPS PIECE"
HF 260 PRINT"{2 SPACES}{RED}
       {10 @}":PRINT"
       {2 SPACES}{RVS}{RED} IN
       ST DEL {OFF}{BLU}
       {7 SPACES}ERASES PIECE"
AS 270 PRINT"{2 SPACES}{RED}
       {16 @}":PRINT"
       {2 SPACES}{RVS}{RED} SH
       IFT CLR HOME {OFF}{BLU}
        ERASES SCREEN"
GH 280 PRINT"{2 SPACES}{RED}
       {38 @}";
EK 290 PRINT"{2 SPACES}{RVS}
       {RED}PUSH ← AND THEN F1
       , F3, F5{12 SPACES}"
MA 300 PRINT"{2 SPACES}{BLU}TO
        CHANGE BORDER, SCREEN,
        & PICTURE{5 SPACES}COL
       OR. F7 RESUMES PROGRAM"
BP 310 PRINT"{RVS}{GRN}":FORN=
       1TO40:PRINT"{Y}";:NEXT
HR 320 PRINT"{13}{9 SPACES}USE
       {SPACE}PORT 2 FOR JOYST
       ICK.";
ME 330 V=53248:SB=15360+BA:SC=
       BA:H1=33:CB=1:MM=254:MN
       =-1
MQ 340 SE=8:FORSP=0TO2:FORN=0T
       O63:READD:POKESB+N+(64*
       SP),D:NEXT:NEXT
BP 350 FORA=53088TO53247:READI
       :POKEA,I:NEXT:SYS53088:
       POKE17400,241:POKE17401
       ,240
AB 360 POKE17402,242:POKEV+39,
       1:POKEV+3,205:POKEV+2,2
       4:POKEV+40,10:POKEV+28,
       1
DS 370 POKEV+37,5:POKEV+38,1:P
       OKEV+28,PEEK(V+28)OR4:P
       OKEV+41,1
BA 380 DIMA$(80,3):X=1:FORC=1T
       O20
KD 390 CO=CO+1:FORR=1TO3:A$(C,
       R)=STR$(X):X=X+32:NEXT:
       X=X-93:IFCO=10THENCO=0:
       X=X+66
XX 400 NEXTC:CO=0:FORC=21TO50:
       CO=CO+1:A$(C,1)=STR$(X)
       :A$(C,2)="256":A$(C,3)=
       "256"
QK 410 X=X+3:IFCO=10THENCO=0:X
       =X+2
SB 420 NEXT:POKE53265,PEEK(532
       65)AND239:POKE56578,PEE
       K(56578)OR3
GQ 430 POKE56576,(PEEK(56576)A
       ND252)OR2:A=4:SK=-16:PO
       KE53270,PEEK(53270)OR16
       :PY=1
RF 440 FORN=1TONS:GOSUB690:POK
       E53281,15:PRINT"{CLR}":
       POKE53281,0:NEXT:PY=0:A
       =4:SK=-16
PR 450 POKE53282,2:POKE53283,1
       :GOSUB690:POKEV+21,3:PO
       KEV,24:POKEV+1,50
JA 460 POKEV+27,0:POKE53265,PE
       EK(53265)OR16:X=24:Y=50
QE 470 J2=PEEK(56320):FR=J2AND
       16:JV=15-(J2AND15):IFJV
       =1THENY=Y-SE:IFY<50THEN
       Y=50
MC 480 IFJV=2THENY=Y+SE:IFY>19
       4THENY=194
KQ 490 IFJV=4THENX=X-SE:IFX<24
       THENX=24
SD 500 IFJV=8THENX=X+SE:IFX>31
       2THENX=312
HE 510 TR=(X>255):POKEV+16,(PE
       EK(V+16)AND(MM))ORTR*(M
       N):POKEV+SP*2,X+(TR*255
       )
XB 520 POKEV+(SP*2)+1,Y+2:IFFR
       =0THEN640
GP 530 GETKH$:IFKH$=""THEN470
XM 540 IFVAL(KH$)<4ANDVAL(KH$)
       >0THENSE=8*VAL(KH$):GOT
       O470
BR 550 IFKH$="-"THENCB=CB-9:IF
       CB<0THENCB=37
FF 560 IFKH$="+"THENCB=CB+9:IF
       CB>37THENCB=1
FG 570 IFKH$="+"ORKH$="-"THENG
       OSUB730:GOTO530
RB 580 IFKH$="{RIGHT}"ORKH$="
       {LEFT}"THEN GOSUB780:GO
       TO470
BD 590 IFASC(KH$)=20THEN640
JS 600 IFKH$="{CLR}"THENPOKE53
       281,15:PRINT"{CLR}":POK
       E53281,0:GOSUB730:GOTO4
       70
RR 610 IFKH$="{F1}"THENGOSUB69
       0
XP 620 IFKH$="{F7}"THENGOSUB12
       60
GR 630 GOTO470
MF 640 X1=(X-24)/8:Y1=((Y-50)/
       8)*40:S=SC+X1+Y1:FORR=1
       TO3:FORC=0TO2
FP 650 P=VAL(A$(CB+CX,R))+C:IF
       KH$<>""THEN:IFASC(KH$)=
       20THENP=32
QM 660 IFP>255THENP=32
BQ 670 POKES+C+(R-1)*40,P:NEXT
       :IFVAL(A$(CB+CX,1))+C>1
       85THENR=3
KK 680 NEXT:FR=1:GOTO470
CK 690 A=A+2:SK=SK+16:IFA>12OR
       (A-6)/2=NSTHENA=6:SK=0
MR 700 SC=BA+1024*SK/16:POKE53
       272,(PEEK(53272)AND15)O
       RSK
CR 710 POKE53272,(PEEK(53272)A
       ND240)ORA:POKE648,SC/25
       6:IFPY=1THENRETURN
KG 720 POKESC+1016,241:POKESC+
       1017,240:POKESC+1018,24
       2:GOSUB730:RETURN
MQ 730 VA=0:CO=0:FORC=1TO9:CO=
       CO+1:FORR=1TO3:FORN=1TO
       3:P=VAL(A$(CB+C-1,R))-1
       +N
DS 740 IFP>255THENP=32
JM 750 POKESC+880+VA+N+(C-1)*4
       +(R-1)*40,P:NEXT:IFVAL(
       A$(CB+C-1,R))-1+N>185TH
       ENN=3
KS 760 NEXT:IFCO=9THENCO=0:VA=
       VA+120
EX 770 NEXT:GOSUB800:RETURN
SC 780 IFKH$="{LEFT}"THEN H1=H
       1-32:CX=CX-1:IFCX<0THEN
       CX=8:H1=288
BB 790 IFKH$="{RIGHT}"THEN H1=
```

```
        H1+32:CX=CX+1:IFCX>8THE
        NCX=0:H1=40
XM 800  IFVAL(A$(CB+CX,1))+C>18
        5THENSP=2:SV=-16:POKEV+
        21,6:MM=251:MN=-4
HQ 810  IFVAL(A$(CB+CX,1))+C<18
        9THENSP=0:SV=0:POKEV+21
        ,3:MM=254:MN=-1
EM 820  TZ=(H1>255):POKEV+16,(P
        EEK(V+16)AND(253))ORTZ*
        (-2):POKEV+2,H1+(TZ*255
        )
GP 830  RETURN
EH 840  DATA 3,252,0,7,254,0,15
        ,254,0,31
EP 850  DATA 110,0,31,247,0,29,
        255,0,55,253
DS 860  DATA 128,61,191,128,59,
        223,128,50,238,128
EB 870  DATA 115,187,128,99,187
        ,128,3,185,0,3,144,0
JK 880  DATA2,128,0,3,128,0,3,1
        28,0,2
PD 890  DATA128,0,1,0,0,0,0,0,0
        ,0
QA 900  DATA0,0,85,85,85,127,25
        5,253,112,0
RC 910  DATA13,112,0,13,112,0,1
        3,112,0,13
GE 920  DATA112,0,13,112,0,13,1
        12,0,13,112
SS 930  DATA0,13,112,0,13,112,0
        ,13
SG 940  DATA112,0,13,112,0,13,1
        12,0,13,112
PK 950  DATA0,13,112,0,13,112,0
        ,13,112,0
FH 960  DATA13,127,255,253,85,8
        5,85,247,85,85
EB 970  DATA85,106,170,169,96,0
        ,9,96,0,9
JP 980  DATA 96,0,9,96,0,9,96,0
        ,9,106
DK 990  DATA170,169,85,85,85,0,
        0,0,0,0
FQ 1000 DATA0,0,0,0,0,0,0,0,0,0,
        0
GR 1010 DATA 0,0,0,0,0,0,0,0,0,0
        ,0
DS 1020 DATA0,0,0,0,0,0,0,0,0,0,
        0
DG 1030 DATA 0,0,120,173,40,3,
        141,248,207,173
QA 1040 DATA41,3,141,249,207,1
        73,250,207,141,40
EM 1050 DATA3,173,251,207,141,
        41,3,88,96,165
CM 1060 DATA145,201,253,208,11
        6,120,169,0,133,198
XJ 1070 DATA32,159,255,32,228,
        255,201,0,240,246
XQ 1080 DATA201,133,240,15,201
        ,134,240,23,201,135
SK 1090 DATA240,31,201,136,240
        ,84,76,130,207,173
AK 1100 DATA32,208,24,105,1,14
        1,32,208,76,130
RJ 1110 DATA 207,173,33,208,24
        ,105,1,141,33,208
BH 1120 DATA 76,130,207,165,24
        3,141,252,207,165,244
PH 1130 DATA 141,253,207,169,0
        ,133,243,169,216,133
RS 1140 DATA 244,173,134,2,24,
        105,1,141,134,2
CF 1150 DATA 162,4,160,0,145,2
        43,136,208,251,202
GX 1160 DATA 240,5,230,244,76,
        216,207,173,252,207
SP 1170 DATA 133,243,173,253,2
        07,133,244,76,130,207
CS 1180 DATA 88,108,248,207,23
        7,246,123,207,2,0
GK 1190 DATA 255,255
AA 1200 FORI=1TOLEN(X$):POKE67
        8+I,ASC(MID$(X$,I)):NE
        XT
JM 1210 POKE781,167:POKE782,2:
        POKE780,LEN(X$):SYS654
        69
CX 1220 POKE780,1:POKE781,DI:P
        OKE782,0:SYS65466
MK 1230 POKE780,0:POKE781,0:PO
        KE782,CHSET/256:SYS654
        93
MB 1240 IF(PEEK(783)AND1)OR(19
        1ANDST)THENPRINT"LOAD
        {SPACE}ERROR":STOP
RJ 1250 CLOSE1:RETURN
FG 1260 POKE56578,PEEK(56578)O
        R3:POKE 56576,(PEEK(56
        576)AND252)OR3
FR 1270 POKE53270,PEEK(53270)A
        ND239:POKE53272,(PEEK(
        53272)AND15)OR16
EX 1280 POKE53272,(PEEK(53272)
        AND240)OR4:XX=PEEK(V+2
        1):POKEV+21,0:POKE5328
        1,0
EH 1290 GETK$:IFK$=""THEN1290
JA 1300 POKE56578,PEEK(56578)O
        R3:POKE56576,(PEEK(565
        76)AND252)OR2
XF 1310 POKE53270,PEEK(53270)O
        R16:POKE53272,(PEEK(53
        272)AND15)ORSK
BE 1320 POKE53272,(PEEK(53272)
        AND240)ORA:POKE53281,0
        :POKEV+21,XX:RETURN
```

Program 2: The Construction Set—House Character Set

See instructions in article on page 39 before typing in.

```
7000:00 00 00 00 00 00 00 00 E0
7008:00 00 00 03 03 00 00 00 31
7010:3C 3C 3C FF FF 3C 3C 3C 4B
7018:00 00 00 C0 C0 00 00 00 0B
7020:51 55 51 45 51 45 54 45 0B
7028:14 55 45 14 55 15 51 15 09
7030:51 45 51 45 11 55 45 11 03
7038:11 45 6A 60 60 60 6A 60 FF
7040:45 05 AA 28 28 28 AA 28 3C
7048:11 45 A9 09 09 09 A9 09 91
7050:11 54 55 45 55 15 5A 5B DC
7058:10 55 15 55 69 BE EB AA 57
7060:51 54 15 55 11 55 A5 E5 06
7068:15 14 55 51 55 15 55 54 96
7070:04 55 15 55 55 51 55 45 80
7078:41 55 14 55 55 14 55 15 E1
7080:11 54 55 15 5A 18 58 58 37
7088:51 51 55 55 AA 28 28 28 D4
7090:45 44 55 51 A5 25 25 24 15
7098:00 00 00 00 00 00 00 00 79
70A0:00 00 00 00 00 00 00 00 81
70A8:00 00 00 00 03 0F 33 FF 44
70B0:28 14 14 14 FF FF FF 3F AD
70B8:00 00 00 00 FF 3F FF FF 96
70C0:00 00 00 00 FF 3F FF F3 92
70C8:00 00 00 00 F0 3C FF FF 22
70D0:00 00 00 00 00 00 00 C0 72
70D8:00 00 00 00 00 00 00 00 B9
70E0:54 45 55 16 5A 5B 58 1B 55
70E8:55 61 A9 8A CE CF 00 CF 30
70F0:45 55 54 55 91 95 95 85 3D
70F8:00 00 00 00 00 00 00 00 D9
7100:00 00 00 00 00 00 00 00 E2
7108:00 00 00 00 00 01 06 1B 16
7110:3C 3C 3C 3C 55 AA 33 CF F6
7118:00 00 00 00 00 40 90 E4 02
7120:55 51 55 45 51 54 45 51 B8
7128:44 51 15 54 15 54 45 54 42
7130:55 51 45 55 11 55 45 55 CD
7138:60 60 6A 55 55 6A 60 60 7B
7140:28 28 AA 55 55 AA 28 28 B9
7148:09 09 A9 55 55 A9 09 09 E8
7150:5B 5B 5B 5B 58 5B 5B 5B 1B
7158:FF FF F7 F7 37 37 FF FF 50
7160:E5 25 E5 E5 E5 E5 E5 E5 13
7168:15 54 55 15 5A 18 58 58 23
7170:55 51 55 55 AA 28 28 28 C0
7178:55 44 55 51 A5 25 25 24 07
7180:18 58 58 18 5A 55 15 55 B9
7188:28 28 28 28 AA 55 15 55 3B
7190:25 25 25 25 A4 55 14 55 3E
7198:00 00 00 00 03 0F AA 56 7B
71A0:03 0F 3C FF 33 FF AA 95 D4
71A8:FC CF FC FF 33 FF AA 55 E1
71B0:FC FF F0 FF FF 3F AA 45 C7
71B8:FC FF F3 FF CF FF AA 54 C0
71C0:FC FF F3 FF CF FF AA 54 C8
71C8:FC FF F3 FF CF FF AA 54 D0
71D0:F0 FC F3 FF CF FF AA 54 12
71D8:00 00 00 C0 30 FC AA 69 FB
71E0:5B 5B 59 49 59 59 1A 59 C5
71E8:CF CF 65 65 65 65 AA 65 26
71F0:95 91 95 95 94 95 85 95 AA
71F8:00 00 00 00 00 00 00 00 DB
7200:00 00 00 00 00 00 00 01 E5
7208:5B 5B 5B 5B 58 5B 5A 55 CC
7210:C3 C3 00 00 00 F3 AA 55 42
7218:E5 E5 E5 E5 25 E5 A5 55 E5
7220:55 51 45 54 45 50 55 55 5D
7228:05 51 45 54 45 55 45 55 31
7230:51 55 11 55 45 51 51 55 F1
7238:60 60 60 60 60 60 6A 55 26
7240:28 28 28 28 28 28 AA 55 57
7248:09 09 09 09 09 09 A9 55 BA
7250:5B 5B 5B 5B 59 5A 6A AA 8E
7258:FF FF FF FF AA AA AA AA 3D
7260:E5 25 E5 E5 65 A5 A9 AA 5C
7268:18 58 58 18 5A 55 15 55 A3
7270:28 28 28 28 AA 55 15 55 25
7278:25 25 25 25 A4 55 14 55 28
7280:50 55 15 55 55 51 55 45 BA
7288:54 55 15 55 55 51 55 45 C4
7290:55 55 15 55 55 51 55 45 4D
7298:56 55 55 55 54 55 55 55 F5
72A0:94 95 95 94 95 95 91 95 EC
72A8:55 51 55 55 55 45 54 55 4A
72B0:55 51 55 55 55 45 54 55 52
72B8:55 51 55 55 55 45 54 55 5A
72C0:55 51 55 55 55 45 54 55 62
72C8:55 51 55 55 55 45 54 55 6A
72D0:55 51 55 55 55 45 54 55 72
72D8:69 65 65 65 65 65 65 65 BF
72E0:59 59 59 59 5A 6A 55 55 06
72E8:65 65 65 65 AA AA 55 55 DC
72F0:95 95 95 95 95 A5 55 55 55
72F8:00 00 00 00 00 00 00 00 DD
7300:00 00 00 00 00 00 00 00 E6
7308:00 00 00 00 00 00 00 00 EE
7310:00 00 00 00 00 00 00 00 F6
7318:02 01 09 05 26 1B 9B 56 DC
7320:80 40 60 50 98 E4 E6 95 24
7328:00 00 00 00 00 00 00 00 0F
7330:00 00 00 00 00 00 00 00 17
7338:00 00 00 00 00 00 00 00 1F
7340:02 03 0B 0F 2F 3F BE F9 29
7348:AA FF FF FF FF FF BF 6F 73
7350:AA FF FF FF FF FF FF FF 8C
7358:AA FF FF FF FF FF FA E5 70
7360:AA FF FF FF FF FF FF BF 5C
7368:AA FF FF FF FF FF FE F9 9C
7370:80 C0 E0 F0 F8 FC BE 6F 9B
7378:00 00 00 00 00 00 00 00 5F
7380:95 95 95 95 95 95 95 95 67
7388:55 50 50 50 55 55 55 55 3D
7390:56 56 56 56 56 56 56 56 77
7398:00 00 00 00 00 00 00 00 7F
73A0:00 00 00 00 00 00 00 00 87
73A8:82 69 55 55 55 55 55 55 2B
73B0:00 00 00 00 00 00 00 00 97
73B8:A8 2F 2F 2F 20 20 20 20 7A
73C0:00 00 C0 F0 FC 0F 03 00 F8
73C8:15 56 5A 6A 5B 5B 5B 5B 1C
73D0:AA AA AA AA FF FF C3 C3 03
73D8:56 95 A5 A9 E5 E5 E5 A5 D7
73E0:A8 54 54 54 54 54 54 54 F1
```

```
73E8:02 01 01 01 01 01 01 01 50
73F0:A8 54 54 54 54 54 54 54 02
73F8:00 00 00 00 00 00 00 00 DF
7400:00 00 00 00 00 00 00 00 E8
7408:00 00 00 00 00 00 00 00 F0
7410:02 01 09 01 25 15 95 56 6A
7418:55 55 56 56 5B 6F BF D5 1F
7420:55 55 95 95 E5 F9 FE 57 81
7428:80 40 60 50 58 54 56 95 C8
7430:00 00 00 00 00 00 00 00 19
7438:02 03 0B 0F 2F 3F BF AA D5
7440:F9 F9 F9 FA EA FF FF AA 95
7448:6F 6F 6F AF AB FF FF AA B5
7450:FF FF FF FF FF FF FF AA E3
7458:E5 E5 E5 EA AA FF FF AA 19
7460:BF BF BF BF AF FF FF AA 35
7468:F9 F9 F9 FA EA FF FF AA BD
7470:6F 6F 6F AF AB FF FF AA DD
7478:80 C0 E0 F0 F8 FC FE AA 61
7480:95 95 95 95 95 95 95 95 69
7488:05 05 05 55 55 55 55 50 26
7490:56 56 56 56 56 56 56 56 79
7498:00 00 00 00 00 00 00 00 81
74A0:00 00 00 00 00 00 00 00 89
74A8:45 45 45 55 55 55 55 51 7F
74B0:00 00 02 01 01 01 01 01 F8
74B8:20 20 22 A9 A9 65 55 55 7B
74C0:00 00 00 00 00 00 00 00 A9
74C8:5B 5B 5B 5B 5B 58 5B 5B A5
74D0:C3 C3 C3 FF FF 3F 3F FF 80
74D8:E5 E5 E5 E5 E5 E5 E5 E5 C1
74E0:54 56 55 55 54 54 54 54 7A
74E8:01 A9 55 55 55 55 55 55 BC
74F0:54 54 54 54 54 54 54 14 99
74F8:00 00 00 00 00 00 00 00 E1
7500:00 00 00 00 00 00 00 00 EA
7508:02 01 09 06 2B AF AF AA D7
7510:5B 6D BD FD FD FF FF AA 36
7518:6A 68 68 60 60 55 FF AA 68
7520:A9 29 29 09 09 55 FF AA 28
7528:E5 79 7E 7F 7F FF FF AA D2
7530:80 40 60 90 E8 F8 FE AA 54
7538:55 55 56 59 65 95 55 AA 5A
7540:65 99 56 55 55 55 55 AA B9
7548:55 55 55 95 65 59 56 AA 1F
7550:55 55 56 59 65 95 55 AA 72
7558:65 99 56 55 55 55 55 AA D1
7560:55 55 55 95 65 59 56 AA 37
7568:55 55 56 59 65 95 55 AA 8A
7570:65 99 56 55 55 55 55 AA E9
7578:55 55 55 95 65 59 56 AA 4F
7580:95 95 95 95 95 95 95 95 6B
7588:50 50 50 55 55 55 55 55 0F
7590:56 56 56 56 56 56 56 56 7B
7598:A0 50 50 55 55 55 55 55 47
75A0:A0 50 50 55 55 55 55 55 4F
75A8:51 51 51 55 55 55 55 55 10
75B0:01 01 01 01 01 01 01 01 9B
75B8:55 55 55 55 55 55 55 55 A3
75C0:00 00 00 00 00 00 00 00 AB
75C8:5B 5B 5B 5B 5A 5A 69 A5 0E
75D0:FF FF FF FF 96 55 69 AA 43
75D8:E5 A5 E5 E5 A5 A5 69 5A 2C
75E0:44 45 45 45 45 55 55 7B
75E8:55 55 55 55 55 55 55 55 D3
75F0:14 14 14 14 54 54 54 54 9F
75F8:00 00 00 00 00 00 00 00 E3
7600:00 00 00 00 00 00 00 00 EC
7608:88 88 AA 88 88 AA 88 88 C1
7610:88 88 AA 88 88 AA 88 88 C9
7618:88 88 AA 88 88 AA 88 88 D1
7620:BC BF 8F BF BF 8F BF 8F 94
7628:00 C3 EB EB EB 00 EB EB 65
7630:3E FE F2 FE FE F2 FE FE 0B
7638:AA BF BF BF 82 80 80 AA 20
7640:AA EB EB E8 28 28 28 AA 66
7648:AA FE FE 22 02 02 02 AA 13
7650:55 55 56 59 65 95 95 AA F4
7658:69 96 55 7D FF 7D 55 AA 6D
7660:55 55 95 65 59 56 56 AA D1
7668:AA 95 95 95 B4 9C 90 AA FF
7670:AA 55 41 00 00 00 00 AA DA
7678:AA 56 56 56 1E 36 06 AA 01
7680:02 01 01 A9 55 55 55 55 69
7688:A0 50 50 5A 55 55 55 55 89
7690:2A 15 15 95 55 55 55 55 D3
7698:55 15 55 54 55 15 55 55 64
76A0:55 15 55 54 55 15 55 55 6C
76A8:55 15 55 54 55 15 55 55 74
76B0:AA AA A8 AA 2A A8 AA AA 51
76B8:AA AA A8 AA 2A A8 AA AA 59
76C0:AA 2A A8 AA AA AA 8A AA 0D
76C8:FC FF FF FF 3F FF FF F3 22
76D0:FF FF FF FF 3F FF FF F3 AB
76D8:FF F3 FF FF 3F FF FF F3 B0
76E0:55 56 5B 6F 6F BF BF BF 2A
76E8:A9 FE FF FF FF FF FF FF 6A
76F0:54 54 94 E4 E4 F8 F8 F8 F3
76F8:00 00 00 00 00 00 00 00 E5
7700:00 00 00 00 00 00 00 00 EE
7708:00 00 00 03 0F 3F FF AA 55 EC
7710:3C FF FF FF FF FF AA 55 C7
7718:00 00 C0 F0 FC FF AA 55 C0
7720:A0 A0 AA A5 A5 A6 A8 A0 F0
7728:00 00 AA 69 82 00 00 00 17
7730:0A 0A AA 5A 5A 9A 2A 0A 3D
7738:A0 A0 A0 A0 A0 A0 A0 A0 27
7740:00 00 00 00 00 00 00 00 2F
7748:0A 0A 0A 0A 0A 0A 0A 0A 37
7750:00 00 00 00 00 00 00 00 3F
7758:55 55 55 55 55 55 55 55 47
7760:40 40 40 40 40 40 40 40 4F
7768:AA 80 80 80 80 80 80 AA 96
7770:AA 00 00 00 00 00 00 AA 5F
7778:AA 02 02 02 02 02 02 AA 64
7780:00 00 00 00 0A AA A5 55 0B
7788:00 00 0A AA A5 55 51 55 DD
7790:0A AA A5 51 55 45 55 55 B8
7798:A0 AA 5A 55 15 51 55 55 11
77A0:00 00 A0 AA 5A 55 15 55 F5
77A8:00 00 00 00 A0 AA 5A 15 11
77B0:00 00 00 02 01 0B 27 BD FF
77B8:28 BE 7D D7 FF 7D D7 FF 3E
77C0:00 00 00 80 40 E0 D8 7E 6D
77C8:02 00 00 00 00 00 00 00 B8
77D0:A0 BF BD BD 83 80 80 80 33
77D8:00 00 F0 F0 F0 00 00 00 7C
77E0:BF BF BF 6F 6F 5B 56 55 79
77E8:FF FF FF FF FF FF FE A9 7F
77F0:F8 F8 F8 E4 E4 94 54 54 7E
77F8:00 00 00 00 00 00 00 00 E7
```

Program 3: The Construction Set—Creature Character Set

See instructions in article on page 39 before typing in.

```
7000:00 00 00 00 00 00 00 00 E0
7008:00 00 00 03 03 0F 05 0A 81
7010:00 30 FC FF FF FF 55 AA F1
7018:00 00 00 00 00 C0 40 80 FC
7020:00 00 00 01 05 3F 55 0A EA
7028:14 55 55 69 55 FF 55 AA A9
7030:00 00 00 40 50 FC 55 A0 D6
7038:00 03 0F 0F 3F 3E 2A EA DE
7040:FF FF FF FF FF FF BE AA 49
7048:00 C0 F0 F0 FC BC A8 AB 5E
7050:0C 23 00 00 01 05 15 15 5B
7058:00 00 D7 69 55 55 55 55 CA
7060:30 C8 00 00 40 50 54 54 CB
7068:35 D5 50 4A C8 C0 F0 FF 33
7070:FF 7D 55 3C 3C 3C EA EA B2
7078:5C 57 05 A3 23 03 0F FF 7B
7080:AA 80 80 83 8F 8F 8F 8F 88
7088:AA 00 00 FF FF FF FF FF BE
7090:AA 02 02 C2 F2 F2 F2 F2 EF
7098:00 00 00 02 02 09 09 09 E8
70A0:2A AA AA AA AA 08 08 59 20
70A8:00 80 80 A0 A0 18 18 58 B1
70B0:00 03 0D 0D 0D 0A 0A 3F A8
70B8:D5 55 55 55 55 AA AA FF 84
70C0:55 C9 7A 7A 70 A0 A0 FC F9
70C8:00 03 0F 3F 3F 30 30 30 8B
70D0:FF FF FF FF 3F 3F 0C 0C CC
70D8:F0 FC FF FF CF 03 03 03 05
70E0:00 00 01 05 15 55 55 14 EE
70E8:15 55 55 55 55 55 55 14 68
70F0:00 40 50 54 54 55 55 14 E7
70F8:00 00 00 00 00 00 00 00 D9
7100:00 00 00 00 00 00 00 00 E2
7108:0B 0A 20 28 2A 22 08 0A 6D
7110:EF FE 64 64 9A 9A 98 FE EC
7118:C0 80 20 A0 A0 20 80 80 90
7120:25 AA A0 A1 2A 2A 0A 0A 86
7128:EB 7D 28 69 AA BE 82 AA FB
7130:58 AA 0A 4A A8 A8 A8 A8 B1
7138:E9 F9 FA FB FE FF 3E 3A 5C
7140:AA AA BE FF FF 96 AA A9 54
7148:6B 6F AF EF BF FF BC AC D5
7150:15 57 56 55 15 17 05 01 C3
7158:55 D7 96 7D FF 00 FF 7D 04
7160:54 D5 95 55 54 D0 50 40 B1
7168:FF FE FE FF 3D 3D 35 17 4B
7170:AA AA AF 9D 55 55 69 6A AA 2C
7178:BF AF AF 7F 5C 5C 54 B4 C6
7180:8F 8F 8F 8F 83 80 88 AA D3
7188:FF FF FF FF FF 82 00 AA 20
7190:F2 F2 F2 F2 C2 02 22 AA 44
7198:02 02 00 00 00 00 00 00 FC
71A0:99 A2 A2 AA 80 2A 2A 08 01
71A8:A1 A1 80 80 80 00 00 00 E0
71B0:3F 2A FF BF AD AF AB A9 E6
71B8:BF AF AE 6E 7D FF D7 AA 6A
71C0:BC AC BF 7E 7A FA FA 6A 2D
71C8:30 3C 0F 0F 0F 0F 0F 0F 87
71D0:0C 0F 0F 3F F0 FC CF 03 71
71D8:03 0F 0C CC FC FC FC 3C 61
71E0:14 05 05 05 05 01 01 01 2F
71E8:D7 04 55 51 51 96 FF 55 B2
71F0:14 50 50 50 50 40 40 40 45
71F8:00 00 00 00 00 00 00 00 DB
7200:00 00 00 00 00 00 00 00 E4
7208:0F 32 02 03 03 0D 3F 3F 7B
7210:DF 56 AA AB EF FD 57 FF B0
7218:C0 F0 00 00 00 C0 FC FC 93
7220:02 02 03 03 0D 35 15 15 93
7228:2A 80 AA EA FF 4C 55 75 97
7230:20 A0 B0 F0 D0 5C 54 55 68
7238:0E 03 00 03 3F 3F FF FF 0C
7240:96 AA FF FF FF FF FF FF 1B
7248:B0 C0 00 C0 FC FC FF FF 9D
7250:02 05 17 55 57 5F 7F 7F 66
7258:55 96 69 D7 D7 7D FF FF EC
7260:80 50 D4 55 D5 F5 FD FD 0A
7268:00 00 03 0F 3F F3 FB EB 4C
7270:3C FF FF FF FF FF FF FF 73
7278:00 00 C0 F0 FC CF EF EB 77
7280:00 00 00 00 00 00 03 0F 7A
7288:28 28 28 28 28 EB EF FF E3
7290:00 00 00 00 00 00 C0 F0 E7
7298:00 08 22 20 22 20 20 20 B7
72A0:AA 00 08 80 2A 80 2A 80 0C
72A8:80 28 22 82 22 82 20 82 22
72B0:AB 2A 2A 2A 0A 0A 16 16 98
72B8:55 D7 FF BE AA AA AA AA 2A
72C0:E8 A8 A8 A8 A0 A0 90 94 21
72C8:03 00 00 00 00 02 09 09 52
72D0:C0 FC 7F 1F 95 56 58 58 46
72D8:0C 30 F0 40 80 00 00 00 F5
72E0:22 22 21 22 22 22 22 22 A5
72E8:55 96 AA AA 69 69 55 69 23
72F0:88 88 48 88 88 88 88 88 CD
72F8:00 00 00 00 00 00 00 00 DD
7300:00 00 00 00 00 00 00 00 E6
7308:F3 F3 3C 3C 0F 0D 0F 02 FD
7310:DF FF DF FF 7D DF FF AB F9
7318:FF CF 3C 3C F0 70 F0 C0 2A
7320:55 51 55 15 17 0E 06 01 FF
7328:55 5D 55 75 55 5D 55 55 33
7330:55 55 45 45 54 7C A0 40 2A
7338:FF F3 F3 F0 FC 3C 3E 0E 0D
7340:FF FF FF FF FF FF FF FF 27
7348:FF FF CF CF CF 3C BC B0 BF
7350:7F 5F 17 05 02 00 00 03 15
7358:FF FF FF FF BF 3F 7F D5 0F
7360:FD F5 D4 D0 A0 00 40 F0 E1
7368:B0 A0 28 0A 82 A0 A4 A6 FB
7370:FF FF FF BE AA 28 00 82 BB
7378:0E 0A 28 A0 82 0A 1A 9A 03
7380:3C 3C 0F 03 00 00 00 00 A6
7388:AA AA AA EB AF BE BA AA 1C
7390:3C 3C F0 C0 00 00 00 00 CE
7398:00 08 08 08 00 00 0A 0A 21
73A0:08 A2 08 A2 08 A2 88 00 3B
```

```
73A8:02 82 00 82 00 80 A8 28 D4
73B0:55 55 55 51 51 51 51 51 1B
73B8:AA 69 55 55 69 55 69 55 18
73C0:55 55 55 55 45 45 45 45 B6
73C8:25 09 09 02 02 00 00 00 D5
73D0:58 58 58 5A 56 95 95 25 04
73D8:00 00 00 00 80 80 80 60 27
73E0:21 22 22 22 22 22 22 57 7C
73E8:69 69 69 AA 96 AA AA AA 16
73F0:48 88 88 88 88 88 88 C4 F3
73F8:00 00 00 00 00 00 00 00 DF
7400:00 00 00 00 00 00 00 00 E8
7408:0F 0F 0F 0F 0F 0F 0F 0F F0
7410:FE FF CF CF CF CF 03 03 36
7418:A0 F8 F8 F2 C0 C0 C0 C0 29
7420:01 00 05 15 35 D5 D5 D5 FD
7428:55 28 55 55 55 55 55 41 B1
7430:00 00 50 54 5C 57 57 57 AE
7438:0E 02 03 0F 0F 3F 3F 3F 2D
7440:BF FF FF FF FF FF C3 03 93
7448:A0 C0 F0 F0 F0 FC FC FC 51
7450:0F 3F 0F 0F 03 00 00 00 7B
7458:FF FF FF FF FF FF FF 3F 80
7460:FC FF FC FC F0 C0 C0 00 43
7468:A6 A6 26 06 82 A0 A4 A6 F9
7470:A6 A6 A6 A6 A6 24 00 82 DD
7478:9A 9A 98 90 82 0A 1A 9A 7C
7480:00 0A 09 09 0A 09 09 09 2D
7488:AA AA 55 55 AA 55 55 55 1C
7490:00 A0 60 60 A0 60 60 60 5B
7498:02 00 02 08 08 08 08 08 BB
74A0:00 A2 08 00 00 00 00 00 33
74A8:20 80 20 08 08 08 08 08 BE
74B0:A5 F5 35 05 00 05 05 05 04
74B8:69 55 69 41 00 55 55 51 3E
74C0:5A 5F 5C 50 00 50 50 50 71
74C8:00 00 00 00 01 04 11 41 2D
74D0:09 02 00 00 14 54 69 80 04
74D8:60 58 96 96 96 98 98 58 E4
74E0:17 13 07 03 03 03 03 03 58
74E8:FF FF C3 C3 C3 C3 C3 C3 FE
74F0:D5 D4 C4 C0 C0 C0 C0 C0 E9
74F8:00 00 00 00 00 00 00 00 E1
7500:00 00 00 00 00 00 00 00 EA
7508:0F 3C 3C 3C 3C 3C 3C 00 20
7510:03 03 03 03 00 00 00 00 CD
7518:C0 C0 C0 C0 F0 F0 F0 00 E4
7520:05 05 01 01 05 05 05 05 4A
7528:41 41 41 41 00 00 00 00 40
7530:5F 7F 7F 7F 53 50 50 50 5F
7538:FC F0 F0 F0 3C 3C 3C 3F 95
7540:00 00 00 00 00 00 03 03 34
7548:FF 3F 0F 3F FC F0 F0 C0 27
7550:00 00 00 00 03 0F 0F 02 AF
7558:3F 3F FF FF FF F3 F3 80 EA
7560:00 00 C0 C0 F0 FC FC A0 85
7568:A6 A6 26 06 80 A0 0A 00 10
7570:A6 A6 A6 A6 A6 00 02 A8 79
7578:9A 9A 98 80 02 28 80 00 24
7580:0A 09 09 0A 00 02 02 00 80
7588:AA 55 55 AA 80 22 22 80 C4
7590:A0 60 60 A0 80 20 20 80 3F
7598:08 00 02 02 02 02 02 00 04
75A0:00 00 00 00 00 00 00 80 0C
75A8:08 00 20 20 20 20 20 80 DF
75B0:01 01 01 01 01 01 01 04 9E
75B8:40 40 40 40 40 40 01 01 E5
75C0:50 50 50 50 50 50 40 00 3B
75C8:14 55 58 64 61 18 16 05 01
75D0:56 81 06 12 49 A5 16 68 9E
75D8:58 58 50 60 60 80 00 00 1B
75E0:03 03 03 03 01 01 01 05 B1
75E8:C3 C3 C3 C3 41 41 41 55 42
75F0:C0 C0 C0 C0 40 40 40 50 64
75F8:00 00 00 00 00 00 00 00 E3
7600:00 00 00 00 00 00 00 00 EC
7608:00 00 00 00 02 02 02 2A 3B
7610:00 00 00 AA AA AA 96 AA 7F
7618:00 00 00 00 80 80 80 A8 B4
7620:00 00 00 00 00 00 FF FF 0D
7628:3F FF FF FF FB EA FF FF 40
7630:00 C0 C0 C0 C0 C0 FF FF 7A
7638:00 00 00 03 0D 0D 0A 0A 10
7640:0D 35 57 57 55 55 AA AA 61
7648:C0 F0 0A 0A C0 70 A0 A0 5D
7650:00 00 00 01 01 07 55 55 71
7658:14 55 55 55 54 55 FF 55 F1
7660:00 00 00 40 40 D0 55 55 96
7668:04 05 05 05 05 05 05 05 D4
7670:41 41 41 41 41 41 41 41 5D
7678:10 50 50 50 50 50 50 50 45
7680:0F 03 03 03 03 03 03 03 73
7688:C0 00 00 00 00 00 00 00 D5
7690:FC 30 30 30 30 30 30 30 E3
7698:A0 F0 F0 E0 F0 FB FF CF 85
76A0:00 00 00 00 00 00 C3 C3 D8
76A8:0A 0F 0F 0B 0F EF FF F3 1D
76B0:FF FF FF FF FF FF FF FF 9D
76B8:FF FF F7 FF FF F7 FF FF 84
76C0:FF FF FF FF FF FF FF FF AD
76C8:55 55 55 55 55 55 55 55 B5
76D0:59 55 55 59 55 59 55 55 10
76D8:55 55 55 55 55 55 55 55 C5
76E0:AA 80 AA 80 AA 80 AA AA F7
76E8:AA 00 AA 00 AA 00 AA AA D5
76F0:AA 02 AA 02 AA 02 AA AA 86
76F8:00 00 00 00 00 00 00 00 E5
7700:00 00 00 00 00 00 00 00 EE
7708:00 02 02 02 02 02 02 0A FD
7710:00 AA 99 99 99 99 99 AA 87
7718:00 80 80 80 80 80 80 A0 E6
7720:82 28 0A 22 82 00 00 02 D3
7728:20 08 02 20 80 82 88 A0 2B
7730:20 80 88 22 A8 00 00 00 C7
7738:00 00 00 00 00 00 00 02 29
7740:AA AA AA AA AA AA AA AA 2F
7748:00 00 00 00 00 00 00 80 B7
7750:0A 0B 0B 0B 0B 0A 0B 2A D9
7758:AA AE EE FA FE FE AE AA D1
7760:00 00 00 00 00 00 00 80 CF
7768:0C 0F 0F 0F 0E 0F FC FC 97
7770:C3 C3 C3 E3 8A C3 C3 C3 97
7778:30 F0 F0 B0 F0 F0 3F 3F ED
7780:00 03 03 03 03 0F 33 C3 3F
7788:00 00 00 00 00 C0 33 33 14
7790:00 30 30 30 30 F0 3C 33 85
7798:00 00 00 00 03 04 0C 08 CF
77A0:0F 3F 3D FF CC 44 CC 88 28
77A8:00 C0 C0 F0 CC 44 CC 88 88
77B0:0C 3F 3E 02 00 00 00 00 5D
77B8:00 00 00 00 80 20 22 08 78
77C0:0F 3F 3B 08 20 80 00 00 F1
77C8:15 05 05 05 05 05 05 04 BE
77D0:41 41 41 41 41 41 41 41 BF
77D8:50 50 50 50 50 50 50 10 87
77E0:00 00 00 00 03 0D 01 00 1E
77E8:C0 C0 00 C0 F0 DC D0 C0 D1
77F0:0C 0C 00 0C 3F DD 1D 0C 61
77F8:00 00 00 00 00 C0 00 00 EA
```

Program 4: The Construction Set—Shapes Character Set

See instructions in article on page 39 before typing in.

```
7000:00 00 00 00 00 00 00 00 E0
7008:AA AA A0 A0 A0 A0 A0 A0 70
7010:AA AA 00 00 00 00 00 00 F0
7018:AA AA 0A 0A 0A 0A 0A 0A 71
7020:AA AA AA AA AA AA AA AA 01
7028:AA AA AA AA AA AA AA AA 09
7030:AA AA AA AA AA AA AA AA 11
7038:55 55 55 55 55 55 55 55 19
7040:55 55 55 55 55 55 55 55 21
7048:55 55 55 55 55 55 55 55 29
7050:FF FF FF FF FF FF FF FF 31
7058:FF FF FF FF FF FF FF FF 39
7060:FF FF FF FF FF FF FF FF 41
7068:FF FF F0 F0 F0 F0 F0 F0 94
7070:FF FF 3C 3C 3C 3C 3C 3C 24
7078:FF FF 0F 0F 0F 0F 0F 0F 0E
7080:00 00 00 00 00 00 00 00 61
7088:0C 0C 0C 3F 3F 3F FF FF DE
7090:00 00 00 00 00 00 C0 C0 B3
7098:00 00 00 00 00 00 00 02 7B
70A0:20 20 20 20 20 20 A8 56 C8
70A8:00 00 00 00 00 00 00 00 89
70B0:20 26 26 26 26 26 26 26 8E
70B8:00 00 40 66 66 66 66 66 08
70C0:02 26 66 66 66 66 66 66 5F
70C8:00 00 00 00 00 00 00 00 A9
70D0:04 04 04 0C 0C 0C 0C 0C AA
70D8:00 00 00 00 00 00 00 00 B9
70E0:00 00 00 00 00 00 00 00 C1
70E8:14 14 14 14 14 14 14 14 C9
70F0:00 00 00 00 00 00 00 00 D1
70F8:00 00 00 00 00 00 00 00 D9
7100:00 00 00 00 00 00 00 00 E2
7108:A0 A0 A0 A0 A0 A0 A0 A0 EA
7110:00 00 00 00 00 00 00 00 F2
7118:0A 0A 0A 0A 0A 0A 0A 0A FA
7120:AA AA AA AA AA AA AA AA 03
7128:AA AA AA AA AA AA AA AA 0B
7130:AA AA AA AA AA AA AA AA 13
7138:55 55 55 55 55 55 55 55 1B
7140:55 55 55 55 55 55 55 55 23
7148:55 55 55 55 55 55 55 55 2B
7150:FF FF FF FF FF FF FF FF 33
7158:FF FF FF FF FF FF FF FF 3B
7160:FF FF FF FF FF FF FF FF 43
7168:F0 F0 F0 FF F0 F0 F0 F0 3C
7170:3C 3C 3C FF 3C 3C 3C 3C 8F
7178:0F 0F 0F FF 0F 0F 0F 0F 6A
7180:00 03 03 03 0F 03 03 03 42
7188:FF FF FF FF FF FF FF FF 6B
7190:C0 F0 F0 F0 FC F0 F0 F0 BB
7198:09 25 96 5A 69 69 69 5A DF
71A0:65 A9 9A 56 55 65 55 56 9A
71A8:80 60 58 94 A4 A5 A4 94 D1
71B0:26 26 26 26 26 26 26 26 93
71B8:66 66 66 66 66 66 66 66 9B
71C0:66 66 66 66 66 66 66 66 A3
71C8:0A 09 09 09 FF 09 09 09 E3
71D0:AE 5D 5D 7F F7 7F 5D 5D DB
71D8:A8 58 58 58 FF 58 58 58 21
71E0:00 00 00 00 55 55 00 00 C3
71E8:14 14 14 14 55 55 14 14 DA
71F0:00 00 00 00 55 55 00 00 D3
71F8:00 00 00 00 00 00 00 00 DB
7200:00 00 00 00 00 00 00 00 E4
7208:A0 A0 A0 A0 A0 A0 AA AA 0B
7210:00 00 00 00 00 00 AA AA F4
7218:0A 0A 0A 0A 0A 0A AA AA DE
7220:AA AA AA AA AA AA AA AA 05
7228:AA AA AA AA AA AA AA AA 0D
7230:AA AA AA AA AA AA AA AA 15
7238:55 55 55 55 55 55 55 55 1D
7240:55 55 55 55 55 55 55 55 25
7248:55 55 55 55 55 55 55 55 2D
7250:FF FF FF FF FF FF FF FF 35
7258:FF FF FF FF FF FF FF FF 3D
7260:FF FF FF FF FF FF FF FF 45
7268:F0 F0 F0 F0 F0 F0 FF FF 7A
7270:3C 3C 3C 3C 3C 3C FF FF A0
7278:0F 0F 0F 0F 0F 0F FF FF 30
7280:00 00 00 00 00 00 00 00 65
7288:FF FF FF 3F 3F 3F 0C 0C 7C
7290:C0 C0 C0 00 00 00 00 00 1E
7298:96 25 09 02 00 00 00 00 53
72A0:9A A9 65 56 A8 20 20 20 75
72A8:58 60 80 00 00 00 00 00 E1
72B0:26 26 26 26 26 26 26 26 95
72B8:66 66 66 66 66 66 66 66 9D
72C0:66 66 66 66 66 66 66 66 A5
72C8:0A 00 00 00 00 00 00 00 B2
72D0:AE 0C 0C 0C 0C 0C 04 04 EE
72D8:A8 00 00 00 00 00 00 00 12
72E0:00 00 00 00 00 00 00 00 C5
72E8:14 14 14 14 14 14 14 14 CD
72F0:00 00 00 00 00 00 00 00 D5
72F8:00 00 00 00 00 00 00 00 DD
7300:00 00 00 00 00 00 00 00 E6
7308:55 55 00 00 55 55 00 00 EE
7310:55 55 00 00 55 55 00 00 F6
7318:55 55 00 00 55 55 00 00 FE
7320:FF FF 00 00 FF FF 00 00 07
7328:FF FF 00 00 FF FF 00 00 0F
7330:FF FF 00 00 FF FF 00 00 17
7338:AA AA 00 00 AA AA 00 00 1F
7340:AA AA 00 00 AA AA 00 00 27
7348:AA AA 00 00 AA AA 00 00 2F
7350:80 00 20 00 08 00 02 00 BF
7358:00 00 00 00 00 00 00 00 3F
```

```
7360:02 00 08 00 20 00 80 00 4B
7368:00 00 0A AA 2A 02 00 00 94
7370:0A AA A8 00 A0 AA 2A 00 20
7378:AA 80 00 00 00 00 A0 AA C0
7380:AA FF FF AA FF FF AA FC B9
7388:AA FF FF AA FF FF AA CC 91
7390:AA FF FF AA FF FF AA FF CC
7398:00 00 00 00 00 00 00 00 7F
73A0:00 00 00 00 00 00 00 00 87
73A8:00 01 01 05 05 15 15 55 3C
73B0:00 40 40 50 50 54 54 55 86
73B8:00 00 00 00 00 00 00 00 9F
73C0:00 00 00 00 00 00 00 00 A7
73C8:00 00 00 00 02 02 0A 0A E5
73D0:24 24 A5 A5 A5 A5 A5 A5 96
73D8:00 00 00 00 40 40 50 50 B3
73E0:AA BF 3F 2F 0F 0B 03 02 94
73E8:AA FF FF FF FF FF FF FF 25
73F0:AA FE FC F8 F0 E0 C0 80 29
73F8:00 00 00 00 00 00 00 00 DF
7400:00 00 00 00 00 00 00 00 E8
7408:55 55 00 00 55 55 00 00 F0
7410:55 55 00 00 55 55 00 00 F8
7418:55 55 00 00 55 55 00 00 01
7420:FF FF 00 00 FF FF 00 00 09
7428:FF FF 00 00 FF FF 00 00 11
7430:FF FF 00 00 FF FF 00 00 19
7438:AA AA 00 00 AA AA 00 00 21
7440:AA AA 00 00 AA AA 00 00 29
7448:AA AA 00 00 AA AA 00 00 31
7450:00 00 00 00 00 00 00 00 39
7458:82 00 28 00 28 00 82 00 CD
7460:00 00 00 00 00 00 00 00 49
7468:00 00 0A AA 2A 02 00 00 96
7470:0A AA A8 00 A0 AA 2A 02 24
7478:A8 80 00 00 00 00 A0 AA C1
7480:FC A8 FC FC A8 FC FC A8 5D
7488:CC 88 CC CC 88 CC CC 88 F9
7490:FF AA FF FF AA FF FF AA 24
7498:00 00 00 00 00 00 00 00 81
74A0:00 01 01 05 05 15 15 55 36
74A8:55 55 55 55 55 55 55 55 91
74B0:55 55 55 55 55 55 55 55 99
74B8:00 40 40 50 50 54 54 55 90
74C0:00 00 00 00 00 00 00 00 A9
74C8:2A 2A AA AA AA 55 55 15 BB
74D0:A5 A5 A5 A5 A5 5A 5A 5A AA
74D8:54 54 55 55 55 AA AA A8 54
74E0:00 00 00 00 00 00 00 00 C9
74E8:FF BE 3C 3C 3C 3C 3C BE D6
74F0:00 00 00 00 00 00 00 00 D9
74F8:00 00 00 00 00 00 00 00 E1
7500:00 00 00 00 00 00 00 00 EA
7508:55 55 00 00 55 55 00 00 F2
7510:55 55 00 00 55 55 00 00 FA
7518:55 55 00 00 55 55 00 00 03
7520:FF FF 00 00 FF FF 00 00 0B
7528:FF FF 00 00 FF FF 00 00 13
7530:FF FF 00 00 FF FF 00 00 1B
7538:AA AA 00 00 AA AA 00 00 23
7540:AA AA 00 00 AA AA 00 00 2B
7548:AA AA 00 00 AA AA 00 00 33
7550:02 00 08 00 20 00 80 00 3F
7558:00 00 00 00 00 00 00 00 43
7560:80 00 20 00 08 00 02 00 D3
7568:00 00 0A AA 2A 02 00 00 98
7570:0A AA A8 80 A0 AA 2A 02 2E
7578:A8 80 00 00 00 00 A0 AA C3
7580:FC FC AA FF FF AA FF AA D3
7588:CC CC AA FF FF AA FF AA B7
7590:FF FF AA FF FF AA FF AA 26
7598:00 01 01 05 05 15 15 55 30
75A0:55 55 55 55 55 55 55 55 8B
75A8:55 55 55 55 55 55 55 55 93
75B0:55 55 55 55 55 55 55 55 9B
75B8:55 55 55 55 55 55 55 55 A3
75C0:00 40 40 50 50 54 54 55 9A
75C8:15 05 05 01 01 00 00 00 38
75D0:5A 5A 5A 5A 5A 5A 5A 18 79
75D8:A8 A0 A0 80 80 00 00 00 60
75E0:00 02 03 0B 0F 2F 3F AA BB
75E8:FF FF FF FF FF FF FF AA 7E
75F0:00 80 C0 E0 F0 F8 FC AA 32
75F8:00 00 00 00 00 00 00 00 E3
7600:00 00 00 00 00 00 00 00 EC
7608:FF 02 02 02 02 02 02 02 F3
7610:FF 55 55 55 55 55 55 55 52
7618:FF 80 80 80 80 80 80 80 C4
7620:00 00 03 0F 0F 3F 3F FF 52
7628:3C FF FF FF FF D7 D7 55 97
7630:00 00 C0 F0 F0 FC FC FF B9
7638:C0 C0 C0 EA D5 D5 D5 D5 04
7640:00 00 00 AA 55 55 55 55 D7
7648:03 03 03 AB 57 57 57 57 B0
7650:A0 A0 AA A0 A0 AA A0 A0 A6
7658:00 00 AA 00 00 AA 00 00 45
7660:0A 0A AA 0A 0A AA 0A 0A E3
7668:00 00 00 00 02 08 20 80 46
7670:00 00 28 AA 00 00 00 00 0D
7678:00 00 00 00 80 20 08 02 FB
7680:F0 5C 5C 5C 57 57 57 55 6A
7688:00 00 00 00 00 00 00 C3 39
7690:0F 35 35 35 D5 D5 D5 55 53
7698:55 40 40 40 40 40 40 55 25
76A0:55 00 00 00 00 00 00 55 8D
76A8:55 01 01 01 01 01 01 55 14
76B0:80 80 AA 80 AA 80 AA 80 88
76B8:00 00 AA 00 AA 00 AA 00 A5
76C0:02 02 AA 02 AA 02 AA 02 59
76C8:EE BB EE BB EE BB EE BB B5
76D0:EE BB EE BB EE BB EE BB BD
76D8:EE BB EE BB EE BB EE BB C5
76E0:04 55 04 04 55 04 04 55 FD
76E8:10 55 10 10 55 10 10 55 96
76F0:41 55 41 41 55 41 41 55 97
76F8:00 00 00 00 00 00 00 00 E5
7700:00 00 00 00 00 00 00 00 EE
7708:02 02 02 02 02 02 02 FF F4
7710:55 55 55 55 55 55 55 FF A9
7718:80 80 80 80 80 80 80 FF 86
7720:FD FF 3F 3F 0F 03 00 00 6E
7728:55 55 D7 D7 FF FF FF 3C CB
7730:7F FF FC FC F0 C0 00 00 D8
7738:D5 D5 D5 D5 EA C0 C0 C0 3C
7740:55 55 55 55 AA 00 00 00 84
7748:57 57 57 57 AB 03 03 03 8B
7750:AA A0 A0 AA A0 A0 AA A0 F8
7758:AA 00 00 AA 00 00 AA 00 9C
7760:AA 0A 0A AA 0A 0A AA 0A EA
7768:80 80 20 08 02 00 00 00 4C
7770:00 00 00 00 00 AA 28 00 5A
7778:02 02 08 20 80 00 00 00 EF
7780:55 55 57 57 57 5C 5C FC B1
7788:7D C3 00 00 00 00 00 00 27
7790:55 55 D5 D5 D5 35 35 3F C4
7798:FF C0 C0 C0 C0 C0 C0 FF 66
77A0:FF 00 00 00 00 00 00 FF 8F
77A8:FF 03 03 03 03 03 03 FF 13
77B0:AA 80 80 80 80 80 80 AA DE
77B8:AA 00 00 00 00 00 00 AA A7
77C0:AA 02 02 02 02 02 02 AA AC
77C8:66 99 66 99 66 99 66 99 B7
77D0:66 99 66 99 66 99 66 99 BF
77D8:66 99 66 99 66 99 66 99 C7
77E0:0C 0C FF 0C 0C FF 0C 0C 1E
77E8:30 30 FF 30 30 FF 30 30 11
77F0:C3 C3 FF C3 C3 FF C3 C3 58
77F8:00 00 00 00 00 00 00 00 E7
```

128 DOS Wedge

See instructions in article on page 42 before typing in.

```
1A00:AD 04 03 AE 05 03 8D C8 6F
1A08:1A 8E C9 1A A9 23 A2 1A 01
1A10:8D 04 03 8E 05 03 60 AD F7
1A18:C8 1A AE C9 1A 20 10 1A 35
1A20:4C 37 4D AD 00 02 A0 09 1F
1A28:D9 DE 1A F0 06 88 10 F8 BE
1A30:6C C8 1A 98 0A 48 20 6D B8
1A38:1A 68 A8 B9 CB 1A 48 B9 55
1A40:CA 1A 48 60 8A F0 20 A2 6A
1A48:03 A0 0C 20 BD FF A2 00 DC
1A50:20 68 FF A9 0F A2 08 A8 05
1A58:20 BA FF 20 C0 FF A9 0F B5
1A60:38 20 C3 FF 4C 37 4D A2 AD
1A68:E8 A0 1A D0 4F A2 FF 20 AE
1A70:80 03 F0 11 C9 22 F0 04 91
1A78:E8 9D 03 0C C8 B1 3D F0 22
1A80:04 C9 22 D0 F3 E8 A9 00 11
1A88:9D 03 0C 8D 00 FF 85 7A 2C
1A90:60 8A D0 08 8D 04 0C A9 70
1A98:2A 8D 03 0C A0 EE 2C A0 20
1AA0:F0 2C A0 8A 2C A0 EF A2 7B
1AA8:20 D0 07 A0 11 2C A0 14 9A
1AB0:A2 FE 8E 00 0C 8C 01 0C 68
1AB8:A2 00 A0 0C 86 3D 84 3E 83
1AC0:A9 22 8D 02 0C 4C DC 4D BC
1AC8:00 00 43 1A 43 1A 90 1A C4
1AD0:90 1A 9E 1A AA 1A A4 1A 6A
1AD8:A1 1A AD 1A 16 1A 40 3E 93
1AE0:24 2A 2F 25 5F 5E 3B 21 F5
1AE8:99 44 53 24 00 00 00 00 A7
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Vocab Builder

Article on page 23.

```
HR 5 SYS65517:IFPEEK(781)=40TH
     ENPRINT"{WHT}":GOTO10
KB 6 PRINT"{BLK}"
RX 10 PRINTCHR$(14):DIMA%(100)
      ,W$(100),D$(100),WR$(100
      ):D=0
PD 30 PRINT"{CLR}{7 DOWN}{RVS}
      ENTER YOUR NAME{OFF}":IN
      PUTNM$:IFLEN(NM$)=0THEN3
      0
EK 35 IFLEN(NM$)>13THENNM$=LEF
      T$(NM$,13)
PA 50 PRINT"{CLR}{RVS}"TAB(INT
      ((22-LEN(NM$))/2)-1)NM$;
      "'S{OFF}":PRINTTAB(3)"
      {RVS}VOCABULARY TEST
      {OFF}"
PH 55 N$="":A$="":B$="":DA$=""
      :X%=0:I=0:J=0:FORQ=1TO10
      0:A%(Q)=0:W$(Q)="":D$(Q)
      =""
KF 60 WR$(Q)="":NEXT:PRINTSPC(
      9);"{2 DOWN}MENU":PRINTS
      PC(2);"{2 DOWN}1) ENTER
      {SPACE}NEW WORDS"
JD 80 PRINTSPC(2);"{DOWN}2) ST
      UDY YOUR WORDS":PRINTSPC
      (2);"{DOWN}3) TAKE A TES
      T"
QC 100 PRINTSPC(2);"{DOWN}4) E
       ND":PRINT"{2 DOWN}{RVS}
        PRESS 1-4 TO CONTINUE
       {OFF}"
DX 120 GETZ$:C$=Z$:V=VAL(Z$):I
       FZ$=""OR(V<1ORV>4)THEN1
       20
GF 130 IFV=4THEN190
SR 140 INPUT"{2 DOWN}{RVS}D
       {OFF}ISK OR {RVS}T{OFF}
       APE";Z$:IFZ$<>"T"ANDZ$<
       >"D"THENPRINT"{4 UP}":G
       OTO130
QF 150 A$="{LEFT}{22 SPACES}":
       IFZ$="D"THEND=1
GQ 160 ONVGOTO230,810,810
FC 190 END
CH 230 PRINT"{CLR}{6 DOWN}ENTE
       R # OF NEW WORDS "
KS 235 INPUTX%:IFX%>100ORX%<1T
       HEN230
MB 250 FORI=1TOX%
BH 260 PRINT"{CLR}{3 DOWN}ENTE
       R WORD"I:INPUTW$(I)
```

```
GP 265 IFLEN(W$(I))=0THENPRINT
       "{6 UP}":GOTO260
CE 270 PRINT"{DOWN}ENTER DEFIN
       ITION":INPUTD$(I)
JS 275 IFLEN(D$(I))=0THENPRINT
       "{3 UP}":GOTO270
SD 280 NEXT
QA 300 PRINT"{CLR}":FORI=1TOX%
GA 310 PRINT" WORD"I"{LEFT}:";
       W$(I)
RB 320 PRINT"{DOWN} DEFINITION
       :"D$(I):PRINT"{DOWN}"
RC 330 IF(I/3)=INT(I/3)ANDX%=3
       THEN360
CF 332 IF(I/3)=INT(I/3)ANDX%<>
       ITHENGOSUB2000
FG 340 NEXT
PM 360 PRINT"{DOWN}{RVS}ANY CO
       RRECTIONS(Y/N)? {OFF}"
AR 380 GETZ$:IFZ$=""OR(Z$<>"Y"
       ANDZ$<>"N")THEN380
PG 390 IFZ$="Y"THEN420
FX 400 IFZ$="N"THEN520
GR 420 I=0:INPUT"{CLR}{4 DOWN}
       WHICH ENTRY";I
DH 435 IFI=0ORI>X%THENPRINT"
       {4 UP}":GOTO420
PX 440 PRINT"{2 DOWN}WORD"I"
       {LEFT}:";W$(I)
FF 450 PRINT"{DOWN}DEFINITION:
       ";D$(I)
HF 470 PRINT"{2 DOWN}ENTER WOR
       D"I:INPUTW$
CG 480 W$(I)=W$
CK 490 PRINT"{DOWN}ENTER DEFIN
       ITION":INPUTD$
CG 500 D$(I)=D$
MB 510 GOTO300
CB 520 INPUT"{2 DOWN}TODAY'S D
       ATE: ";DA$
KM 540 PRINT"{CLR}{8 DOWN}
       {5 SPACES}{RVS}PLEASE W
       AIT"
KE 550 PRINT"{2 DOWN}{RVS}I'M
       {SPACE}SAVING YOUR WORD
       S {OFF}{2 DOWN}"
HP 560 N$="TEST "+DA$
PD 565 IFD=1THENN$="TEST "+DA$
       +",S,W"
CC 570 OPEN15,8,15:OPEN1+D,1+7
       *D,1+D,N$:INPUT#15,A$,B
       $
PJ 572 IFA$="63"THENCLOSE(1+D)
       :CLOSE15:GOTO520
PG 575 IFA$<>"00"THENPRINTB$:F
       ORI=1TO3000:NEXT:CLOSE(
       1+D):CLOSE15:GOTO50
PG 580 PRINT#(1+D),X%:FORI=1TO
       X%
QJ 590 PRINT#(1+D),W$(I):PRINT
       #(1+D),D$(I)
AK 600 NEXTI:CLOSE(1+D):CLOSE1
       5:GOTO50
QP 810 PRINT"{CLR}{DOWN} {RVS}
       A TEST MADE JUST FOR
       {OFF}":PRINTTAB(INT((22
       -LEN(NM$))/2)-1)"{RVS}"
       NM$"{OFF}"
BB 820 PRINT"{2 DOWN}ENTER THE
        TEST DATE ":INPUTDA$
DQ 870 PRINT"{CLR}{5 DOWN}
       {5 SPACES}{RVS}PLEASE W
       AIT"
JH 880 PRINT"{DOWN}{2 SPACES}
       {RVS}LOADING THE WORDS
       {OFF}"
DQ 890 N$="TEST "+DA$
JJ 895 IFD=1THENN$="TEST "+DA$
       +",S,R"
AG 900 OPEN15,8,15:OPEN1+D,1+7
       *D,2*D,N$:INPUT#15,A$,B
       $
QP 902 IFA$="62"THENCLOSE1+D:C
       LOSE15:GOTO810
JC 905 IFA$<>"00"THENPRINTB$:F
       ORI=1TO3000:NEXT:CLOSE(
       1+D):CLOSE15:GOTO50
HA 910 INPUT#(1+D),X%:FORI=1TO
       X%
FJ 920 INPUT#(1+D),W$(I)
QF 930 IFW$(I)=""THENGOTO950
CC 940 GOTO960
QK 950 I=X%:GOTO970
MK 960 INPUT#(1+D),D$(I)
JD 970 NEXTI
MP 980 CLOSE(1+D):CLOSE15
EK 1030 FORI=1TOX%
CM 1040 A%=X%*RND(1)+1:IFI=1TH
        ENA%(I)=A%:GOTO1070
CP 1050 FORJ=1TOX%
CA 1060 IFA%=A%(J)THENJ=X%:NEX
        T:GOTO1040
CF 1065 NEXT
QH 1070 A%(I)=A%:NEXT
DX 1110 IFC$="2"THENGOTO1510
BF 1120 PRINT"{CLR}{DOWN}{RVS}
         ";NM$;"'S QUIZ {OFF}"
SS 1130 N%=0
BS 1140 FORI=1TOX%
BX 1150 PRINT"{2 DOWN}"
HH 1160 N=A%(I)
DG 1170 PRINT"DEFINITION: ";:P
        RINTD$(N)
MP 1180 PRINT"{DOWN}ENTER THE
        {SPACE}WORD":INPUTWO$:
        IFWO$=""THEN1180
EA 1190 IFWO$=W$(N)THENPRINT"
        {DOWN}CORRECT !":FORZ=
        1TO1500:NEXT:GOTO1230
RS 1200 PRINT"{DOWN}SORRY, THE
         WORD WAS:":PRINTW$(N)
        :FORZ=1TO1500:NEXT
FC 1210 N%=N%+1
KK 1220 WR$(N)=W$(N)
KH 1230 PRINT"{CLR}":NEXTI
AG 1250 PRINT"{6 DOWN}YOU GOT"
        N%"WRONG":PRINT"OUT OF
        "X%:GOSUB1900:IFN%=0TH
        EN2100
BA 1260 FORI=1TO3000:NEXT
GQ 1270 PRINT"{CLR}{2 DOWN}THE
         WORD(S) THAT YOU":PRI
        NT"GOT WRONG ARE:"
FF 1280 J=1:PQ=0:FORI=1TOX%
EF 1290 IFWR$(I)=""THEN1310
QB 1300 PQ=PQ+1:PRINT"{DOWN}";
        WR$(I)
XJ 1305 IFPQ<>5*JTHEN1310
QP 1306 J=J+1:PRINT"{DOWN}
        {RVS}ANY KEY TO CONTIN
        UE{OFF}":POKE198,0
AD 1307 GETZ$:IFZ$=""THEN1307
PF 1308 PRINT"{CLR}":NEXTI:GOT
        O50
FP 1310 NEXTI:GOSUB2000:GOTO50
RQ 1510 PRINT"{CLR}{RVS} HIT T
        HE BOOKS "
RH 1520 N%=0
JM 1530 FORI=1TOX%
EQ 1540 PRINT"{2 DOWN}":N=A%(I
        )
PA 1550 J%=1:J1%=1
QX 1560 PRINT"DEFINITION: ";:P
        RINTD$(N)
BC 1570 WO$="":PRINT"{DOWN}ENT
        ER THE WORD":INPUTWO$
DQ 1580 IFWO$=W$(N)THENPRINT"
        {DOWN}CORRECT !":GOTO1
        790
MP 1590 IFLEFT$(WO$,2)=LEFT$(W
        $(N),2)THENGOTO1610
DP 1600 GOTO1680
RA 1610 IFJ1%>1THENGOTO1750
HJ 1620 PRINT"{DOWN}YOU'RE CLO
        SE":PRINT"TRY AGAIN"
PG 1630 J1%=J1%+1
KP 1640 IFW$(N)=WR$(N)THENGOTO
        1670
SF 1650 WR$(N)=W$(N)
HJ 1670 PRINT"{2 DOWN}":GOTO15
        60
QB 1680 IFJ%>2THENGOTO1750
HR 1690 PRINT"{DOWN}SORRY, ";N
        M$:PRINT"YOU'RE NOT EV
        EN CLOSE":PRINT"{DOWN}
        TRY AGAIN"
CG 1691 FORZ=1TO2500:NEXT:PRIN
        T"{CLR}"
XA 1700 PRINT"{2 DOWN}":J%=J%+
        1
RP 1710 IFW$(N)=WR$(N)THENGOTO
        1740
RK 1720 WR$(N)=W$(N)
QD 1740 GOTO1560
FR 1750 PRINT"{DOWN}SORRY, THE
         WORD WAS:":PRINT"
        {DOWN} "W$(N):FORZ=1TO
        3000:NEXT:PRINT"{CLR}"
CS 1760 IFW$(N)=WR$(N)THENGOTO
        1790
SR 1770 WR$(N)=W$(N)
EJ 1780 N%=N%+1
BA 1790 FORTD=1TO1500:NEXT:PRI
        NT"{CLR}":NEXTI:GOTO12
        50
XJ 1900 IFN%=0THENPRINT"
        {2 DOWN}{RVS}GREAT JOB
        ":RETURN
BD 1910 N=10-(N%/X%*10)
PS 1920 IFN<6.5THENPRINT"
        {DOWN}BETTER STUDY MOR
        E":RETURN
KG 1930 IFN<8THENPRINT"{DOWN}A
         LITTLE MORE STUDY":PR
        INT"NEEDED":RETURN
FC 1940 IFN<9THENPRINT"{DOWN}N
        OT BAD":RETURN
ED 1950 IFN<9.5THENPRINT"
        {DOWN}GOOD JOB":RETURN
HH 1960 IFN<10THENPRINT"{DOWN}
        VERY GOOD"RETURN
DQ 1970 RETURN
BC 2000 PRINT"{DOWN}{RVS} ANY
        {SPACE}KEY TO CONTINUE
        {2 SPACES}"
QA 2001 GETMT$:IFMT$=""THEN200
        1
PX 2002 PRINT"{CLR}":RETURN
CG 2100 GOSUB2000:GOTO50
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

64 Mode Speed-Up

Article on page 44.

```
XB 10 PRINT"{CLR}{WHT}SPEED UP
      - 64 MODE ONLY"
KQ 20 FORI=49152TO49258:READX:
      C=C+X:POKEI,X:NEXT
BQ 30 IFC<>12470THENPRINT"DATA
       ERROR":END
ER 40 SYS 49152
XA 50 DATA 120,173,105,192,141
      ,18,208,173
GX 60 DATA 17,208,41,127,141,1
      7,208,169
```

```
RK 70 DATA 129,141,26,208,169,
      192,160,32
QM 80 DATA 141,21,3,140,20,3,8
      8,96
EP 90 DATA 173,25,208,141,25,2
      08,41,1
SQ 100 DATA 208,3,76,49,234,17
      3,18,208
JA 110 DATA 205,106,192,176,14
      ,172,106,192
MX 120 DATA 140,18,208,169,0,1
      41,48,208
XX 130 DATA 76,78,192,172,105,
      192,140,18
HP 140 DATA 208,169,1,141,48,2
      08,104,168
QD 150 DATA 104,170,104,64,120
      ,169,234,141
MB 160 DATA 21,3,169,49,141,20
      ,3,169
EP 170 DATA 0,141,48,208,141,2
      6,208,88
FS 180 DATA 96,50,250
```

Printer Wedge

Article on page 54.

Program 1: Printer Wedge— Main Program

```
ED 10 BS=1358:S0=160:S1=32:D0=
      209:D1=81:M=1:MF(0)=1:FO
      RJ=1TO6:M=M*2:MF(J)=M:NE
      XT
AQ 20 PRINTCHR$(147)CHR$(5)"
      {3 SPACES}PRINTER WEDGE
      {SPACE}CHARACTER DESIGNE
      R"
HJ 30 PRINTCHR$(17)"MOVE WITH
      {SPACE}CURSOR KEYS"
JE 40 PRINT"PLACE DOT WITH F7"
      :PRINT"ERASE DOT WITH F8
      "
XJ 50 PRINTCHR$(17)"ASSIGN CHA
      RACTER WITH F1"
EC 60 FORI=1TO5:PRINTCHR$(17):
      NEXT:PRINT"ASSIGN CHARAC
      TER STRING WITH F3"
EK 70 PRINT"ACTIVATE WEDGE WIT
      H F2"
FC 80 CH=S0:FORI=0TO9:FORJ=0TO
      6:GOSUB110:AR(I,J)=0:NEX
      TJ:NEXTI
MQ 90 I=0:J=0:CH=S1:GOSUB110:G
      OTO140
RS 100 I=0:J=0:CH=S1:GOSUB110:
      GOTO140
CF 110 X=BS+I+J*40:POKEX,CH:PO
      KEX+54272,1:RETURN
DQ 120 GETZ$:IFZ$=""THEN120
SS 130 RETURN
EQ 140 GOSUB120:DC=0:REM *CHAR
      ACTER EDITOR*
MR 150 IFZ$=CHR$(29)THENDC=1
EG 160 IFZ$=CHR$(157)THENDC=2
GS 170 IFZ$=CHR$(17)THENDC=3
KA 180 IFZ$=CHR$(145)THENDC=4
SG 190 IFDC>0THENGOSUB370:GOSU
      B380
AQ 200 ONDCGOSUB290,310,330,35
      0
JC 210 GOSUB370:IFDC>0ANDLC=S0
      THENCH=S1:GOSUB110
HS 220 IFDC>0ANDLC=D0THENCH=D1
      :GOSUB110
SJ 230 IFZ$=CHR$(136)THENAR(I,
      J)=1:CH=D1:GOSUB110
CS 240 IFZ$=CHR$(140)THENAR(I,
      J)=0:CH=S1:GOSUB110
JE 250 IFZ$=CHR$(133)THEN410
AA 260 IFZ$=CHR$(137)THENPRINT
      CHR$(147)CHR$(17)"PLEAS
      E WAIT":GOTO660
SS 270 IFZ$=CHR$(134)THEN550
AE 280 GOTO140
EH 290 I=I+1:IFI>9THENI=0
HJ 300 RETURN
QP 310 I=I-1:IFI<0THENI=9
XM 320 RETURN
JF 330 J=J+1:IFJ>6THENJ=0
HP 340 RETURN
GM 350 J=J-1:IFJ<0THENJ=6
XR 360 RETURN
CC 370 LC=PEEK(BS+I+J*40):RETU
      RN
JB 380 IFLC=S1THENCH=S0
BK 390 IFLC=D1THENCH=D0
EA 400 GOSUB110:RETURN
HA 410 FORI=0TO9:REM *COMPUTE
      {SPACE}CHARACTER*
GR 420 CL(I)=128:FORJ=0TO6:CL(
      I)=CL(I)+AR(I,J)*MF(J):
      NEXTJ:NEXTI
HR 430 PRINT"ASSIGN CHARACTER
      {SPACE}TO WHICH KEY?"
BM 440 GOSUB120:Z=ASC(Z$):B=30
      00+Z*10:REM *PRINT DATA
       STATEMENTS*
ER 450 PRINTCHR$(147)CHR$(17)C
      HR$(17)CHR$(17)B"DATA"Z
      "{4 SPACES}";
AF 460 FORI=0TO9:IFCL(I)=128AN
      DCL(I+1)=128THEN480
HC 470 PRINT","CL(I)CHR$(157);
      :NEXTI
MF 480 PRINT:FORDX=1TO5:PRINT3
      000+Z*10+DX:NEXT
DK 490 PRINT"RUN"CHR$(19);:FOR
      J=1TO3:PRINTCHR$(17);:N
      EXT
JP 500 FORJ=1TO14:PRINTCHR$(29
      );:NEXT:PRINT","ICHR$(1
      9)
EF 510 IFI>0THEN540
JG 520 PRINTCHR$(19);:FORJ=1TO
      3:PRINTCHR$(17);:NEXT
SQ 530 FORJ=1TO20:PRINTCHR$(32
      );:NEXT
CB 540 POKE198,9:FORI=0TO8:POK
      E631+I,13:NEXT:END
RA 550 PRINTCHR$(147)"ENTER CH
      ARACTER STRING":INPUTA$
      :L=LEN(A$):IFL<1ORL>39T
      HEN550
RK 560 PRINT"ASSIGN TO WHICH K
      EY?":GOSUB120:DX=0:Z=AS
      C(Z$):I=1:B=3000+Z*10+D
      X
ED 570 PRINTCHR$(147)CHR$(17)C
      HR$(17)CHR$(17)B"DATA"Z
      "{4 SPACES},15,17";:LX=
      2
CM 580 PRINT","ASC(MID$(A$,I,1
      ))CHR$(157);:LX=LX+1:I=
      I+1
RQ 590 IFLX>9ANDI<=LTHENLX=0:D
      X=DX+1:B=3000+Z*10+DX:G
      OSUB650
MJ 600 IFI<=LTHEN580
RX 610 PRINT:FORDX=DX+1TO5:PRI
      NT3000+Z*10+DX:NEXT
EA 620 PRINT"RUN"CHR$(19);:FOR
      J=1TO3:PRINTCHR$(17);:N
      EXT
CJ 630 FORJ=1TO14:PRINTCHR$(29
      );:NEXT:PRINT","L+2
RM 640 PRINTCHR$(19):POKE198,9
      :FORI=0TO8:POKE631+I,13
      :NEXT:END
PJ 650 PRINTCHR$(13)B"DATA"ASC
      (MID$(A$,I,1))CHR$(157)
      ;:LX=LX+1:I=I+1:RETURN
JX 660 FORI=57344TO57452:REM P
      OKE HI CODE
SA 670 READA:POKEI,A:NEXT
GB 680 PRINTCHR$(147)"
      {2 SPACES}WHERE SHOULD
      {SPACE}PWEDGE BE LOCATE
      D?"
AG 690 PRINTCHR$(17)"854
      {3 SPACES}= CASSETTE BU
      FFER"
PE 700 PRINT"679{3 SPACES}= UN
      USED RAM (MAYBE)"
XD 710 PRINT"49152 = RESERVED
      {SPACE}FOR ML PROGRAMS"
BK 720 PRINTCHR$(17)"LOCATION
      {3 SPACES}854"CHR$(157)
      CHR$(157)CHR$(157)CHR$(
      157)CHR$(157);
CR 730 INPUTBA:FORI=BATOBA+59:
      READA:POKEI,A:NEXT
FG 740 IFBA=854THEN830
SP 750 P=BA+60:GOSUB990:POKEBA
      +10,L:POKEBA+11,H:POKEB
      A+49,L:POKEBA+50,H
CR 760 P=BA+61:GOSUB990:POKEBA
      +13,L:POKEBA+14,H:POKEB
      A+52,L:POKEBA+53,H
JS 770 P=BA+62:GOSUB990:POKEBA
      +1,L:POKEBA+2,H:POKEBA+
      55,L:POKEBA+56,H
RG 780 POKE57354,L:POKE57355,H
      :POKE57449,L:POKE57450,
      H
JK 790 P=BA+64:GOSUB990:POKEBA
      +39,L:POKEBA+40,H:POKE5
      7409,L:POKE57410,H
PQ 800 POKE57425,L:POKE57426,H
      :POKE57437,L:POKE57438,
      H
RS 810 POKE57443,L:POKE57444,H
MK 820 P=BA+25:GOSUB990:POKE57
      395,L:POKE57396,H
BD 830 P=BA:GOSUB990:POKE806,L
      :POKE807,H
EH 840 REM POKE CHARACTER DATA
JF 850 I=57601:CT=0:Z=I+256
HJ 860 READC:IFC=0THENPOKEZ-25
      7,CT:POKEZ+255,0:GOTO91
      0
HR 870 READL:IF(L+I+3)>=ZTHEN8
      90
KX 880 POKEI,C:POKEI+1,L:FORK=
      1TOL:READA:POKEI+1+K,A:
      NEXT:I=I+L+2:CT=CT+1:GO
      TO860
SG 890 POKEZ-257,CT:CT=1:I=Z:Z
      =I+256:POKEI,C:POKEI+1,
      L
CQ 900 FORK=1TOL:READA:POKEI+1
      +K,A:NEXT:I=I+L+2:GOTO8
      60
KM 910 PRINTCHR$(147)"PWEDGE I
      S NOW INSTALLED."
SD 920 PRINTCHR$(17)"DO YOU WA
      NT A PRINTER TEST?"
DD 930 GOSUB120:IFZ$<>"Y"THENE
      ND
QC 940 OPEN4,4,7:PRINT#4,"THE
      {SPACE}QUICK BROWN FOX
      {SPACE}JUMPED OVER THE
      {SPACE}LAZY DOG."
GE 950 FORI=33TO64:PRINT#4,CHR
      $(I);:NEXT:PRINT#4
AQ 960 FORI=65TO96:PRINT#4,CHR
      $(I);:NEXT:PRINT#4
JP 970 FORI=193TO218:PRINT#4,C
      HR$(I);:NEXT:PRINT#4:CL
      OSE4
GD 980 END
BD 990 H=INT(P/256):L=P-H*256:
      RETURN
BD 1000 REM *HI CODE*
```

```
BA 1010 DATA 172,0,225,192,0,2
        40,32,162,1,173
DX 1020 DATA 148,3,221,0,225,2
        40,36,232,138,24
EA 1030 DATA 125,0,225,170,232
        ,136,208,237,238,2
RK 1040 DATA 224,238,14,224,23
        8,22,224,208,217,169
GQ 1050 DATA 225,141,2,224,141
        ,14,224,141,22,224
GG 1060 DATA 76,111,3,173,2,22
        4,141,70,224,141
DD 1070 DATA 79,224,169,8,141,
        150,3,232,189,0
KM 1080 DATA 225,141,109,224,1
        60,1,232,189,0,225
QR 1090 DATA 153,150,3,232,200
        ,206,109,224,208,243
PG 1100 DATA 169,128,153,150,3
        ,200,169,15,153,150
GX 1110 DATA 3,200,169,17,141,
        148,3,208,186
QB 1200 REM *LO CODE*
GQ 1210 DATA 141,148,3,165,154
        ,201,4,208,45,142
RD 1220 DATA 146,3,140,147,3,1
        20,165,1,41,253
KM 1230 DATA 133,1,76,0,224,16
        5,1,9,2,133
PF 1240 DATA 1,88,192,0,240,12
        ,162,0,189,150
PH 1250 DATA 3,32,202,241,232,
        136,208,246,174,146
AJ 1260 DATA 3,172,147,3,173,1
        48,3,76,202,241
RD 10000 DATA 0
```

Program 2: Character Set 1

```
MQ 2999 REM *NON PROPORTIONAL
        {SPACE}SPACED- G, P AN
        D Q MODIFIED*
BB 3340 DATA 34,5,128,135,128,
        135,128,71,5,136,212
EP 3710 DATA 212,212,184,80,5,
        252,148,148,148,136
PR 3800 DATA 81,5,184,196,212,
        164,216
```

Program 3: Character Set 2

```
SK 3000 REM *PROPORTIONAL SPAC
        ED CHARACTERS WITHOUT
        {SPACE}DESCENDERS*
JK 3010 DATA34,4,128,135,128,1
        35,39,3,132,130
DQ 3020 DATA129,66,4,255,196,1
        96,184,67,4,184
GD 3030 DATA196,196,196,68,4,1
        84,196,196,255,69
DD 3040 DATA4,184,212,212,152,
        71,5,136,212,212
PA 3050 DATA212,184,72,4,255,1
        36,132,248,73,3
BD 3060 DATA196,253,192,74,4,1
        60,192,192,189,75
MJ 3070 DATA4,255,144,168,196,
        76,3,129,255,192
CM 3080 DATA77,7,252,132,132,1
        84,132,132,248,79
SJ 3090 DATA4,184,196,196,184,
        80,5,252,148,148
AE 3100 DATA148,136,81,5,184,1
        96,212,164,220,82
QK 3110 DATA4,252,136,132,132,
        86,5,156,160,192
FC 3120 DATA160,156,87,7,188,1
        92,192,184,192,192
QS 3130 DATA252,89,4,204,208,2
        08,188,201,3,193
MK 3140 DATA255,193,205,7,255,
        130,132,152,132,130
AQ 3150 DATA255,215,7,255,160,
        144,140,144,160,255
```

Program 4: Character Set 3

```
FR 3000 REM *PROPORTIONAL SPAC
        ED CHARACTERS WITH DES
        CENDERS*
HC 3010 DATA17,5,129,133,148,2
        08,192,33,1,175
HM 3020 DATA34,3,135,128,135,3
        7,6,163,147,136
FS 3030 DATA132,178,177,39,3,1
        32,130,129,40,3
PQ 3040 DATA140,146,161,41,3,1
        61,146,140,44,2
MD 3050 DATA192,176,46,2,176,1
        76,47,6,160,144
SQ 3060 DATA136,132,130,129,48
        ,4,158,169,165,158
KH 3070 DATA49,3,162,191,160,5
        0,4,162,177,169
MH 3080 DATA166,51,4,146,161,1
        65,154,52,4,140
PF 3090 DATA138,137,191,53,4,1
        67,165,165,153,54
FQ 3100 DATA4,158,169,169,144,
        55,4,177,137,133
SJ 3110 DATA131,56,4,154,165,1
        65,154,57,4,134
AC 3120 DATA169,169,158,58,1,1
        48,59,2,192,180
XC 3130 DATA60,4,136,148,162,1
        62,61,4,148,148
XQ 3140 DATA148,148,62,4,162,1
        62,148,136,63,5
KG 3150 DATA130,129,217,133,13
        0,65,5,152,164,164
FH 3160 DATA156,160,66,4,191,1
        64,164,152,67,4
KX 3170 DATA152,164,164,164,68
        ,4,152,164,164,191
EM 3180 DATA69,5,152,172,172,1
        72,136,70,4,190
PB 3190 DATA137,129,130,71,4,1
        36,212,212,184,72
CF 3200 DATA4,191,132,132,184,
        73,2,189,160,74
RS 3210 DATA4,160,192,192,189,
        75,4,191,136,148
QC 3220 DATA160,76,3,129,191,1
        60,77,7,188,132
SJ 3230 DATA132,184,132,132,18
        4,78,5,132,184,132
EE 3240 DATA132,184,79,5,152,1
        64,164,164,152,80
FK 3250 DATA4,252,148,148,136,
        81,5,136,148,148
HB 3260 DATA248,192,82,4,188,1
        36,132,132,83,4
HF 3270 DATA200,212,212,160,84
        ,3,132,190,164,85
SJ 3280 DATA5,156,160,160,156,
        160,86,5,140,144
JC 3290 DATA160,144,140,87,7,1
        88,160,160,156,160
SK 3300 DATA160,156,88,5,164,1
        48,136,148,164,89
EX 3310 DATA4,140,208,208,188,
        90,4,164,180,172
RC 3320 DATA164,193,5,156,162,
        161,159,160,194,4
HK 3330 DATA191,165,165,186,19
        5,4,156,162,161,161
EP 3340 DATA196,5,191,161,161,
        162,156,197,4,191
CG 3350 DATA165,165,161,198,4,
        191,133,133,129,199
KB 3360 DATA5,156,162,169,169,
        144,200,4,191,132
CS 3370 DATA132,191,201,3,161,
        191,161,202,4,144
ME 3380 DATA160,161,159,203,4,
        191,132,138,177,204
KG 3390 DATA4,191,160,160,160,
        205,7,191,129,129
XD 3400 DATA142,129,129,190,20
        6,5,129,190,129,129
AC 3410 DATA190,207,4,158,161,
        161,158,208,4,191
QF 3420 DATA137,137,134,209,5,
        156,162,169,145,174
RG 3430 DATA210,5,191,137,137,
        150,160,211,4,162
EX 3440 DATA165,165,152,212,5,
        129,129,191,129,129
FC 3450 DATA213,5,159,160,160,
        159,160,214,5,143
GC 3460 DATA144,160,144,143,21
        5,7,159,160,160,156
PF 3470 DATA160,160,191,216,5,
        177,138,132,138,177
PH 3480 DATA217,5,129,130,188,
        130,129,218,5,161
CB 3490 DATA177,169,165,163
```

Program 5: Graphics To Text Conversion

```
CE 3000 REM *CHARACTER STRINGS
         FOR CONVERTING GRAPHI
        CS SYMBOLS TO TEXT*
BB 3010 REM *WHEN PRINTING A L
        ISTED FILE*
RX 3020 DATA5,7,15,17,91,87,72
        ,84,93,17
CQ 3030 DATA8,15,17,91,68,79,8
        7,78,93,18
KG 3040 DATA7,15,17,91,82,86,8
        3,93,19,8
DE 3050 DATA15,17,91,72,79,77,
        69,93,28,7
FP 3060 DATA15,17,91,82,69,68,
        93,29,9,15
PD 3070 DATA17,91,82,73,71,72,
        84,93,30,7
MC 3080 DATA15,17,91,71,82,78,
        93,31,7,15
PE 3090 DATA17,91,66,76,85,93,
        129,7,15,17
XB 3100 DATA91,60,49,62,93,133
        ,6,15,17,91
BP 3110 DATA70,49,93,134,6,15,
        17,91,70,51
GR 3120 DATA93,135,6,15,17,91,
        70,53,93,136
HB 3130 DATA6,15,17,91,70,55,9
        3,137,6,15
XF 3140 DATA17,91,70,50,93,138
        ,6,15,17,91
KE 3150 DATA70,52,93,139,6,15,
        17,91,70,54
HS 3160 DATA93,140,6,15,17,91,
        70,56,93,144
SJ 3170 DATA7,15,17,91,66,76,7
        5,93,145,6
KQ 3180 DATA15,17,91,85,80,93,
        146,7,15,17
CD 3190 DATA91,79,70,70,93,147
        ,7,15,17,91
FD 3200 DATA67,76,82,93,149,7,
        15,17,91,60
FK 3210 DATA50,62,93,150,7,15,
        17,91,60,51
XX 3220 DATA62,93,151,7,15,17,
        91,60,52,62
AM 3230 DATA93,152,7,15,17,91,
        60,53,62,93
JC 3240 DATA153,7,15,17,91,60,
        54,62,93,154
BB 3250 DATA7,15,17,91,60,55,6
        2,93,155,7
FA 3260 DATA15,17,91,60,56,62,
        93,156,7,15
MR 3270 DATA17,91,80,85,82,93,
        157,8,15,17
HD 3280 DATA91,76,69,70,84,93,
```

```
        158,7,15,17
SQ 3290 DATA91,89,69,76,93,159
        ,7,15,17,91
RH 3300 DATA67,89,78,93
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Hi-Res Print

See instructions in article on page 24 before typing in.

Program 1: HRPRINT

```
C000:20 F1 B7 8E 13 03 4C 15 40
C008:C0 20 FD AE 20 9E AD 20 93
C010:CE B1 A5 65 60 20 09 C0 C6
C018:8D FA 02 A5 64 8D FB 02 0D
C020:20 09 C0 8D FC 02 A5 64 84
C028:8D FD 02 20 09 C0 8D FE 97
C030:02 A5 64 8D FF 02 60 A0 EA
C038:00 A9 00 99 A7 02 C8 C0 55
C040:51 D0 F8 20 F1 B7 8E 34 7F
C048:03 20 F1 B7 8E 36 03 20 80
C050:F1 B7 8E 35 03 20 FD AE 21
C058:20 9E AD A5 0D D0 13 20 93
C060:DD BD A0 00 B9 00 01 C9 ED
C068:00 F0 26 99 A7 02 C8 4C A7
C070:64 C0 20 A3 B6 A0 00 B1 7C
C078:64 85 02 C8 B1 64 85 FB 80
C080:C8 B1 64 85 FC A0 00 B1 D3
C088:FB 99 A7 02 C8 C4 02 D0 B1
C090:F6 A9 A7 85 F9 A9 02 85 45
C098:FA AD 13 03 C9 01 D0 12 9B
C0A0:AD 34 03 85 FB A9 10 85 EA
C0A8:FC 20 B5 C0 AD FC 03 8D 68
C0B0:34 03 4C D1 C0 A9 00 8D ED
C0B8:FD 03 8D FC 03 A2 08 46 74
C0C0:FB 90 03 18 65 FC 6A 6E A8
C0C8:FC 03 CA D0 F2 8D FD 03 BC
C0D0:60 AD 35 03 85 FB A9 A0 D4
C0D8:85 FC 20 B5 C0 18 AD FC 7A
C0E0:03 6D FC 03 8D FC 03 AD 23
C0E8:FD 03 6D FD 03 8D FD 03 05
C0F0:AD FC 03 8D FE 03 AD FD 1F
C0F8:03 8D FF 03 AD 36 03 85 61
C100:FB A9 08 85 FC 20 B5 C0 D9
C108:18 AD FC 03 6D FE 03 8D CD
C110:FC 03 AD FD 03 6D FF 03 39
C118:8D FD 03 18 AD FC 03 6D 98
C120:FA 02 8D FC 03 AD FD 03 F0
C128:6D FB 02 8D FD 03 AD FC CE
C130:03 8D 37 03 AD FD 03 8D A8
C138:38 03 A0 FF C8 B1 F9 C9 77
C140:00 D0 01 60 8D 39 03 C9 3F
C148:40 90 15 C9 60 F0 0B B0 DC
C150:09 38 E9 40 8D 39 03 4C 4B
C158:60 C1 38 E9 80 8D 39 03 D1
C160:AD 39 03 85 FB A9 08 85 DD
C168:FC 20 B5 C0 18 AD FC 03 A9
C170:6D FC 02 8D 3A 03 AD FD 3A
C178:03 6D FD 02 8D 3B 03 A2 BA
C180:FF 8C F9 02 A0 00 E8 8E EB
C188:F8 02 18 AD F8 02 6D 37 C8
C190:03 85 FD A9 00 6D 38 03 7A
C198:85 FE 18 AD F8 02 6D 3A 61
C1A0:03 85 FB A9 00 6D 3B 03 50
C1A8:85 FC B1 FB 91 FD E0 07 71
C1B0:D0 D4 AC F9 02 AD 35 03 3B
C1B8:85 FB A9 28 85 FC 20 B5 CB
C1C0:C0 18 AD FC 03 6D 36 03 6D
C1C8:8D FC 03 AD FD 03 69 00 5C
C1D0:8D FD 03 18 AD FC 03 6D 51
C1D8:FE 02 85 FB AD FD 03 6D A5
C1E0:FF 02 85 FC AD 13 03 C9 EE
C1E8:01 D0 08 AD 34 03 11 FB C8
C1F0:4C F6 C1 AD 34 03 91 FB 38
C1F8:18 A9 08 6D 37 03 8D 37 E2
C200:03 A9 00 6D 38 03 8D 38 69
C208:03 4C 3C C1 00 00 00 00 C5
```

Program 2: CHARSETMAKER

```
JS 100 REM CREATE CHARACTER SE
       T FOR
BG 110 REM 'HI-RES PRINT' ML R
       OUTINE
MG 120 PRINT CHR$(142):REM USE
        CHR$(14) FOR{2 SPACES}
       LOWERCASE
AS 130 PRINT CHR$(8):REM DISAB
       LE CASE CHANGE FROM KEY
       BOARD
QD 140 INPUT "LOCATION OF CHAR
       ACTER SET";N
RB 150 A=N-INT(N/256)*256: B=I
       NT(N/256)
XC 160 INPUT "FILENAME";F$
EC 170 IF LEN(F$)=0 THEN 160
DA 180 PRINT "WORKING..."
JE 190 REM COPY FIRST 64 CHARA
       CTERS FROM ROM CHARACTE
       R SET
SS 200 POKE 56334, PEEK(56334)
        AND 254
AP 210 POKE 1, PEEK(1) AND 251
KK 220 FOR J=0 TO 511
CK 230 POKE J+N, PEEK(53248+J)
QF 240 NEXT J
FG 250 POKE 1, PEEK(1) OR 4
PF 260 POKE 56334, PEEK(56334)
        OR 1
HF 270 CLOSE 15:OPEN 15, 8, 15
       , "I0"
FA 280 GOSUB 410
KD 290 OPEN 2, 8, 2, F$ + ",P,
       W"
JA 300 GOSUB 410
HQ 310 PRINT#2, CHR$(A);:PRINT
       #2, CHR$(B);
AC 320 GOSUB 410
SS 330 FOR J=0 TO 511
PH 340 PRINT#2, CHR$(PEEK(J+N)
       );
DQ 350 NEXT J
AH 360 GOSUB 410
XD 370 CLOSE 2:CLOSE 15
BJ 380 PRINT CHR$(9):REM ENABL
       E CASE CHANGE FROM KEYB
       OARD
PS 390 END
DQ 400 REM CHECK DISK DRIVE ER
       ROR STATUS
QR 410 INPUT#15,EX,EX$,TR,SE
BE 420 IF EX=0 THEN RETURN
RF 430 PRINT CHR$(18) "DISK ER
       ROR"
SB 440 PRINT EX;EX$;TR;SE
AB 450 GOTO 370
```

Program 3: Demo

```
KB 100 REM 'HRPRINT' DEMO
EQ 110 IF A=0 THEN A=1:LOAD "H
       RPRINT", 8, 1
DP 120 IF A=1 THEN A=2:LOAD "H
       RCHARSET", 8, 1
HD 130 POKE 53280,15
HC 140 REM SET SYS ADDRESSES
SA 150 IN=49152: PR=49207
JG 160 REM BI ARRAY IS USED TO
        DRAW CURVE
RK 170 FOR J=0 TO 7:BI(J)=2↑J:
       NEXT
KF 180 REM BITMAP AT 8192
MP 190 BASE=8192
JB 200 POKE 53272,PEEK(53272)
       {SPACE}OR 8
XF 210 REM HI-RES
QQ 220 POKE 53265, PEEK(53265)
        OR 32
BM 230 REM INITIALIZE MULTI/RV
       S(2),BITMAP AT 8192,CHA
       RSET AT 16384,COLOR AT
       {SPACE}1024
RG 240 SYS IN, 2, 8192, 16384,
        1024
BK 250 REM CLEAR SCREEN AND DR
       AW CURVE
MA 260 F$="":FOR J=1 TO 40:F$=
       F$+CHR$(32):NEXT
RH 270 FOR J=0 TO 24:SYS PR, 1
       5, 0, J, F$:NEXT
PJ 280 FOR Y=0 TO 199 STEP .5
XX 290 X = INT(160+40 * SIN(Y/
       10))
DG 300 BY = BASE+40 * (Y AND 2
       48) + (Y AND 7)+(X AND
       {SPACE}504)
HM 310 POKE BY, PEEK(BY) OR (B
       I(NOT X AND 7))
AB 320 NEXT Y
MX 330 REM INITIALIZE HIRES(1)
       ,BITMAP AT 8192,CHARSET
        AT 16384,COLOR AT 1024
GD 340 SYS IN, 1, 8192, 16384,
        1024
QH 350 REM PRINT MESSAGES ON H
       I-RES SCREEN
HM 360 SYS PR, 6, 1, 1, "PRINT
        HR DEMO"
DJ 370 SYS PR, 2, 1, 2, "(C) 1
       987 COMPUTE!"
RQ 380 SYS PR, 1, 17, 21, "HIT
        ANY KEY TO EXIT"
HR 390 SYS PR, 8, 17, 5, "YOU
       {SPACE}CAN PRINT NORMAL
       LY"
AM 400 REM INITIALIZE FOR REVE
       RSE
DJ 410 SYS IN, 2, 8192, 16384,
        1024
EM 420 SYS PR, 5, 17, 13, "OR
       {SPACE}IN REVERSE LETTE
       RS"
JA 430 REM NORMAL CHARACTERS A
       GAIN
RM 440 SYS IN, 1, 8192, 16384,
        1024
CG 450 SYS PR, 14, 2, 9, "TIME
       R:"
HM 460 SYS PR, 14, 2, 17, "SCO
       RE:"
CS 470 REM UPDATE SCORE AND TI
       MER
PD 480 FOR J=0 TO 10000
EX 490 SYS PR, 0, 8, 9, TI
FP 500 SYS PR, 0, 8, 17, J
BB 510 GET X$
JD 520 IF X$="" THEN NEXT J
AD 530 REM BACK TO NORMAL TEXT
        SCREEN
EP 540 POKE 53265,27:POKE 5327
       2,21
PS 550 PRINT CHR$(147):END
```

Program 4: Relocator

```
KJ 100 REM THIS PROGRAM RELOCA
       TES THE
PJ 110 REM 'HRPRINT' ML ROUTIN
       E AND WRITES
CA 120 REM THE RELOCATED CODE
       {SPACE}TO DISK
QS 130 IF A=0 THEN A=1:LOAD "H
       RPRINT",8,1
```

```
HJ 140 INPUT "NEW LOCATION FOR
       HRPRINT";N
JD 150 INPUT "NEW FILENAME FOR
       HRPRINT";F$
EX 160 A=INT(N/256):B=N-256*A
XB 170 PRINT "WORKING..."
RX 180 CLOSE 15:OPEN 15,8,15,"
       I0"
PM 190 GOSUB 410
GK 200 OPEN 2, 8, 2, F$ + ",P,
       W"
AR 210 GOSUB 410
RJ 220 PRINT#2, CHR$(B);CHR$(A
       );
JQ 230 GOSUB 410
XD 240 FOR J=49152 TO 49675
QR 250 P=PEEK(J)
QK 260 IF P<>76 AND P<>108 AND
       P<>32 THEN PRINT#2,CHR
       $(P);:NEXT J
HX 270 Z=PEEK(J+2)
JS 280 IF Z<192 OR Z>194 THEN
       {SPACE}PRINT#2,CHR$(P);
       :NEXT J
QJ 290 X=49152-N
HX 300 Y=PEEK(J+1)
AS 310 A=Y+Z*256-X
JQ 320 Z=INT(A/256)
PE 330 Y=A-256*Z
KP 340 PRINT#2, CHR$(P);CHR$(Y
       );CHR$(Z);
EF 350 J=J+2
XR 360 NEXT J
FK 370 PRINT "NEW SYS VALUE FO
       R 'IN':";N
KF 380 PRINT "NEW SYS VALUE FO
       R 'PR':";N+55
DD 390 CLOSE 2:CLOSE 15:END
DQ 400 REM CHECK DISK DRIVE ER
       ROR STATUS
QR 410 INPUT#15,EX,EX$,TR,SE
BE 420 IF EX=0 THEN RETURN
RF 430 PRINT CHR$(18) "DISK ER
       ROR"
SB 440 PRINT EX;EX$;TR;SE
SD 450 GOTO 390
```

BEFORE TYPING . . .

Before typing in programs, please refer to "How to Type In COMPUTE!'s GAZETTE Programs," elsewhere in this issue.

Power BASIC: 128
Instant Keywords

Article on page 43.

```
FG 100 PRINT"{CLR}":S=INPUT("]
       300"):={B}15:FORD=STOS+
       176:READ A$:POKE D,INPU
       T(A$):NEXT
HB 110 M=S+22:V%=M/256:V=M-256
       *V%:POKE S+12,V%:POKE S
       +14,V:M=D-52:V%=M/256:V
       =M-256*V%
FM 120 POKE S+65,V%:POKE S+64,
       V: SYS(S):PRINT"
       {2 DOWN}{RVS}{WHT}INSTA
       NT KEYWORDS ACTIVATED!
RK 130 PRINT"{DOWN}{RVS}TO DE/
       ACTIVATE SYS"S:NEW
JG 140 DATA A9,C8,CD,37,03,F0,
       04,A2,05,D0,04,A9,13,A2
       ,16,8D,37,03,8E,36
KE 150 DATA 03,60,EA,48,A6,F4,
       F0,04,68,4C,05,C8,A4,D4
       ,AD,3E,03,85,24,AD
AR 160 DATA 3F,03,85,25,B1,24,
       EA,38,E9,41
QA 170 DATA C9,1A,B0,E6,A2,02,
       E4,D3,D0,02,69,19,A8,B9
       ,7D,13,C9,27,B0,09
EE 180 DATA 69,7E,AA,A0,09,A9,
       46,D0,05,AA,A0,17,A9,44
       ,84,24,85,25,A0,00
MK 190 DATA CA,10,0F,B1,24,48,
       E6,24,D0,02,E6,25,68,10
       ,F4,30,EF,C8,B1,24
BH 200 DATA 30,05,20,0C,56,D0,
       F6,29,7F,20,0C,56,68,60
       ,EA,0B,18,C7,0D,D5
KA 210 DATA 81,89,D2,85,CF,E5,
       EC,CA,82,A3,99,E3,8E,07
       ,A7,FB,87,FD,97,DE
QD 220 DATA C8,C4,19,E7,0F,0A,
       03,8D,D1,D4,04,E0,E6,06
       ,17,A6,DF,E4,8C,DA
KQ 230 DATA 05,FC,83,1A,C2,E2,
       C9
```

Program 2:

```
GA 10 GOTO30
EF 20 REM **PLACE KEYWORD HERE
      **
AG 30 ={B}15:B=PEEK(45)+12+PEE
      K(46)*256:PRINT"KEYWORD
      {SPACE}VALUE(S) = ";
PC 40 H=PEEK(B):PRINTRIGHT$(DI
      M(H),2);" ";
AE 50 IFH=254ORH=206THENB=B+1:
      GOTO40
```

DOS Calc

See instructions in article on page 31 before typing in.

```
0801:0B 08 0A 00 9E 32 30 36 2E
0809:31 00 00 00 A0 00 A2 10 0C
0811:A9 00 85 FB A9 C0 85 FC BE
0819:A9 01 85 FD A9 08 85 FE 46
0821:B1 FD 91 FB C8 D0 F9 E6 DF
0829:FE E6 FC CA D0 F2 4C 31 DA
0831:C0 AD 08 03 8D AA C0 AD 84
0839:09 03 8D AB C0 A9 4D 8D CF
0841:08 03 A9 C0 8D 09 03 A9 97
0849:FF 8D 6A C1 60 A5 3A C9 FD
0851:FF F0 03 6C AA C0 AD 00 78
0859:02 C9 40 D0 F6 A9 8E 20 8D
0861:D2 FF A9 13 A0 01 A2 01 90
0869:20 84 C1 A9 AD A0 C0 20 0F
0871:E9 C1 A9 BB A0 C0 38 20 70
0879:FB C1 20 66 C2 A9 00 8D AC
0881:AC C0 8D BD CE A9 20 8D 90
0889:8C CE A9 41 A0 52 20 12 7D
0891:C3 AA BD 4A C1 8D 9F C0 CE
0899:BD 4B C1 8D A0 C0 20 E2 97
08A1:FC 4C 7D C0 20 6C C2 6C A3
08A9:02 A0 00 00 00 46 55 4E F4
08B1:43 54 49 4F 4E 20 4D 45 69
08B9:4E 55 FF 45 58 49 54 00 2B
08C1:44 49 52 45 43 54 4F 52 41
08C9:59 00 4C 4F 41 44 00 4C 6C
08D1:4F 41 44 20 52 45 4C 4F F3
08D9:43 41 54 45 44 00 53 41 C4
08E1:56 45 00 52 45 53 41 56 E3
08E9:45 00 42 53 41 56 45 00 08
08F1:52 45 2D 42 53 41 56 45 D7
08F9:00 53 43 52 41 54 43 48 96
0901:00 52 45 4E 41 4D 45 00 FE
0909:43 4F 50 59 00 56 41 4C 58
0911:49 44 41 54 45 00 49 4E 51
0919:49 54 49 41 4C 49 5A 45 A3
0921:00 52 45 41 44 20 45 52 04
0929:52 2E 20 43 48 4E 2E 00 FF
0931:4E 45 57 20 41 20 44 49 05
0939:53 4B 45 54 54 45 00 43 B0
0941:41 4C 43 55 4C 41 54 4F 24
0949:52 FF A4 C0 3B C9 DD CA AC
0951:DA CA 6D CB 6A CB 64 CB 05
0959:67 CB 63 CC E0 CC DD CC 0E
0961:54 CD 58 CD 60 CD 5A CE B6
0969:2C C5 FF 00 41 00 00 00 0D
0971:00 00 00 00 00 00 00 00 83
0979:00 00 00 0A 0D 0E A0 A8 B6
0981:B0 A4 AC B4 48 8A EE 6A AA
0989:C1 AE 6A C1 9D 72 C1 8D 59
0991:6B C1 98 9D 75 C1 68 9D D7
0999:78 C1 E0 00 D0 26 AD A7 96
09A1:02 8D 70 C1 AD A7 02 8D DF
09A9:6F C1 AD A7 02 8D 71 C1 FE
09B1:38 20 F0 FF 8E 6D C1 8C 40
09B9:6E C1 A9 00 8D A7 02 A9 61
09C1:00 8D A7 02 A9 04 85 FC B1
09C9:AE 6A C1 BD 7E C1 85 FE E6
09D1:20 AC C2 AD 6A C1 D0 05 53
09D9:A9 93 20 D2 FF 20 F9 C2 0E
09E1:A9 00 20 CF C2 20 F6 C2 11
09E9:60 A2 41 8E 6C C1 18 20 A0
09F1:FB C1 A9 02 20 CF C2 20 AD
09F9:F6 C2 60 85 FB 84 FC A9 32
0A01:18 90 02 A9 38 8D 15 C2 05
0A09:A9 62 20 D2 FF AC 6A C1 05
0A11:BE 78 C1 CA CA 18 90 16 75
0A19:AD 6C C1 20 D2 FF A9 2E 71
0A21:20 D2 FF A9 20 20 D2 FF BB
0A29:EE 6C C1 CA CA CA A0 00 77
0A31:B1 FB F0 07 C9 FF F0 03 DE
0A39:C8 D0 02 A9 20 20 D2 FF E7
0A41:CA D0 ED B1 FB 85 FD 84 3E
0A49:FE A9 62 20 D2 FF 20 F6 63
0A51:C2 A5 FD D0 10 E6 FE 18 2F
0A59:A5 FB 65 FE 85 FB 90 02 1B
0A61:E6 FC 4C 08 C2 60 A9 04 21
0A69:20 CF C2 60 AE 6A C1 BD 40
0A71:7E C1 85 FC A9 04 85 FE 1D
0A79:20 AC C2 CE 6A C1 AE 6A 30
0A81:C1 E0 FF F0 0A BD 72 C1 AB
0A89:8D 6B C1 20 F9 C2 60 AD C2
0A91:6F C1 8D A7 02 AD 70 C1 63
0A99:8D A7 02 AD 71 C1 8D A7 CE
0AA1:02 AE 6D C1 AC 6E C1 18 E6
0AA9:20 F0 FF 60 A5 01 29 FE 92
0AB1:85 01 A0 00 84 FB 84 FD F7
0AB9:A2 04 B1 FB 91 FD C8 D0 FC
0AC1:F9 E6 FC E6 FE CA D0 F2 52
0AC9:A5 01 09 01 85 01 60 48 5B
0AD1:AA BD F0 C2 20 D2 FF AC ED
0AD9:6A C1 BE 78 C1 CA CA A9 6B
0AE1:60 20 D2 FF CA D0 F8 68 7C
0AE9:AA BD F1 C2 20 D2 FF 60 D9
0AF1:75 69 AB B3 6A 6B EE 6B 16
0AF9:C1 AE 6A C1 BD 7B C1 8D F0
0B01:A7 02 BC 75 C1 AE 6B C1 BB
0B09:18 20 F0 FF 60 A9 41 AC 2A
0B11:6C C1 85 FB 84 FC 20 E4 7B
0B19:FF C9 00 F0 F9 C5 FB 90 20
0B21:F5 C5 FC B0 F1 38 E5 FB 86
0B29:0A 60 48 A9 8F 85 FB A9 34
0B31:CE 85 FC 68 85 FD 18 20 AA
0B39:F0 FF A9 00 85 FE A5 C6 37
0B41:85 CC F0 FA 78 A5 CF F0 06
0B49:0C A5 CE AE 87 02 A0 00 19
0B51:84 CF 20 13 EA 58 20 E4 B0
0B59:FF C9 14 F0 21 C9 0D F0 AE
0B61:39 C9 20 90 D9 C9 60 B0 FA
0B69:D5 A6 FD E4 FE F0 CF AC 2A
0B71:BD CE 91 FB 20 D2 FF E6 3F
0B79:FE EE BD CE D0 C0 A4 FE 41
0B81:F0 BC C6 FE CE BD CE 88 9B
0B89:A9 9D 20 D2 FF A9 20 20 14
0B91:D2 FF A9 9D 20 D2 FF 4C B8
0B99:3E C3 A6 FE AC BD CE F0 6F
0BA1:0C A9 20 88 D1 FB D0 04 D8
0BA9:88 CA D0 F8 C8 8C BD CE 23
0BB1:8A 60 46 55 4E 43 54 49 B4
0BB9:4F 4E 53 3A FF 45 58 49 28
0BC1:54 00 43 4C 45 41 52 20 23
0BC9:45 4E 54 52 59 00 43 4C 63
0BD1:45 41 52 20 53 54 41 43 D8
0BD9:4B 00 57 52 41 50 20 55 86
```

```
0BE1:50 00 57 52 41 50 20 44 FF
0BE9:4F 57 4E 00 52 4F 4C 4C FB
0BF1:20 55 50 00 52 4F 4C 4C 2C
0BF9:20 44 4F 57 4E 00 45 58 E5
0C01:43 48 41 4E 47 45 00 43 6C
0C09:48 41 4E 47 45 20 53 49 6E
0C11:47 4E 00 49 4E 54 45 47 8A
0C19:45 52 00 53 51 55 41 52 52
0C21:45 20 52 4F 4F 54 00 50 3F
0C29:4F 57 45 52 00 50 49 00 60
0C31:53 49 4E 45 00 43 4F 53 62
0C39:49 4E 45 00 54 41 4E 47 BD
0C41:45 4E 54 00 41 52 43 54 48
0C49:41 4E 47 45 4E 54 00 44 DA
0C51:45 47 2D 3E 52 41 44 00 87
0C59:52 41 44 2D 3E 44 45 47 1B
0C61:00 4C 4F 47 2F 45 00 45 BE
0C69:58 50 2F 45 FF 53 54 41 33
0C71:43 4B 3A 00 00 00 00 00 45
0C79:00 00 00 00 FF 52 45 53 B8
0C81:55 4C 54 3A 00 00 00 FF 85
0C89:4D 45 53 53 41 47 45 53 3E
0C91:3A FF 49 4C 4C 45 47 41 FB
0C99:4C 20 51 55 41 4E 54 49 94
0CA1:54 59 FF 4F 56 45 52 46 E1
0CA9:4C 4F 57 FF 44 49 56 49 E3
0CB1:53 49 4F 4E 20 42 59 20 71
0CB9:5A 45 52 4F FF 55 4E 4B CC
0CC1:4E 4F 57 4E FF 00 00 00 A4
0CC9:00 00 00 00 00 00 00 00 E1
0CD1:00 00 00 00 00 00 00 00 E9
0CD9:00 00 00 00 00 00 41 42 B6
0CE1:43 44 45 46 47 48 49 4A F1
0CE9:4B 4C 4D 4E 4F 50 51 52 F9
0CF1:53 54 55 2B 2D 2A 2F 0D A3
0CF9:AE C6 BF C6 DB C6 EC C6 1A
0D01:F8 C6 04 C7 20 C7 3C C7 A6
0D09:53 C7 5B C7 60 C7 65 C7 5B
0D11:77 C7 88 C7 8D C7 92 C7 DE
0D19:97 C7 9C C7 B1 C7 BA C7 EA
0D21:BF C7 7E C6 8A C6 96 C6 AC
0D29:A2 C6 70 C6 A9 12 A0 15 AC
0D31:A2 00 20 84 C1 A9 B2 A0 A3
0D39:C3 20 E9 C1 A9 BD A0 C3 DF
0D41:38 20 FB C1 20 66 C2 A9 E4
0D49:13 A0 01 A2 00 20 84 C1 AA
0D51:A9 7D A0 C4 20 E9 C1 A9 D5
0D59:6D A0 C4 18 20 FB C1 A9 8A
0D61:02 20 CF C2 20 F6 C2 A9 B6
0D69:88 A0 C4 18 20 FB C1 A9 28
0D71:7A A0 C4 18 20 FB C1 20 9F
0D79:66 C2 AD 00 03 48 AD 01 C2
0D81:03 48 A9 78 8D 00 03 A9 08
0D89:C8 8D 01 03 20 DB C6 20 D9
0D91:F1 C7 A9 00 A0 DE 20 12 9E
0D99:C3 6A 48 A9 13 8D 6B C1 3B
0DA1:20 F9 C2 A9 87 A0 C4 18 9D
0DA9:20 FB C1 68 C9 30 90 07 C8
0DB1:C9 3A B0 03 4C 25 C6 C9 D3
0DB9:5E D0 04 A9 45 D0 F5 C9 75
0DC1:DD D0 04 A9 2D D0 ED C9 6C
0DC9:2E F0 E9 C9 14 D0 03 4C 47
0DD1:60 C6 A0 00 A2 1A D9 DE F1
0DD9:C4 F0 06 C8 CA D0 F7 F0 5A
0DE1:B1 C0 0C F0 05 A9 00 8D F1
0DE9:7D C6 98 48 20 D4 C7 68 58
0DF1:0A AA BD F8 C4 8D 08 C6 36
0DF9:BD F9 C4 8D 09 C6 20 C4 4B
0E01:C7 BA CA CA 8E 7A C8 20 C5
0E09:07 C6 B0 03 20 CC C7 4C B0
0E11:8F C5 20 20 20 20 20 20 4E
0E19:20 20 20 20 20 20 20 20 35
0E21:20 30 FF 00 00 48 AD 24 F9
0E29:C6 D0 0B AD 7D C6 D0 06 C7
0E31:20 20 C7 20 00 C8 68 8D E1
0E39:24 C6 AC 23 C6 C0 10 F0 2B
0E41:1C 99 12 C6 C8 8C 23 C6 06
0E49:A9 FF 99 12 C6 A9 03 8D FE
0E51:6B C1 20 F9 C2 A9 12 A0 B8
0E59:C6 18 20 FB C1 4C 92 C5 CC
0E61:AC 23 C6 F0 F8 88 A9 FF C1
0E69:99 12 C6 8C 23 C6 D0 DD 2C
0E71:20 20 C7 20 CC C7 A9 01 7A
0E79:8D 7D C6 38 60 00 20 04 5F
0E81:C7 A0 C4 A9 C5 20 67 B8 13
0E89:18 60 20 04 C7 A0 C4 A9 02
0E91:C5 20 50 B8 18 60 20 04 B4
0E99:C7 A0 C4 A9 C5 20 28 BA AE
0EA1:18 60 20 04 C7 A0 C4 A9 1A
0EA9:C5 20 0F BB 18 60 20 6C 3D
0EB1:C2 20 6C C2 68 68 68 8D 34
0EB9:01 03 68 8D 00 03 60 20 E9
0EC1:2E C7 A9 FF 8D 7D C6 A9 B5
0EC9:30 8D 12 C6 A9 FF 8D 13 8B
0ED1:C6 A9 00 8D 23 C6 8D 24 08
0ED9:C6 38 60 A2 19 A0 00 98 81
0EE1:99 C5 C4 C8 CA D0 F9 20 0F
0EE9:C7 C6 38 60 20 04 C7 A2 EB
0EF1:D9 A0 C4 20 D4 BB 38 60 24
0EF9:A9 D9 A0 C4 20 A2 BB 20 E4
0F01:20 C7 18 60 A2 14 A0 00 D0
0F09:B9 CA C4 99 C5 C4 C8 CA 86
0F11:D0 F6 A9 00 A8 A2 05 99 FD
0F19:D9 C4 C8 CA D0 F9 38 60 5A
0F21:A2 14 A0 13 B9 C5 C4 99 E2
0F29:CA C4 88 CA D0 F6 A9 00 51
0F31:A8 A2 05 99 C5 C4 C8 CA 24
0F39:D0 F9 38 60 A2 05 A0 00 B5
0F41:B9 CA C4 99 C5 C4 C8 CA BE
0F49:D0 F6 A2 CA A0 C4 20 D4 BB
0F51:BB 38 60 A5 66 49 FF 85 9F
0F59:66 18 60 20 CC BC 18 60 A8
0F61:20 71 BF 18 60 A0 C4 A9 1E
0F69:CA 20 8C BA 20 C4 C7 20 F5
0F71:7B BF 20 04 C7 18 60 A0 81
0F79:AE A9 A8 20 A2 BB AD 7D 4D
0F81:C6 D0 03 20 20 C7 18 60 4A
0F89:20 6B E2 18 60 20 64 E2 9F
0F91:18 60 20 B4 E2 18 60 20 7B
0F99:0E E3 18 60 A0 C7 A9 AC E4
0FA1:20 A2 BB A0 C4 A9 C5 20 72
0FA9:0F BB 18 60 86 65 2E E0 4E
0FB1:D4 A0 C7 A9 AC 20 28 BA E6
0FB9:18 60 20 EA B9 18 60 20 BD
0FC1:ED BF 18 60 A0 C4 A9 C5 01
0FC9:20 A2 BB 60 A2 C5 A0 C4 50
0FD1:20 D4 BB 60 AD 24 C6 F0 2F
0FD9:17 A9 12 85 7A A9 C6 85 16
0FE1:7B 20 79 00 20 F3 BC 20 5F
0FE9:1B BC 20 CC C7 20 C7 C6 AA
0FF1:60 A9 03 8D 6B C1 20 F9 80
0FF9:C2 A0 C4 A9 C5 20 2E C8 A8
1001:A9 08 8D 6B C1 20 F9 C2 A5
1009:A0 C4 A9 CA 20 2E C8 20 F7
1011:F6 C2 A0 C4 A9 CF 20 2E B8
1019:C8 20 F6 C2 A0 C4 A9 D4 F0
1021:20 2E C8 20 F6 C2 A0 C4 C0
1029:A9 D9 20 2E C8 60 20 A2 26
1031:BB 20 DD BD 20 59 C8 A0 67
1039:00 B9 00 01 F0 03 C8 D0 CD
1041:F8 A2 0F 88 B9 00 01 9D 5E
1049:67 C8 CA 88 C0 FF D0 F4 CD
1051:A9 67 A0 C8 18 20 FB C1 BB
1059:60 A9 20 A0 00 A2 10 99 66
1061:67 C8 C8 CA D0 F9 60 20 7C
1069:20 20 20 20 20 20 20 20 89
1071:20 20 20 20 20 20 39 FF A3
1079:8A A2 00 9A 48 A9 13 8D CD
1081:6B C1 20 F9 C2 68 C9 0E C4
1089:D0 09 A9 92 A0 C4 18 20 1B
1091:FB C1 60 C9 0F D0 09 A9 40
1099:A3 A0 C4 18 20 FB C1 60 A2
10A1:C9 14 D0 09 A9 AC A0 C4 5C
10A9:18 20 FB C1 60 A9 BD A0 3F
10B1:C4 18 20 FB C1 60 57 49 85
10B9:4C 44 20 43 41 52 44 20 45
10C1:4F 52 20 52 45 54 55 52 BF
10C9:4E FF 20 20 20 20 20 20 F8
10D1:20 20 20 20 20 20 20 20 F1
10D9:20 20 20 20 20 20 20 20 F9
10E1:20 20 20 20 FF 20 20 20 01
10E9:20 20 20 20 20 20 20 20 0A
10F1:20 20 20 20 20 20 20 20 12
10F9:20 20 20 20 20 20 20 20 1A
1101:20 20 20 20 20 20 FF 45 08
1109:58 49 54 00 4E 45 58 54 C0
1111:20 50 41 47 45 00 4E 45 FF
1119:57 20 53 45 41 52 43 48 CF
1121:FF 24 30 3A 2A 20 20 20 28
1129:20 20 20 20 20 20 20 20 4B
1131:20 20 20 20 24 30 3A 2A F1
1139:FF 00 00 20 79 CD 90 01 82
1141:60 A9 16 A0 06 A2 04 20 AD
1149:84 C1 A9 B6 A0 C8 20 E9 11
1151:C1 A9 34 A0 C9 18 20 FB 3A
1159:C1 20 66 C2 A9 24 85 FB 42
1161:A9 C9 85 FC AD BD CE 48 95
1169:A9 00 8D BD CE A2 07 A0 9D
1171:0A A9 10 20 34 C3 D0 07 60
1179:A9 2A 8D 24 C9 A9 01 18 FD
1181:69 03 A2 21 A0 C9 20 BD A9
1189:FF 68 8D BD CE A9 01 A2 15
1191:08 A0 00 8C 3A C9 20 BA 9C
1199:FF 20 C0 FF 20 6F CD 90 C6
11A1:05 A9 00 4C 2C CA 20 6C AE
11A9:C2 A2 01 20 C6 FF 20 CF 3E
11B1:FF C9 22 D0 F9 A0 00 99 83
11B9:CA C8 20 CF FF C8 C9 00 2B
11C1:D0 F5 20 CC FF A9 23 A0 28
11C9:02 A2 01 20 84 C1 A9 CA 01
11D1:A0 C8 20 E9 C1 A9 07 A0 7C
11D9:C9 38 20 FB C1 A9 11 8D 17
11E1:39 C9 A2 01 20 C6 FF A2 36
11E9:04 A8 20 CF FF CA D0 F9 FF
11F1:20 B3 CA 20 CF FF C9 00 7E
11F9:F0 06 99 E5 C8 C8 D0 F3 A6
1201:20 CC FF AD EA C8 C9 22 73
1209:D0 01 A9 E5 A0 C8 38 20 32
1211:FB C1 CE 39 C9 D0 CB 20 5A
1219:66 C2 20 0D C3 C9 02 D0 10
1221:0B AD 3A C9 D0 F4 20 6C 21
1229:C2 4C C5 C9 48 A9 01 20 22
1231:C3 FF 68 C9 04 D0 06 20 70
1239:6C C2 4C 41 C9 48 20 8A 1C
1241:CD 68 AE AC C0 D0 17 20 9E
1249:6C C2 60 EE 3A C9 A9 E5 81
1251:A0 C8 18 20 FB C1 A9 01 38
1259:20 C3 FF 4C 17 CA C9 00 BA
1261:D0 05 20 6C C2 38 60 4A FB
1269:18 69 04 AA A0 0C 20 F0 85
1271:FF A9 03 A2 03 A0 FF 20 45
1279:BA FF A9 00 20 BD FF 20 48
1281:C0 FF A2 03 20 C6 FF A2 49
1289:10 AC BD CE 20 CF FF 99 5F
1291:8F CE C9 22 F0 04 C8 CA 80
1299:D0 F2 8C BD CE A9 FF 99 07
12A1:8F CE 20 CC FF A9 03 20 DE
12A9:C3 FF 20 6C C2 A9 00 85 BC
12B1:D4 18 60 20 91 B3 20 DD CD
12B9:BD A0 00 B9 01 01 F0 06 74
12C1:99 E5 C8 C8 D0 F5 A9 20 A3
12C9:99 E5 C8 C8 60 4C 4F 41 ED
12D1:44 FF 4E 41 4D 45 00 30 A5
12D9:3A FF A9 01 2C A9 00 8D F5
12E1:0D CB A9 18 A0 06 A2 05 9D
12E9:20 84 C1 A9 CD A0 CA 20 B8
12F1:E9 C1 A9 D2 A0 CA 18 20 5E
12F9:FB C1 20 66 C2 A2 09 A0 4A
1301:09 A9 10 20 2A C3 F0 31 8D
1309:A9 01 A2 08 A0 00 20 BA 19
1311:FF AD BD CE A2 8F A0 CE AA
1319:20 BD FF A9 00 85 9D A6 51
1321:2B A4 2C 20 D5 FF 86 2D 76
1329:84 2E 20 60 CD 90 06 20 04
1331:6C C2 4C 44 A6 20 6C C2 5D
1339:60 20 9C CC 90 CA B0 F5 FE
1341:28 52 45 29 20 28 42 29 9A
1349:20 53 41 56 45 FF 4E 41 E9
1351:4D 45 00 30 3A FF 00 53 97
1359:54 41 52 54 3A 00 00 45 A0
1361:4E 44 3A FF A9 40 2C A9 57
1369:C0 2C A9 80 2C A9 00 85 C5
1371:02 A9 18 A0 06 A2 05 20 F4
1379:84 C1 A9 40 A0 CB 20 E9 E9
1381:C1 A9 4E A0 CB 18 20 FB C1
1389:C1 24 02 50 08 A9 56 A0 13
1391:CB 18 20 FB C1 20 66 C2 85
1399:A2 09 A0 09 A9 10 20 2A EF
13A1:C3 D0 06 24 02 30 66 10 8E
13A9:60 24 02 10 0B EE AC C0 78
13B1:A9 53 8D 8C CE 20 BE CE 3F
13B9:AD BD CE A2 8F A0 CE 20 E6
13C1:BD FF 24 02 50 1F A2 0B BA
13C9:A0 0E 20 14 CC B0 F7 48 6A
13D1:98 48 A2 0D A0 0C 20 14 05
```

```
13D9:CC B0 F7 84 FD 85 FE 68 46
13E1:85 FB 68 85 FC A9 01 A2 62
13E9:08 A0 00 84 9D 20 BA FF 67
13F1:24 02 50 09 A9 FB A6 FD CD
13F9:A4 FE 4C 03 CC A9 2B A6 F5
1401:2D A4 2E 20 D8 FF 20 60 18
1409:CD 20 6C C2 60 20 9C CC 63
1411:90 97 B0 F5 A9 12 85 FB 79
1419:A9 C6 85 FC A9 00 8D BD 6E
1421:CE A9 05 20 34 C3 A9 0D CE
1429:91 FB 98 AA A0 00 A9 2F 5E
1431:D1 FB B0 1D B1 FB C9 3A 74
1439:B0 17 C8 CA D0 F0 20 D9 A9
1441:C7 20 9B BC 18 A5 62 65 16
1449:63 38 D0 05 A5 64 A4 65 09
1451:18 60 53 43 52 41 54 43 BF
1459:48 FF 4E 41 4D 45 00 53 56
1461:30 3A FF A9 18 A0 06 A2 BC
1469:05 20 84 C1 A9 52 A0 CC 6D
1471:20 E9 C1 A9 5A A0 CC 18 FD
1479:20 FB C1 20 66 C2 A9 53 CF
1481:8D 8C CE A2 09 A0 0A A9 20
1489:10 20 2A C3 F0 07 20 BE E5
1491:CE 20 6C C2 60 20 9C CC 6C
1499:90 F4 B0 F5 EE AC C0 20 88
14A1:3B C9 08 A9 09 8D 6B C1 8C
14A9:20 F9 C2 A9 8C A0 CE 18 EF
14B1:20 FB C1 A9 00 8D AC C0 0C
14B9:28 60 43 4F 50 59 20 4F E2
14C1:52 20 52 45 4E 41 4D 45 11
14C9:FF 4E 45 57 20 4E 41 4D AD
14D1:45 00 00 4F 4C 44 20 4E 93
14D9:41 4D 45 00 FF A9 43 2C F7
14E1:A9 52 8D 8C CE A9 13 A0 D1
14E9:09 A2 05 20 84 C1 A9 BA 1B
14F1:A0 CC 20 E9 C1 A9 C9 A0 29
14F9:CC 18 20 FB C1 20 66 C2 70
1501:A2 09 A0 0A A9 10 20 2A 6B
1509:C3 F0 19 AC BD CE A9 3D F8
1511:99 8F CE EE BD CE A2 0B 2E
1519:A0 0A A9 10 20 2A C3 F0 6E
1521:07 20 BE CE 20 6C C2 60 34
1529:AD BD CE 8D 44 CD EE AC 30
1531:C0 20 3B C9 B0 EE A9 0B 67
1539:8D 6B C1 20 F9 C2 A9 8F FC
1541:A0 CE 18 69 00 90 01 C8 16
1549:18 20 FB C1 A9 00 8D AC 38
1551:C0 4C 21 CD A9 56 D0 02 3A
1559:A9 49 8D 8C CE 4C BE CE 19
1561:20 79 CD 20 90 CD 08 20 A1
1569:CE CD 20 8A CD 28 60 20 0B
1571:90 CD 90 04 20 CE CD 38 B9
1579:60 A9 0F A2 08 A8 20 BA 28
1581:FF A9 00 20 BD FF 20 C0 07
1589:FF 60 A9 0F 20 C3 FF 60 62
1591:A2 0F 20 C6 FF B0 27 A0 F2
1599:00 20 CF FF C9 0D F0 06 30
15A1:99 15 CE C8 D0 F3 A9 FF ED
15A9:99 15 CE 20 CC FF A9 30 AB
15B1:CD 15 CE D0 07 CD 16 CE 59
15B9:D0 02 18 60 38 60 A0 00 5A
15C1:A2 0D B9 32 CE 99 15 CE B0
15C9:C8 CA D0 F6 F0 EE A9 1E 49
15D1:A0 05 A2 0C 20 84 C1 A9 E2
15D9:F4 A0 CD 20 E9 C1 A9 15 21
15E1:A0 CE 18 20 FB C1 20 66 A2
15E9:C2 A9 00 A0 FE 20 12 C3 4A
15F1:20 6C C2 60 44 49 53 4B DE
15F9:20 45 52 52 4F 52 20 44 3D
1601:49 53 50 4C 41 59 00 50 35
1609:52 45 53 53 20 41 4E 59 4B
1611:20 4B 45 59 FF 20 20 20 3F
1619:20 20 20 20 20 20 20 20 45
1621:20 20 20 20 20 20 20 20 4D
1629:20 20 20 20 20 20 20 20 55
1631:20 FF 44 45 56 49 43 45 ED
1639:20 45 52 52 4F 52 FF 4E 48
1641:45 57 20 41 20 44 49 53 F5
1649:4B 45 54 54 45 FF 4E 41 44
1651:4D 45 2C 49 44 00 4E 30 7E
1659:3A FF A9 18 A0 06 A2 05 C0
1661:20 84 C1 A9 3F A0 CE 20 CB
1669:E9 C1 A9 4E A0 CE 18 20 A5
1671:FB C1 20 66 C2 A2 09 A0 C9
1679:0A A9 13 20 2A C3 F0 08 C3
1681:A9 4E 8D 8C CE 20 BE CE D3
1689:20 6C C2 60 49 30 3A 20 DE
1691:20 20 20 20 20 20 20 20 BD
1699:20 20 20 20 20 20 20 20 C5
16A1:20 20 20 20 20 20 20 20 CD
16A9:20 20 20 20 20 20 20 20 D5
16B1:20 20 20 20 20 20 20 20 DD
16B9:20 20 20 20 20 00 20 79 BE
16C1:CD A2 0F 20 C9 FF A0 00 F0
16C9:AE BD CE E8 E8 E8 B9 8C 10
16D1:CE 20 D2 FF C8 CA D0 F6 D1
16D9:20 CC FF AD AC C0 F0 03 71
16E1:4C 8A CD 4C 63 00 00 00 70
```

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

FACTORY AUTHORIZED

COMMODORE REPAIR CENTER

1-800-772-7289

(312) 879-2888 IL

C64 Repair (PCB ONLY)	42.95	Amiga Repair (PCB ONLY)	99.95
C128 Repair (PCB ONLY)	64.95	Amiga Drive Repair	149.95
1541 Permanent Alignment	29.95	Printers	CALL
1541 Repair	79.95	Monitors	CALL
1571 Repair	79.95	Other Equipment	CALL

CALL BEFORE SHIPPING
PARTS AND LABOR INCLUDED
FREE RETURN SHIPPING
(APO, FPO, AIR ADD $10.00)
24-48 HR. TURNAROUND
(Subject to Parts Availability)
30 DAY WARRANTY ON ALL REPAIRS

COMMODORE PARTS

C-64 Power Supply	34.95
128 Power Supply	59.95
C-64 Over Voltage Sensor	19.95
Other Parts	CALL

(Plus $3.00 Shipping/Handling)

All parts for Commodore equipment usually in stock
For Parts Call (312) 879-2350
Dealer Discounts Available

TEKTONICS PLUS, INC.

150 HOUSTON STREET
BATAVIA, IL 60510
CLIP AND SAVE

MasterCard VISA

SUPER 81 UTILITIES

Super 81 Utilities is a complete utilities package for the Commodore 1581 Disk Drive and C128 computer. Copy whole disks or individual files from 1541 or 1571 format to 1581 partitions. Backup 1581 disks. Contains 1581 Disk Editor, Drive Monitor, RAM Writer, CP/M Utilities and more for only **$39.95.**

1541/1571 DRIVE ALIGNMENT

1541/1571 Drive Alignment reports the alignment condition of the disk drive as you perform adjustments. Includes features for speed adjustment and stop adjustment. Includes program disk, calibration disk and instruction manual. Works on C64, C128, SX64, 1541, 1571. Only **$34.95.**

"...excellent, efficient program that can help you save both money and downtime." Compute!'s Gazette, Dec., 1987.

GALACTIC FRONTIER

Exciting space exploration game fro the C64. Search for life forms among the 200 billion stars in our galaxy. Scientifically accurate. Awesome graphics! For the serious student of astronomy or the causal explorer who wants to boldly go where no man has gone before. Only **$29.95.**

MONDAY MORNING MANAGER

Statistics-based baseball game. Includes 64 all-time great major league teams. Realistic strategy. Great sound & graphics! Apple II systems - **$44.95**, C-64 & Atari systems - **$39.95.**

Order with check, money order, VISA, MasterCard, COD. **Free shipping & handling** on US, Canadian, APO, FPO orders. COD & Foreign orders add $4.00. **Order from:**

VISA

Free Spirit Software, Inc.
905 W. Hillgrove, Suite 6
LaGrange, IL 60525
(312) 352-7323

MasterCard

ADVANCED DETECTION & CONTROL

NO INTERFACE IS EASIER TO USE

- Control lights, appliances, heating and cooling systems, relays, motors and virtually any electrical device.
- Connect to temperature, light, sound, moisture, fluid level and pressure sensors.
- 32 separate buffered digital output lines.
- 32 separate digital input lines.

BH100 I/O Card — User Manual, instructions, sample programs & diagrams of typical hookups. **$129**

8 Channel A/D Plug-in Conversion Module $45 — Fast 8 bit 0-5V A/D converter.

Complete I/O Course — Includes BH100, Beginner's Module, Course Manual and User Manual. **$159**

Security System Module — With siren, switches and software. **$35**

Beginner's I/O Interface Course Manual $15 — Learn all about interfacing!

Beginner's Module $25 — With 8 LEDs, 8 switches and 1 relay. Requires the BH100.

INTELLIGENT I/O

7 WATER ST. • MASSENA, NY 13662

VISA **(315) 769-2716** MasterCard

Call for free brochure today!

Beware of imitations - get the high quality original BH100
Add $4 for postage and specify C-64 or VIC-20.

MADE IN U.S.A.
5¼" DISKETTES

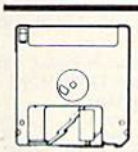
DSDD **33¢** each
DSHD **89¢** each

- Lifetime Warranty
- Error Free

Call for **FREE CATALOG!**
QUALITY SUPPLIES!
Budget—Friendly Prices!

- Sturdy Blank PVC Jacket
- With ID Label
- With Write Protect Tabs
- With Envelopes
- In Factory Sealed Poly Packs

SOLD IN LOTS OF 100



3½" DISKETTES
(UNBRANDED)
SALE **99¢** each
SOLD IN LOTS OF 50

S & H: $4.00, FIRST 100 OR FEWER DISKS; $3.00, EACH SUCCEEDING 100 OR FEWER DISKETTES.

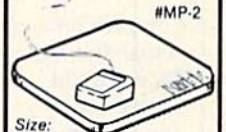
MINIMUM ORDER: $25.00. S&H: Continental USA. Foreign Orders, APO/FPO, please call. MI residents add 4% tax. C.O.D. add $4.00; payment with cash, certified check or money order. Prices subject to change. Hours: 8:30 AM—7:00 PM ET.

COMPUTERFACTS™ Publications from **SAMS**
VERY AFFORDABLE. **Discount Prices!**
•Wiring Diagrams •Repair Data •Parts Lists
•Trouble Shooting Techniques for all popular brands of microcomputers, monitors, printers & disk drives.

Apple II, II+, IIc or IIe	$18.95 ea.
Commodore 1541 disk drive	$18.95 ea.
IBM PC-AT 5170	$37.95 ea.
Okidata 82, 92 or 84 Printer	$18.95 ea.

Call Toll-free for price on other models.

NEW!
MOUSE EASE
#MP-2



Size: 8" x 9½"

Protective nylon laminated rubber mouse pad keeps mouse cleaner and tracking smoother. Protects the mouse and surface it is on. No more scratching. Rubber surface won't let pad slip under the toughest conditions.

$6.95
PLUS $2.00 S & H

COD
Precision Data Products™
P.O. Box 8367, Grand Rapids, MI 49518
(616) 452-3457 • FAX: (616) 452-4914
Michigan 1-800-632-2468
Outside Michigan 1-800-258-0028

Awards 1987

SOFTWARE DISCOUNTERS OF AMERICA

S.D. of A.

For Orders Only—1-800-225-7638
PA Orders—1-800-223-7784
Customer Service 412-361-5291

- Free shipping on orders over $100 in continental USA
- No Surcharge for VISA/MasterCard
- Your card is not charged until we ship

Commodore 64/128 Bargain Basement—Dozens of Titles For Less Than $10!

Simon&Schuster
COMPUTER SOFTWARE

Design, print, fold & fly award-winning high-performance airplanes.

Great International Paper Airplane Construction Kit
Our Discount Price $9.88

ACCOLADE
Dambusters $9.88
Law of the West $9.88
PSI-5 Trading Company $9.88

ACTIVISION
Cross Country Road Race $9.88
Ghostbusters $9.88
Hacker $9.88
Little Computer People $9.88

ARTWORX
Beach Blanket Volleyball $9.88
Equestrian Show Jumper $9.88
Highland Games $9.88
Hole in One + Six .. $9.88
Police Cadet $9.88
Thai Boxing $9.88

AVANTAGE
Deceptor $9.88
Desert Fox $9.88
Power $9.88
Project Space Station $9.88
Sigma 7 ..... $9.88
Spy vs. Spy 1 & 2 ... $9.88

BRODERBUND
Choplifter/David's Midnight Magic ... $9.88
Karateka $9.88
Loderunner $9.88

CDA
America Cooks Series:
American $9.88
Chinese $9.88
French $9.88
Italian $9.88
Mexican $9.88

DATA EAST
Last Mission $8.88
Q-Bert $8.88
TNK III $8.88

EASY WORKING/ SPINNAKER
Filer $6.88
Planner $6.88
Writer $6.88

ELECTRIC DREAMS
Spindizzy $9.88
Zoids $9.88

ELECTRONIC ARTS
Adv. Const. Set $9.88
Age of Adventure ... $9.88
America's Cup Sailing $9.88
Archon $9.88
Archon 2: Adept $9.88
Financial Cookbook . $9.88
Heart of Africa $9.88
Lords of Conquest .. $9.88
Mail Order Monsters $9.88
Mind Mirror $9.88
Movie Maker $9.88

M.U.L.E. $9.88
Murder Party $9.88
Music Const. Set ... $9.88
One-on-One $9.88
Pinball Const. Set ... $9.88
Racing Dest. Set $9.88
Seven Cities of Gold $9.88
Skyfox $9.88
Super Boulder Dash . $9.88
Touchdown Football . $9.88
Ultimate Wizard $9.88

EPYX
Ballblazer $6.88
Barbie $6.88
Jumpman $6.88
Gateway to Apshai .. $6.88
Impossible Mission .. $6.88
Koronis Rift $6.88
Pitstop 1 or 2 ... $6.88 Ea.
Rescue on Fractalus . $6.88
The Eidolon $6.88

FISHER PRICE
Dance Fantasy $4.88

KONAMI/ACTION CITY
Hyper Sports/Ping Pong $9.88
Mikie $9.88
Track & Field $9.88

MASTERTRONIC
Action Biker $4.88
Bounder $6.88
Captain Zap $6.88
Feud $6.88
Games Creator $6.88
Infinity Machine (R) .. $9.88
Knight Games $6.88
Last V-8 ..... $4.88
Master of Magic ... $4.88
Ninja $4.88
Pro Golf $4.88
Prowler: War in 2150 $6.88
Shogun $6.88
Speed King $6.88
Storm $6.88
Vegas Poker & Jackpot $4.88
Water Polo $6.88

THUNDER MOUNTAIN

Top Gun™ puts you in the fighter pilots seat of a technologically advanced F-14 Tomcat.

Top Gun™
Our Discount Price $6.88

The Activision Little Computer People Discovery Kit™

Amazing software lets you meet the little people living inside your computer! Make contact with the little people living inside your computer.

Our Discount Price $9.88

SIMON & SCHUSTER
Great International Paper Airplane Const. Kit . $9.88

SPECTRUM HOLOBYTE
Gato $9.88

SPINNAKER
Alf in the Color Cave $4.88
Bubble Busters $4.88
Cosmic Combat $4.88
Fraction Fever $4.88
Gold Record Race ... $4.88
Letter Scrambler ... $4.88
Monster Voyage $4.88
Ranch $4.88
Story Machine $4.88
*all above titles on cart.

TELARIUM
Amazon $9.88
Dragonworld $9.88
Fahrenheit 451 ..... $9.88
Nine Princes in Amber $9.88
Perry Mason: Case of Mandarin Murder .. $9.88
Rendezvous w/Rama . $9.88

THUNDER MOUNTAIN
Cyrus Chess $6.88
Demolition Mission .. $6.88
Dig Dug $6.88
Eliminator $6.88
Equinox $6.88
Great British Software $6.88
Leviathan $6.88
Maxi Golf $6.88
Mission in Our Solar System $6.88
Ms. Pac Man $6.88
Pac Man $6.88
Paradroid $6.88
Pole Position $6.88
Rambo: First Blood Part II $6.88
Slot Car Racer $6.88
Top Gun $6.88

UXB/SPINNAKER
Dark Tower $6.88
Karate Chop $6.88

VALUE WARE
Artist $4.88
Educator $4.88
Entertainer $4.88
Home Banker $4.88
Home Manager $4.88
Kitchen Manager $4.88

WINDHAM CLASSICS
Alice in Wonderland . $6.88
Below the Root $6.88
Swiss Family Robinson $6.88
Treasure Island $6.88
Wizard of Oz $6.88

Memory Manor $4.88
Number Tumblers ... $4.88
Sea Speller $4.88
*all above titles on cart.

GAMESTAR
Star League Baseball/ On Field Football . $9.88

HES
Microsoft Multiplan . $9.88

HI-TECH EXPRESSIONS
Award Ware $9.88
Card Ware $6.88
Party Ware $9.88

HI-TECH EXPRESSIONS
Sesame Street Series:
Astro-Grover $6.88
Big Bird's Special Delivery $6.88
Ernie's Big Splash ... $6.88
Ernie's Magic Shapes $6.88
Grover's Animal Adv. $6.88
Pals Around Town .. $6.88

KONAMI
Circus Charlie $9.88

MINDSCAPE
Bank St. Music Writer $9.88

RANDOM HOUSE
Charlie Brown's ABC's $6.88
Snoopy's Reading Machine $6.88
Snoopy Writer $6.88
Typing is a Ball, Charlie Brown ... $6.88

SECA
Congo Bongo (R) ... $6.88
Super Zazzon (R) ... $6.88

SSI
Computer Baseball .. $9.88
Computer Quarterback $9.88
Fifty Mission Crush .. $9.88
Gemstone Warrior ... $9.88
Germany 1985 ..... $9.88
Ringside Seat $9.88
Six-Gun Shootout ... $9.88
Wings of War $9.88

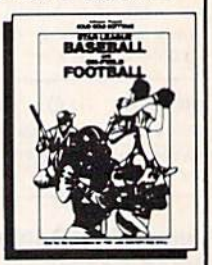
SHARE DATA
Family Feud $7.88
Jeopardy $7.88
Wheel of Fortune $7.88

THUNDER MOUNTAIN

Compete on actual Slot Car Racer circuit courses or design your own.

Slot Car Racer
Our Discount Price $6.88

SOLID GOLD SOFTWARE™



Two Gamestar Original Hits!

Nine innings of hard-hitting baseball & four quarters of punishing gridiron action on one disk!

Star League Baseball & On Field Football
Our Discount Price $9.88

P.O. BOX 111327—DEPT. CG—BLAWNOX, PA 15238

*Please Read The Following Ordering Terms & Conditions Carefully Before Placing Your Order: Orders with cashiers check or money order shipped immediately on in stock items! Personal & Company checks, allow 3 weeks clearance. No C.O.D.'s! Shipping: Continental U.S.A.-Orders under $100 add $3; free shipping on orders over $100. AK, HI, FPO, APO-add $5 on all orders. Canada & Puerto Rico-add $10 on all orders. Sorry, no other International orders accepted! PA residents add 6% sales tax on the total amount of order including shipping charges. CUSTOMER SERVICE HOURS: Mon.-Fri. 9 AM-5:30 PM Eastern Time. REASONS FOR CALLING CUSTOMER SERVICE—412-361-5291 (1)Status of order or back order (2)if any merchandise purchased within 60 days from S.D.of A. is defective, please call for a return authorization number. We will not process a return without a return auth. #! Defective merchandise will be replaced with the same merchandise only. Other returns subject to a 15% restocking charge! After 60 days from your purchase date, please refer to the warranty included with the product purchased & return directly to the manufacturer. Customer service will not accept collect calls or calls on S.D.of A.'s 800# order lines! ORDER LINE HOURS: Mon.-Fri. 9 AM-5:30 PM, SAT 10 AM-4 PM EASTERN TIME. Because this ad had to be written 2-3 mos. before it was published, prices & availability are subject to change! New titles are arriving daily! Please call for more information.

SOFTWARE DISCOUNTERS OF AMERICA

For Orders Only—1-800-225-7638
PA Orders—1-800-223-7784
Customer Service 412-361-5291

- **Free shipping on orders over $100 in continental USA**
- **No Surcharge for VISA/MasterCard**
- **Your card is not charged until we ship**

ABACUS BOOKS
Anatomy of the 1541 ..$14
Anatomy of the C64 ..$14
C128 Basic 7.0 Internals$19
C128 Basic Training Guide $12
C128 Internals $14
C128 Peeks & Pokes ..$12
C128 Tricks & Tips....$14
CPM on the C128 ....$14
1571 Internals $14
GEOS Inside & Out ..$13
GEOS Tricks & Tips ..$13

ABACUS SOFTWARE
Assembler Monitor ...$25
Basic $25
Basic 128 ....$39
Cad Pak $25
Cad Pak 128 ....$39
Chart Pak $25
Chart Pak 128 ....$25
Cobol $25
Cobol 128 ....$25
PPM $25
PPM 128 ....$39
Speed Term 64 or 128 $25
Super C $39
Super C 128 ....$39
Super Pascal $39
Super Pascal 128 ....$39
TAS $25
TAS 128 ....$39

ACCESS
Echelon w/Lip Stik ...$29
Famous Course Disk #1 for World Class L.B. .$14
Famous Course Disk #2 for World Class L.B. .$14
Famous Course Disk #3 for World Class L.B. .$14
Leader Board (Original) 3 Pack $14
Lip Stik Plus $17
Tenth Frame $25
Triple Pack: BH1, BH2, Raid Over Moscow .$14
World Class Leader Board $25

ACCOLADE
Ace of Aces $19
Apollo 18: Mission to the Moon $19
Comics $19
4th & Inches Football .$19
Hardball $19
Mini Putt $19
Test Drive $19

ACTION SOFT
Thunder Chopper $19
Up Periscope! $19

ACTIVISION
Aliens $23
Gamemaker w/Sports & Science Fiction Disks$33
Gee Bee Air Rally $19
Hacker 2 ....$23
Labyrinth $19
Maniac Mansion $23
Might & Magic $25
Music Studio $23
Postcards $16
Shanghai $19
The Last Ninja $23
Transformers $19

AMERICAN EDUCATIONAL
Biology $12
Grammar $12
Learn to Read (Gr. 1-4)$25
Phonics (K-3) $25
Reading Comprehension$25
Science: Grades 3/4 ...$12
Science: Grades 5/6 ...$12
Science: Grades 7/8 ...$12
US Geography $12
US History $12
World Geography $12
World History $12

ARTWORX
Bridge 5.0 ....$19
Cycle Knight $14
International Hockey .$14
Linkword French $16
Linkword German $16
Linkword Russian $16
Linkword Spanish $16
Strip Poker $21
Data Disk #1 Female ..$14
Data Disk #2 Male ...$14
Data Disk #3 Female ..$14

AVALON HILL
Dark Horn $19
Gulf Strike $19
NBA Basketball $25
Spitfire '40 ....$23
Super Sunday $21
SBS 1985 Team Disk ..$14
SBS 1986 Team Disk ..$14
SBS Gen. Mgr. Disk ...$19

BAUDVILLE
Blazing Paddles $23
Video Vegas $19

BERKELEY SOFTWORKS
Geos 128 ....$44
Geo-Calc 128 ....$44
Geo Dex/Desk Pack 128$44
Geo-File 128 ....$44
Writer's Workshop 128 $44
Geos 64 ....$39
*Geo-Calc $33
*Geos Desk Pak 1 ....$23
*Geo Dex $25
*Geo File $33
*Geos Font Pak 1 or 2$19 Ea.
*Geo-Programmer $44
*Geo-Publish $44
*Geo-Spell $19
*Writer's Workshop ...$33
*Requires Geos 64!

BETTER WORKING
Business Form Shop ..$25

BRODERBUND
Carmen Sandiego: USA $25
Carmen Sandiego: World $23
Cauldron 1 & 2 ....$19
Magnetron $19
Print Shop $26
P.S. Companion $23
P.S. Graphics Library #1, #2, or #3 ...$16 Ea.
P.S. Graphics Library Holiday Edition $16
P.S. Paper Pack $14
Super Bike Challenge .$14
Toy Shop $19

CAPCOM
Ghosts & Goblins $19
Gunsmoke $19
Side Arms $19
Speed Rumbler $19

CBS
Success w/Algebra: Binomial Multiplication & Factoring $19
Success w/Algebra: First Degree & Advanced Linear Equations ..$19
Graphing Linear Functions $19
Simultaneous & Quadratic Equations ..$19
Success w/Math: Addition & Subt. ...$19
Success w/Math: Decimals: Add. & Subt.$19
Success w/Math: Decimals: Mult. & Div.$19
Success w/Math: Fractions: Add. & Subt.$19
Success w/Math: Fractions: Mult. & Div.$19
Success w/Math: Multiplication & Division $19

CMS
General Acct. 128 ...$119
Inventory 128 ....$49

DATA EAST
Breakthru $19
Commando $14
Ikari Warriors $19
Kid Niki $19
Speed Buggy $19
Tag Team Wrestling ...$23
Victory Road ....Call

DATASOFT
Alternate Reality: Dungeon $26
221 B. Baker St. $21
Theatre Europe $19
Tomahawk $21
Video Title Shop $21

DAVIDSON
Algeblaster $32
Math Blaster $32
Speed Reader 2 ....$32
Spell It $32
Word Attack $32

DESIGNWARE
Body Transparent $19
European Nations $19
Mission Algebra $19
Spellicopter $19
States & Traits $19

DIGITAL SOLUTIONS
Pocket Filer 2 ....$33
Pocket Planner 2 ....$33
Pocket Writer 2 ....$33
*all 3 in 1 Super Pack .$59

ELECTRIC DREAMS
Rocky Horror Show ..$14
Titanic Recovery Missions$14

ELECTRONIC ARTS
Amnesia $26
Arctic Fox $23
Bard's Tale 1 or 2 .$26 Ea.
Bard's Tale 1 or 2 Hints$9 Ea.
Chessmaster 2000 ....$26
Chuck Yaeger's AFT ..$23
Demon Stalker $21
Earth Orbit Station $21
Instant Music $21
Legacy of Ancients $21
Marble Madness $21
Monopoly $26
Outrageous Pages $33
Patton vs. Rommel $21
Pegasus $21
Rockford $21
Scrabble $23
Skyfox 2 ....$21
Skate or Die $21
Star Fleet 1 ....$26
Strike Fleet $21
World Tour Golf $21

EPYX
Boulder Dash Construction Kit $16
California Games $23
Champ. Wrestling $23
Coil Cop $16
Create A Calendar $19
Destroyer $23
Fast Load (R) $23
Movie Monster $14
Omicron Conspiracy ..$23
P.S. Graphics Scrapbook #1:Sports $16
P.S. Graphics Scrapbook #2:Off the Wall $16
P.S. Graphics Scrapbook #3:School $16
Rad Warrior $16
Spiderbot $16
Spy vs. Spy 3: Arctic Antics $16
Street Sports: Baseball $23
Street Sports: Basketball $23
Sub Battle Simulator ..$23
Summer Games $14
Summer Games 2 ....$23
Temple Apshai Trilogy .$14
Winter Games $23
World Games $23

FIREBIRD
Elite $19
Golden Path ....Call
Guild of Thieves $25
Knight Orc $25
Starglider $25
The Pawn $25
The Sentry $25
Tracker $25

GAMESTAR
Champ. Baseball $19
Champ. Basketball $23
GFL Ch. Football $23
On-Court Tennis $19
Star Rank Boxing 2 ....$19
Top Fuel Eliminator ...$19

HAYDEN
SAT Math $14
SAT Verbal $14

INFOCOM
Beyond Zork 128 ....$29
Border Zone $23
Bureaucracy 128 ....$23
Hitchhiker's Guide $19
Leather Goddesses $23
Nord & Bert Couldn't Make Head or Tail of It ...$23
Plundered Hearts $23
Stationfall $23
The Lurking Horror ...$23
Trinity 128 ....$23
Zork 1 ....$23
Zork Trilogy $39

INKWELL SYSTEMS
#184 Light Pen $44
#170 Deluxe L.P. $69
Flexidraw 5.5 ....$23
Graphics Galleria: Animals $16
Graphics Galleria: Borders & Signs $16
Graphics Galleria: Futuristic Encounters .$16
Graphics Galleria: Holiday Themes $16
Graphics Galleria: Maps of the World ..$16
Graphics Intregrator 2 .$19

KONAMI/ACTION CITY
Rush 'n Attack/Yie Ar Kung Fu $19
Yie Ar Kung Fu 2 ....$19

MICROLEAGUE
Baseball $25
Box Score Stats $16
General Manager $19
1986 Team Data Disk ..$14
1987 Team Data Disk ..$14
WWF Wrestling $25

MICROPROSE
Acrojet $16
Airborne Ranger $23
Conflict in Vietnam ..$25
Crusade in Europe ...$25
F-15 Strike Eagle $23
Gunship $23
Kennedy Approach $16
Pirates $25
Project Stealth Fighter .$25
Silent Service $23
Top Gunner $16

MINDSCAPE
Bop & Rumble $19
Bop & Wrestle $19
Defender of the Crown $23
De Ja Vu $23
Gauntlet $23
High Roller $19
Indoor Sports $19
Infiltrator $19
Infiltrator 2 ....$23
Into the Eagle's Nest ..$19
Mutants $14
Paperboy $23
Perfect Score SAT $44
Super Star Ice Hockey $23
Super Star Soccer $23
Uchi Mata Judo $19
Wizball $14
Xevious $19

MISC
Bob's Term Pro $29
Bob's Term Pro 128 ...$39
C.P. Copy 2 ....$21
Doodle $25
Final Cartridge 3 ....$43
Font Master 2 ....$29
Font Master 128 ....$35
Super Snapshot (R) $43
Superbase 64 ....$39
Superbase 128 ....$44
Superscript 64 ....$29
Superscript 128 ....$39

ORIGIN
Autoduel $32
Moebius $25
Ogre $19
Ultima 1 or 3 ....$25 Ea.
Ultima 4 ....$39
Ultima 5 ....$39

PROFESSIONAL
Fleet System 2 Plus ...$39
Fleet System 4 128 ....$47

SIMON & SCHUSTER
Chem. Lab $25
JK Lasser Money Mgr. .$25
Typing Tutor 4 ....$25

SIR TECH
Deep Space $25
Wizardry: Proving Grounds $25

SOFTWARE SIMULATIONS
Football $19
Pure Stat Baseball $25
Pure Stat College Basketball $25
*Data Disks Avail. Call

SPRINGBOARD
Certificate Maker $25
C.M. Library Vol. 1 ....$19
Newsroom $32
N.R. Clip Art Vol. 1 ...$19
N.R. Clip Art Vol. 2 ...$25
N.R. Clip Art Vol. 3 ...$19
P.S. Graphics Expander .$23

SSI
Battle of Antietam $32
Battle Cruiser $37
Battle Group $37
B-24 ....$23
Eternal Dagger $25
Gettysburg $37
Kampfgruppe $37
Phantasie 1, 2 or 3 $25 Ea.
President Elect 1988 ...$16
Realms of Darkness ...$25
Rings of Zilfin $25
Roadwar 2000 ....$25
Roadwar Europa $25
Shard of Spring $25
Shiloh: Grant's Trial ...$25
Sons of Liberty $23
War Game Const. Set ..$19
War in S. Pacific $37
Warship $37
Wizard's Crown $25

SUBLOGIC
Flight Simulator 2 ....$32
F.S. Scenery Disks ....Call
Jet $26

THREE SIXTY
Dark Castle $23

TIMEWORKS
Accts. Payable $33
Accts Receivable $33
*Data Manager 2 ....$19
Data Manager 128 ....$33
Desktop Publisher $33
General Ledger $33
Partner 64 (R) $33
Partner 128 (R) $39
*Swiftcalc/Sideways (D) .$19
Swiftcalc/Sideways 128 .$33
Swift Tax '87 ....$33
Sylvia Porter's Personal Fin. Planner 64 ....$33
Sylvia Porter's Personal Fin. Planner 128 ....$39
*Word Writer 3 ....$33
Word Writer 128 ....$33
*All 3 titles in 1 Pk. ...$59

UNICORN
Decimal Dungeon $19
Fraction Action $19
Percentage Panic $19
Race Car Rithmetic ...$19
Ten Little Robots $19

UNISON WORLD
Art Gallery 1 or 2 .$16 Ea.
Art Gallery: Fantasy ...$16
Print Master Plus $23

WEEKLY READER
Stickybear Series: ABC's $16
Stickybear Series: Math 1 or 2 ....$16 Ea.
Stickybear Series: Numbers $16
Stickybear Series: Opposites $16
Stickybear Series: Reading $16
Stickybear Series: Shapes $16
Stickybear Series: Spellgrabber $16
Stickybear Series: Typing $16

ACCESSORIES
Animation Station $49
Bonus SS, DD ...$4.99 Bx.
Bonus DS, DD ..$5.99 Bx.
Compuserve Starter Kit $19
Disk (Case (Holds 75) $6.88
Disk Drive Cleaner ..$6.88
Dow Jones News Retrieval Membership Kit $14
Epyx 500 XJ Joystick ...$14
Icontroller $14
Suncom Joysticks ....Call
Wico Bat Handle $17
Wico Boss $12
Wico Three-Way $19.95
XETEC Super Graphix .$59
XETEC Super Graphix Jr.$39
*All programs on disk unless otherwise noted!

P.O. BOX 111327—DEPT. CG—BLAWNOX, PA 15238

*Please Read The Following Ordering Terms & Conditions Carefully Before Placing Your Order: Orders with cashiers check or money order shipped immediately on in stock items! Personal & Company checks, allow 3 weeks clearance. No C.O.D.'s! Shipping: Continental U.S.A.-Orders under $100 add $3; free shipping on orders over $100. AK, HI, FPO, APO-add $5 on all orders. Canada & Puerto Rico-add $10 on all orders. Sorry, no other International orders accepted! PA residents add 6% sales tax on the total amount of order including shipping charges. CUSTOMER SERVICE HOURS: Mon.-Fri. 9 AM-5:30 PM Eastern Time. REASONS FOR CALLING CUSTOMER SERVICE—412-361-5291 (1)Status of order or back order (2)if any merchandise purchased within 60 days from S.D.of A. is defective, please call for a return authorization number. We will not process a return without a return auth. #! Defective merchandise will be replaced with the same merchandise only. Other returns subject to a 15% restocking charge! After 60 days from your purchase date, please refer to the warranty included with the product purchased & return directly to the manufacturer. Customer service will not accept collect calls or calls on S.D.of A.'s 800# order lines! ORDER LINE HOURS: Mon.-Fri. 9 AM-5:30 PM, SAT 10 AM-4 PM EASTERN TIME. Because this ad had to be written 2-3 mos. before it was published, prices & availability are subject to change! New titles are arriving daily! Please call for more information.

MISC

- Most Cables $15
- Casio CZ 700 ..... $275
- Casio CZ 101 ..... $250
- Digiview Stand $50
- AB Switch $30
- Mouse Pad $6
- Time Saver $60
- 256K Buffer $80

commodore

Educator 64

$124.95*

Completely Commodore Compatible REFURBISHED

FULL 90 DAY WARR.

commodore

5 MEG Hard Drives

9060

$299.95

REFURBISHED

DISK ACCESS

DISK CASE (75)	6.95
DISK FILE (120)	9.95
DISK NOTCHER	4.95
3½ DATA CASE (30)	7.95
TEAK 5¼ DISK HOLDER (50)	19.95
TEAK 5¼ DISK HOLDER (90)	29.95
TEAK 3¼ DISK HOLDER (30)	19.95
DISK HEAD CLNR.	8.00

AMSTRAD

PC1512

$599.95

SYSTEM INCLUDES:

- 512K
- CGA Card
- 360K DOUBLE SIDED Drive
- 8086 (8 MHZ)
- Mouse
- Parallel & Serial Port
- Joystick Port
- Tilt Mon. Stand
- XT Style Keyboard
- Quartz Clock

FREE SOFTWARE!
(500.00 Value)
DOS 3.2 ● GEM Desktop
GEM Paint ● GEM Doodle
Basic 2

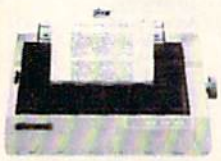
PLUS ● Paper White ● Monochorome Mon. (16 Grey Levels)

SEAGATE W/Controller
20 MEG $275* 30 MEG $315*
*Package Pricing

- Zucker Boards ..... $Call
- 1.2 MEG Drives $119.95
- Multi Function Cards .... $Call

star micronics

PRINTERS



NP10* $125.00

NX10	*$149.95
NX10C	*$159.95
NB2410	*$379.95
NR15	*$419.95
NX15	*$279.95
Powertype	*$159.95
NX 1000	*$CALL

W/2 Ribbon Purchase

AMIGA NEW RELEASES

Terrorpods	$ 24	Dark Castle	$ 24	Deluxe Print II	$ 48	Earl Weaver Baseball	$ 30
Test Drive	$ 27	Heac Coach	$ 30	Word Perfect	$200	Kind Words	$ 60
Ferrari Formula One	$ 30	Gold Runner	$ 24	The Works	$120	Page Flipper	$ 30
Fire Power	$ 15	Silver	$108	Empire	$ 32	Videoscape 3D	$120
Testcraft Plus (in stock)	$ 48	Hollywood Poker	$ 24	Studio Magic	$Call	Lurking Horror	$ 30

MOST SOFTWARE 40% OFF

FREE T-Shirt with the purchase of any (2) Electronic Arts Productal While Supply Lasts

Bard's Tale	$26.00
Bard's Tale II	26.00
Chessmaster 2000	26.00
World Tour Golf	20.00
Marble Madness	20.00
Pegasus	20.00
Moebius	26.00
Murder Party	23.00
Ogre	26.00
Ultima IV	38.00
Auto Duel	32.00
Scrabble	23.00
America's Cup	23.00
Artic Fox	26.00
All Classics	11.00
Amnesia	26.00
Bismark	23.00
Chuck Yeager	23.00
Tobruk	23.00
Paper Clip III	32.00
Outrageous Pages	32.00
Clue Books	$CALL
Dan Dare	14.00
Earth Orbit Station	20.00

MINDSCAPE

Defender of the Crown	$21.00
Bop and Wrestle	18.00
Super Star Hockey	21.00
High Roller	18.00
Indoor Sports	18.00
Infiltrator	18.00
Parallax	18.00
Uridium	18.00
Super Star Soccer	21.00

ACCOLADE

Hard Ball	$17.00
Apollo	$18.00
Test Drive	$18.00
Ace of Aces	18.00
Comics	18.00
Mini-Putt	18.00
Fourth and Inches	18.00

Jeopardy Wheel of Fortune Family Feud	$9.95 each

Berkeley Softworks

GEOS	$36.00
Fontpak 1	18.00
Desk Pack	21.00
Writer Work Shop	30.00
Geodex	24.00
GeoCalc/GeoFile	each 30.00
GeoCable	30.00
GEOS 128	42.00
Geopublish	42.00
Geoprogrammer	$CALL

Abacus

GEOS Tricks and Tips	9.00
Abacus Books	$SAVE
GEOS In and Out (Disk)	9.00
Basic 128	36.00
Basic 64	24.00
Cadpak 128	36.00
Cadpak 64	24.00
Chartpak 128	24.00
Chartpak 64	24.00
Cobol 64	24.00
Cobol 128	36.00
Super C Compiler 64/128	36.00
Super Pascal	36.00

ACCESS

MACH-128	$30.00
Leaderboard	24.00
Echelon	27.00
World Class Leaderboard	24.00
10th Frame	24.00
Exec Tournament	12.00
Tournament 1	12.00
Triple Pak	12.00
Famous Course	12.00

ACTIVISION

Portal	$24.00
Titanic	18.00
Labrynth	21.00
Aliens	21.00
Last Ninja	21.00
Hacker II	21.00
Top Fuel Elim	$CALL
Game Maker	24.00
Music Studio	18.00
Maniac Mansion	$21.00

SPRINGBOARD

Certificate Maker	$30.00
Newsroom	30.00
Clip Art 1	18.00
Clip Art 2	24.00

Firebird

Knight Ork	$24.00
Golden Path	$CALL
Pawn	24.00
Talking Teacher	24.00
Guild of Thieves	24.00

UNISON WORLD

Art Gallery	$16.00
Art Gallery 2	16.00
Print Master	21.00

PRECISION

Superbase 128	$60.00
Superbase 64	48.00
Superscript 128	48.00
Superscript 128	48.00
Visastar 64	54.00
Visawrite 128	54.00
Microlawyer	36.00

HI TECH

Card ware	$ 6.00
Heart ware	6.00
Party ware	6.00
Ware with all	9.00

Broderbund Software

Printshop	$25.00
Graphics Lib	15.00
Carmen Sandiego	21.00
Karateka	18.00
Print shop comp	21.00

LOGIC

Flight Sim II	$30.00
Scenery Disk	18.00
Jet	30.00
Football	24.00
Baseball	30.00
Up Periscope	20.00

TIMEWORKS

Data Mgr. 128	$36.00
Swiftcalc 128	36.00
Wordwriter 3	30.00
All business titles	36.00
Partner 64	36.00
Partner 128	42.00
Sylvia Porters	36.00
Desktop Publisher	$CALL

MicroProse

Gunship	$21.00
F-15	21.00
Project Stealth Fighter	24.00
Airborne Ranger	24.00
Silent Service	24.00
Pirates	24.00
Top Gunner	18.00

Xetec Inc

Font Master II 64	$30.00
Font Master 128	36.00

SSI

Gemstone Healer	$18.00
Gettysburg	36.00
Kampfgruppe	36.00
Phantasie 2	24.00
Ring of Zelfin	24.00
Road War 2000	24.00
Shard of Spring	24.00
Wizard's Crown	24.00

MICROPROSE

Micro L. Baseball	$24.00
General Mgr	24.00
Stat Disk	15.00
86' Team Disk	12.00

Digital Communications Associates

Super Pack 128	45.00
Pocket Writer 2/Filer Planner	36.00
Super Pack 2 (128)	60.00
Pocket Writer 128/Filer/Planner	30.00
Pocket Writer 64/Filer/Planner	24.00

EDYX

Fast Load	$24.00
Rad Warrior	$CALL
Street Sports -Baseball/Basketball	each 24.00
World Karate Champ	15.00
Summer Games II	20.00
Super Cycle	24.00
World Games	24.00
Football	20.00
Multi-Plan	24.00
Winter Games	20.00
Sub Battle	24.00
California Games	24.00
Boulder Dash Construction	$CALL
Spy vs. Spy III	$CALL

Fast Hackem	25.00
Kracker Jax	$$Save

NEW RELEASES

Infiltrator II	$21.00
CMS Acounting	from 99.95
Clue Books	$CALL
Star Dos	$49.95
Air Ralley	18.00
Dungeon Runners	18.00
Instant Music	18.00
Legacy of Ancient	18.00
Patton vs. Rommel	18.00
Skate or Die	18.00
Tomahawk	18.00
Monopoly	24.00
Mavis Beacon Typing	26.00
Beyond Zork	$30.00
King of Chicago	30.00
Thunder Chopper	18.00
Gauntlet	18.00
NBA	$CALL
Paper Boy	18.00
Nord and Bert	21.00
Mousetrap	12.00
Business Card Maker	$CALL
Xevious	18.00
Flexidraw	$CALL
Desk Pack 128	$CALL
Geospell	18.00
Geocalc 128	$CALL
Geofile 128	$CALL
Car Sign Designer	21.00
Geowrite 128	42.00
Skaritten	21.00
Ultima V	$CALL
Microleague Wrestling	$CALL
Wizardry	$CALL

commodore

OVER $100.00 FREE SOFTWARE

64C. . . . $129.95*
128. . . . . $189.95*
128D. $429.95*
C64 . . . Refurbished . . . $99.95*
*package pricing

FSD-2 . . . . . $149.95

1581 . . . . . $179.95
1541c $169.95
1571 . . . . . $219.95

MAGNAVOX 8562 . . . . . $240

1802c $189.95
2002 . . . . . $249.95
1086 . . . . . $CALL

OKIMATE 20 . . . $99.95*
1351 . . . . . $34.95*
Plug & Print . . . . . $Call
Device One Interface w/16K Buffer . . . $59.95
1764 RAM . . . $120.00
*W/Plug & Print Purchase

AMIGA 2000 . . . . . $CALL

COMMODORE AMIGA

500 SPECIAL PROMO

AMIGA 2000 C
• 2 MEG RAM A
• BRIDGE CARD L
• HARD DRIVES L

GENLOCK $200.00
1020 Drive. . . . $180.00

1680 MODEM $120.00
2002. . . . . $Call

Sidecar $700.00
256K $69.95

DIGI-VIEW $130
512K RAM $120
DIGI-VIEW Camera w/lens $200
MARAUDER II . . . $24

3½ Drive. . . $180.00
40 MEG . . . $950.00

2 MEG. . . . $349.95
20 MEG . . . $649.95

1 MEG WITH CLOCK $280.00

AUTHORIZED AMIGA SALES & SERVICE

G064 EMULATOR $59.95

MODEMS

PRACTICAL PERIPHEPHELS
5 YEAR WARR.
2400 . . . $159.95*

KISS ENGINEERING
2400 INTERNAL. . . $149.95

Avatex
1200 Baud $79.95*
1200 Internal . . . $99.95
1200 H.C. $99.95*
2400 Baud. . . $199.95*
*W/Cable Purchase

ALPS - 18 PIN 240CPS ● COLOR

$399.95

1 Year Warranty
EPSON/IBM COMP.
100 CPS L.Q.

FREE TRACTOR FEED

commodore

SX-64

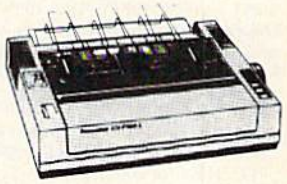
$399.95

LIMITED QUANTITY

REFURBISHED
30 DAY WARRANTY

Panasonic Industrial Company

One Panasonic Way Secaucus, New Jersey 07094



1080i-II $160*
1091i-II $180*
1092i. $280*
1592i-II $380*
1524i-II $520*
*W/2 Ribbon Purchase

SFD 1001 1 MEGA BYTE FROM $99.95

● IEEE CABLES ● VIC SWITCHES
● IEEE INTERFACES . . . . . from 79.95
● PARALLEL TO IEEE INTERFACE . . . . . $Call

Canon COLOR INK JET PJ-1080A

$299.95

LIMITED QUANTITY

REFURBISHED 30 DAY WARR.

FREE-TRIAL OFFER!

HIGH QUALITY DS/DD
DISKETTES
19¢ each
Lifetime Warranty
100% Certified W/Sleeves
Limit One (10) Pack Per Customer

SEIKOSHA

● 2 Year Warr.
*W/2 Ribbon Purch

SP180VC. $125.00*
SP1000VC $149.00*
SP1200AI. $154.00*
MP1300 (300 CPS) $275.00*
LASER . . . . . $Call

DISKS

3½ DS/DD $1.30
3½ DS/DD $1.80
SONY-FUJI-MAXWELL

5¼ DS/DD (100 Lot) . . . 25¢
5¼ DS/DD (50 Lot) . . . 60¢
BASF ● NASHUA

ORDERS ONLY 800-433-7756
IN MICH. 313-427-7713
CUSTOMER SERVICE 313-427-0267 M-F 10-6
Monday thru Friday – 10:00 A.M. to 10 P.M.
Saturday – 10:00 A.M. to 6:00 P.M. (E.S.T.)

SCHOOL P.O.'s ACCEPTED CALL FOR TERMS

M.C.S.

MICROCOMPUTER SERVICES

DEALER INQUIREES INVITED

No Surcharge for MC/VISA/DISCOVER. Sorry no walk in traffic. All returns must have RA#. Merchandise found defective will be repaired or replaced. We do not offer refunds for defective products or for products that do not perform satisfactorily. We make no guarantees for product performance. Any money back guarantee must be handled directly with the manufacturer. Call for shipping & handling info. Prices subject to change without notice.

12864 FARMINGTON ROAD, LIVONIA, MI 48150 We cannot guarantee compatibility.

Classified

SOFTWARE

FREE SOFTWARE for C64, C128, IBM & CPM
Send SASE for info (specify computer) to:
PUBLIC DOMAIN USERS GROUP
PO Box 1442-A3, Orange Park, FL 32067

FREE PUBLIC DOMAIN SOFTWARE -
Request free catalog or send $2 for sample disk and catalog (refundable). C64-128
CALOKE IND., Dept. JK, Box 18477, KC, MO 64133

More than 200 great ML routines for 64 and 128, ready to add to your own programs, in COMPUTE! Books' MACHINE LANGUAGE ROUTINES FOR THE COMMODORE 64/128. Explanations, uses, commented source code. 585 pages, $18.95. Check your local bookstore or call (800) 346-6767.

THOUSANDS OF PD PROGRAMS FOR C64/128! We have Games, Utilities, Music and much more! Information write: Lightspeed Software, POB 7037, Chesapeake, VA 23323

FREE SOFTWARE FOR C64 (disk Only)
Games, Educ, Home, Business, Utilities, Music. Free Brochure! RVH Publications, Dept. S2, 718 Alder Circle, Virginia Beach, VA 23462

RENT 64/128 SOFTWARE! 100's of disks. Lowest prices. No deposit or fee. Free catalog. Centsible Software, PO Box 930, St. Joseph, MI 49085 (616) 982-0327

TALKING SOFTWARE FOR YOUR COMMODORE! Speaking Operating System Utilities. Uses BASIC "SAY" command and SPEECH 64 (or compatible) in Expansion Port. Only $17 (includes S/H) Send to: Don Hunt, P.O. Box 3283, El Segundo, CA 90245

GM CAR - C64 USER PORT LINK
Engine trouble codes & sensor programs. Executive Electronics, Inc. 938 Main St. Yarmouthport, MA 02675 (617)362-3694

SPEEDKIT IMPROVES SPEEDSCRIPT© OR YOUR MONEY BACK! 56% discount & 30 day m.b. guarantee. Blake Lambert's SS 3.0 - 3.2 add-ons (Speedpak,-Mate,-View & 9000 word SpeedCheck dict.) all on one disk. Send $22 +$3 s/h ($6 non-US) to: Upstart G7C, POB 22022, GSO, NC 27420. MC/V ok (919)379-9732, 6–10 pm ET. SASE/phone for info. Offer expires 6/1/88

C-Net Bulletin Board for C64 & C128. The most widely used and respected BBS available. 300/1200/2400 bps, 1000 accounts, 40 message bases/file transfer SIGS, X-Modem/ Punter, a FAST M/L Editor, Online games, and MUCH MORE! SofTech Computer Systems, Inc., PO Box 23397, Lex, KY 40523 (606)268-BAUD or (800)992-SCSI for C.O.D./MC/V

FREE USER GROUP MEMBERSHIP
No fees. No dues. Disk of the month. Newsletter. Super PD library. $5 gets membership, double sided sampler disk + 40pp catalog. **QS! Alliance,** Box 1403, New Albany, IN 47150

INEXPENSIVE C64/128 ASTROLOGY & NEW AGE SOFTWARE. From $9.95 to $39.95. All print FAST hard copy. Free details. HETTIGER'S, 315 Harmony Ct "B", San Antonio, TX 78217 (512)599-6000 Visa/MasterCard accepted.

HARDWARE

Now! **Comdr Qualfd** Repairs! C64,128,SX64, Amiga,PC10,1541,1571. We buy used equipment. Fast Serv! 30 day wnty! Mom & Pop's, RR2, Box 119, Cainsville, MO 64632 (816)872-6311

WANTED: SX-64's
In GOOD CONDITION, will pay $$ TOP DOLLAR $$! Write to: Jack Kennedy, 108 East Fremont Ave., #80, Sunnyvale, CA 94087 or Collect (408) 248-8832

MISCELLANEOUS

PERSONAL COMPUTER OWNERS CAN EARN $1000 to $5000 monthly offering simple services part time. FREE list of 100 best. Write: A.I.M.P.L.
Box 60369, San Diego, CA 92106

COMPUTE!'s Gazette Classified is a low-cost way to tell over 275,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.) Inquire about display rates.

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

Closing: 3rd of the third month preceding cover date (e.g., June issue closes March 3rd). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!'s Gazette, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

advertiser's index

A Proven Achievement

GUNSHIP

Recognized by Reviewers...

"**GUNSHIP** may be the best simulation of any type ever created for the C-64... a perfect simulation for novices and aerial combat veterans alike..."
(RUN, August 1987)

"Take all your combat flight simulators and stick them in the bin. **GUNSHIP** is the ultimate..."
(Computer & Video Games Monthly, U.K., Feb. 1987)

"In terms of action alone, **GUNSHIP** is the best flying game we've ever played... It could become one of your favorites." (Family Computing, June 1987)

"**GUNSHIP**... is a very well-executed simulation, not just a game... MicroProse has another winner."
(Compute!'s Gazette, May 1987)

Acknowledged by our customers...

"... from the eye-catching packaging to the palm-drenching, flak-dodging flight visuals, **GUNSHIP** is a standard setter for the industry."
(J.R., Portland, Oregon)

"I think it's the best flight simulator I've seen for [my computer] and I've seen them all...
(L.C., Crystal Lake, Florida)

"Thank you for releasing the most AWESOME program **GUNSHIP**... should easily be the #1 program of the year..." (R.T., Chicago, Illinois)

"The **GUNSHIP** team has produced a landmark in 8-bit computers. The manual is the STANDARD."
(S.M., Pennsauken, New Jersey)

Through 5 man-years of research and development, the Gunship Design Team has crafted an unparalleled simulation experience...

- Realistic Helicopter Flight Characteristics
- True 3-Dimensional, Out-the-Cockpit Graphics
- Authentic Flight Controls and Electronics
- Accurate Weapons and Counter-Measures
- Up-to-date Enemy Threats
- Complete and Carefully Researched Documentation
- Flight and Combat Tutorials
- Hundreds of Mission Scenarios

GUNSHIP. *For C-64/128, IBM-PC/XT/AT and compatibles, Amiga, Atari ST and Apple IIGS/+/c/e. Call MicroProse or check with your dealer for specific machine availability.*

Available at a Valued MicroProse Retailer (VMR) near you. Call for locations! If product not found locally, call or write MicroProse for MC/VISA orders.

(IBM screen shown. Actual screens may vary.)

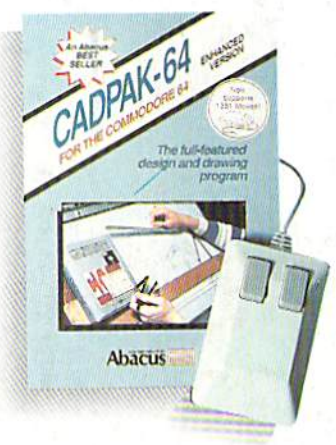
MICROPROSE™
SIMULATION • SOFTWARE

180 Lakefront Drive • Hunt Valley, MD 21030 • (301) 771-1151

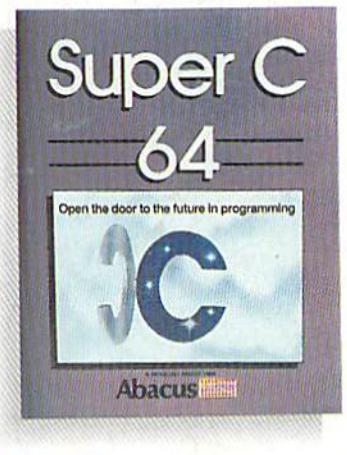
Get Results

with proven software and books from Abacus.

Super Pascal—Get the fastest and complete Pascal for your computer. **Super Pascal** is a full implementation of standard Pascal. Extensive editor features search, replace, etc. Even add machine language routines with the built-in assembler. Fast graphics library. C-64 version has high-speed DOS for faster access. More than just a compiler–**Super Pascal** is a complete system that gives you programming results.
C-64 $59.95 C-128 $59.95

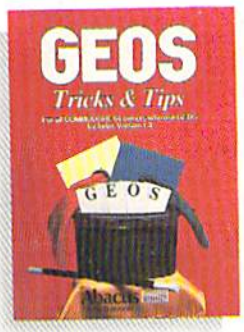


Cadpak—The professional design tool. Enter simple or intricate drawings with the keyboard, lightpen or 1531 mouse. With the flexible object editor you can create libraries of furniture, electronics, etc. as intricate as screen resolution permits. Zoom in to do detailed work. Produce exact scaled output to most printers in inches, feet, etc. Get design results fast with **Cadpak** and your Commodore® computer.
C-64 $39.95 C-128 $59.95



Super C—You can now develop software or just learn C on your computer. Easy-to-use and takes full advantage of this versatile language. Produces 6502 machine code and is many times faster than BASIC. Includes full-screen editor (search, replace and block operations), compiler, linker and handbook. Libraries for graphics and advanced math are included. Whether you want to learn C, or program in a serious C environment for your Commodore, **Super C** is the one to buy.
C-64 $59.95 C-128 $59.95

New!



BASIC Compiler— Now anyone can make their BASIC programs run faster! Easily converts your programs into fast machine language or speed code. Even speed up programs written in Simon's Basic, Video Basic etc. If your program walks or crawls, give it speed to RUN!
C64 $39.95 C128 $59.95

GEOS™ Tricks & Tips— A new book with something for everyone. Contains over 50 tricks and tips that you can use everyday. Converts any word processor file into geoWrite format; edit existing GEOS fonts or create your own; Write in machine language or explore the internals of GEOS. **$16.95**

New!

Amiga®!

TextPro Amiga is a full-function word processing package. Easy-to-use, fast and powerful—with a suprising number of extras. $79.95

BeckerText Amiga is the professional word processor. WYSIWYG formatting. Automatic creation of table of contents and index. Expandable spelling checker. Merge graphics into text. Much more. $150.00

DataRetrieve Amiga is the powerful, and easy-to-use database. Sets up in minutes. Password security. Large capacity. Performs complex searches. Very flexible. $79.95

AssemPro Amiga unlocks the full power of the Amiga's 68000 processor. Integrated Editor, Debugger, Disassembler and Reassembler. Cross-reference list. Conditional assembly. $99.95

Call (616) 698-0330 or write for your free Amiga software and book catalog. More software and books coming soon!

Please note our new address and phone numbers

Abacus

Dept. G1 • 5370 52nd Street SE • Grand Rapids, MI 49508
Phone 616/698-0330 • Telex 709-101 • Telefax 616/698-0325

Call or write today for your **free** catalog or the name of your nearest dealer. Or you can order direct using your Visa, American Express or MasterCard. Add $4.00 per order for shipping and handling. Foreign orders add $12.00 per item. 30-day money back guarantee on software. Dealer inquiries welcome—over 2400 dealers nationwide.