A Candid Interview With Commodore's Nigel Shepherd

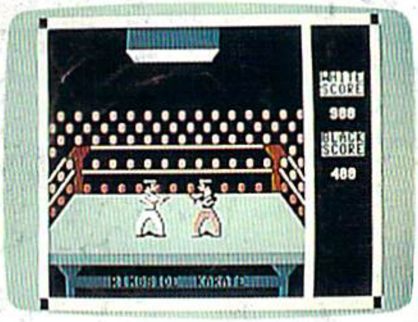
COMPUTE!'s GAZETTE

$3.00
October 1986
Issue 40, Vol. 4, No. 10
02220 $4.25 Canada

FOR **COMMODORE** PERSONAL COMPUTER USERS

Directory Extension

Enhance your 64 disk directories by adding directory descriptions and information—up to 55 characters per filename.



Also In This Issue:

Sequential File Converter For *SpeedScript*

Mastering 128 Sound And Music

Software And Hardware Reviews

And More

Math Attack

Math problems aren't supposed to be this much fun. A 64 action game and educational tool for children of all learning levels.

Ringside Karate

Match your speed, timing, and defensive skills against a friend in this outstanding, arcade-quality game for the Commodore 64.

The New Commodore Educational Software

The Commodore 64 is a popular educational tool in the classroom and at home. Here's a guide to some of the companies with new Commodore products.

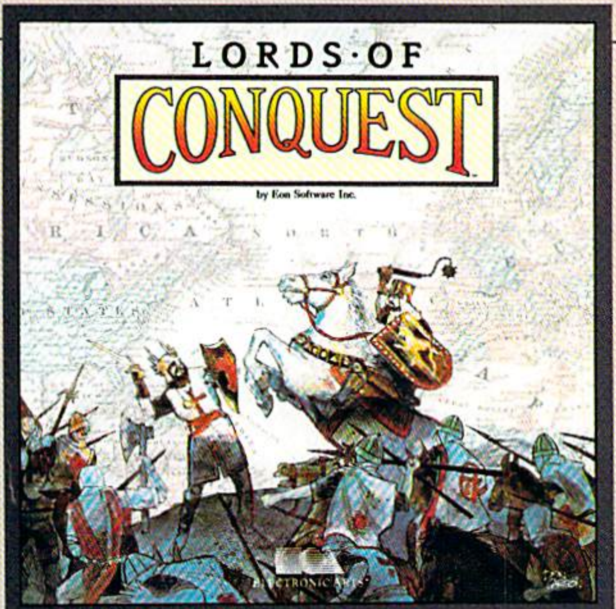
Pigs For Bucks

It's a laugh a minute as Farmer Brown hurries after his pigs to round them up for market. An innovative action game for the Commodore 128.

10
0
71486 02220

Better Than RISK™

Power-*you know you love it.*



You used to play RISK as a kid. Maybe you still do sometimes — whenever you can get enough people together.

Did you ever wish you could play by yourself? Or make changes in the map? Wouldn't it be great if you could attack North Africa from North America? Or Japan from Brazil?

Announcing Lords of Conquest™ from Electronic Arts.

It isn't RISK... it's better.

It's a conquer-the-world board game that explodes with new life and new possibilities because it uses all the power of your computer. Every battle is alive with music, color graphics, and strategic challenges impossible in a board game.

Imagine...

...having an infinite number of unique maps to choose from, so each game is different.

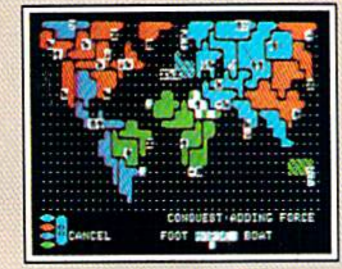
...marching your armies onto warships for attacks across oceans.

...great multi-player games, complete with treaties, trades, and treachery.

...playing challenging solo games against an intelligent computer opponent.

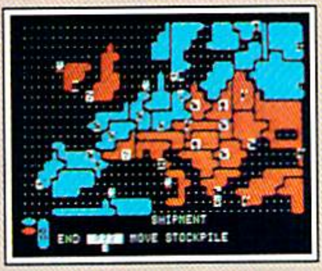
...a built-in game editor powerful enough to be called a "Strategy Game Construction Set."

Lords of Conquest. No more mucking around with cards and dice and little pieces of plastic that roll under the couch. Now you're free to concentrate on more important things. Like exterminating the opposition.



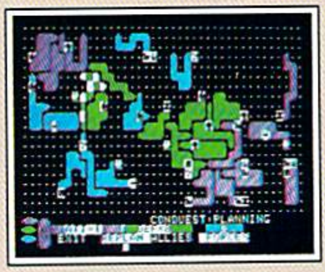
A Great Strategy Game

- *Five kinds of forces to fight with — horses, weapons, boats, territory positions, and cities.*
- *Five kinds of wealth to fight for — gold, timber, iron, coal, and herds of horses.*
- *For 1-4 players. As fun for one as it is for four.*



New Worlds to Conquer

- *An infinite variety of game maps.*
- *Create your own, or play any of 20 built-in maps like Ancient Rome, European Wars, and The World.*
- *The computer can generate unlimited random maps.*



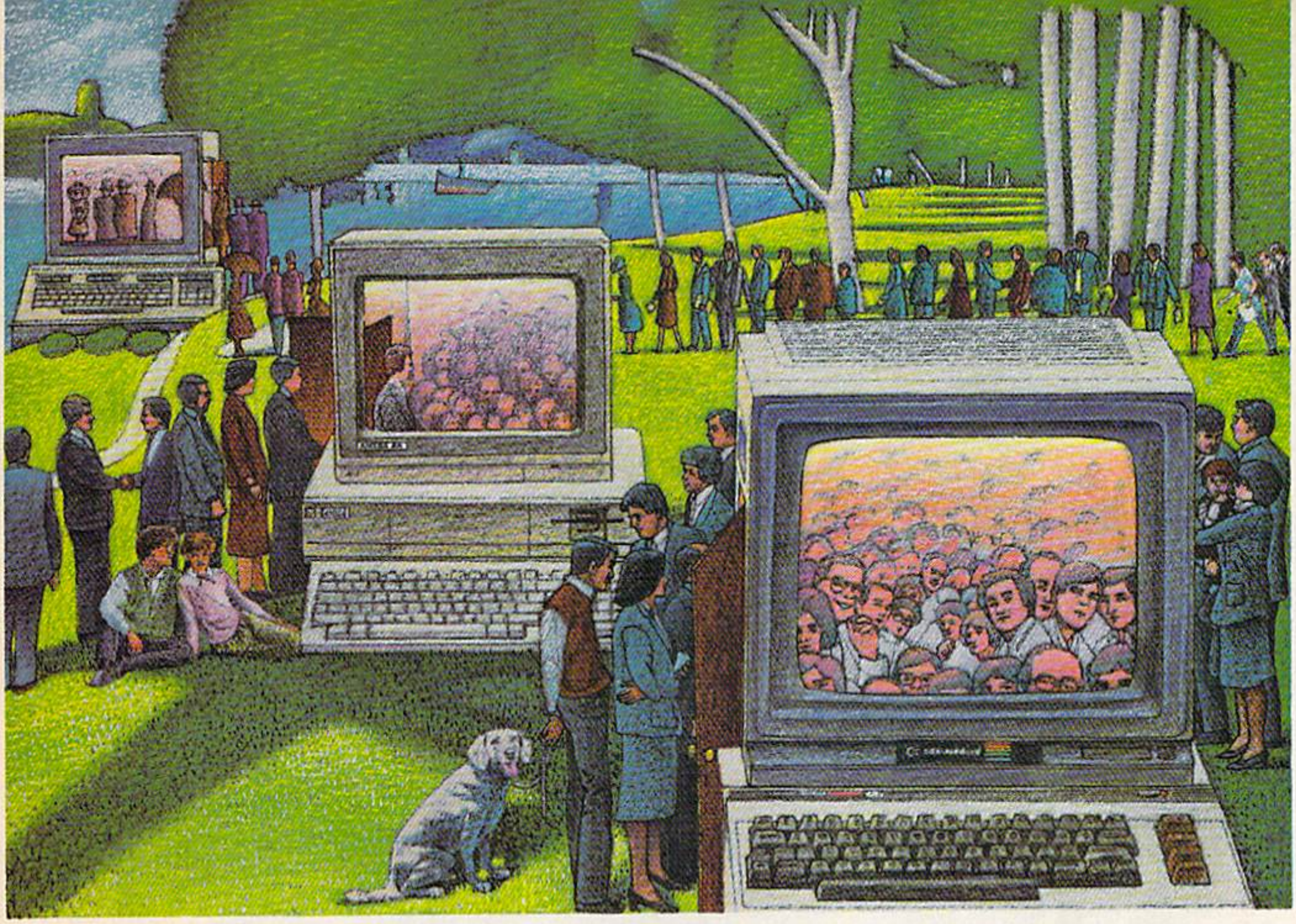
Strategy Game Construction Set

- *Variable game settings — you set the locations and abundance of resources, forces, and transport.*
- *You decide the luck factor.*
- *4 levels of complexity.*
- *9 levels of handicapping.*

ELECTRONIC ARTS™

How to order: Visit your retailer. If you are unable to find the product at your local retailer, you can call 800-245-4525 for direct VISA or Mastercard orders (in CA call 800-562-1112). The direct price is $32.95 for the Commodore version. Apple and Atari versions coming soon. To buy by mail, send check or money order to Electronic Arts Direct Sales, P.O. Box 7530, San Mateo, CA 94403. Add $5 for shipping and handling ($7 Canadian). Allow 4 weeks for delivery. There is a 14-day, money-back guarantee on direct orders. For a complete product catalog, send 50¢ and a stamped, self-addressed envelope to Electronic Arts Catalog, 1820 Gateway Drive, San Mateo, CA 94404. RISK is a registered trademark of Parker Brothers Co. Commodore is a registered trademark of Commodore Business Machines. Apple is a registered trademark of Apple computers. Atari is a registered trademark of Atari. Borderlands and Cosmic Encounter are registered trademarks of EON Software. Lords of Conquest and Electronic Arts are registered trademarks of Electronic Arts.

Lords of Conquest is based on the board game "Borderlands™" by EON — the creators of "Cosmic Encounter.™"



USE THE BRAINS YOUR COMMODORE WASN'T BORN WITH.

Right at Your Fingertips in CompuServe's Commodore® Forums

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

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

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

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

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

Easy access to free software.

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

Information you simply can't find anywhere else.

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

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

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

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

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220
800-848-8199
In Ohio, call 614-457-0802
An H&R Block Company

There seems to be some disagreement as to which one of our Commodore® programs is the most fun.

Sometimes it's tough to make everybody happy all of the time. But that's not your problem.

Mindscape makes so many great Commodore® software titles you'll surely find at least one you have to get your hands on. Maybe two. Even three.

The choice is up to you. No holds barred.

Slam. Bam. With plenty of pow. *Bop 'n Wrestle*™ puts you in the ring with a crazy collection of bone crushers. For one or two players, this first truly 3-dimensional combat sports simulation lets you climb, gouge, and claw your way to the top of the heap. Drop kicks, full nelsons, and the turnbuckle fly have never been this much fun before.

Be prepared to take evasive action. In *Infiltrator*™ by Chris Gray, you're Captain Johnny "Jimbo Baby" McGibbits, ace chopper jockey. Your mission is no day at the races. When you make it through hostile enemy airspace, this realistic flight simulation converts to covert ground-based animated graphic adventure. Now just destroy the mad leader and his angry troops. Thumbs up!

Get ready to work some magic. *Spell of Destruction*™ makes you a wizard's apprentice inside the Castle of Illusions. You must find and destroy the Prime Elemental. You're armed with a supply of spells and fireballs, but there are plenty of lessons to learn before you pass the Loremaster's challenging trial. Music and 3-dimensional graphics of 70 eerie locations make this adventure truly awesome.

Hoist a sail on your monitor. And sit down at the keys. *The American Challenge: A Sailing Simulation*™ will make you a better sailor in a few short strokes. Race against the program, another skipper head-to-head, or via modem from coast-to-coast. When you've mastered the basics you're ready to meet the Aussies and reclaim The America's Cup. Set your sails and your sights on the waters of Silicon Bay.

Without question, this quest is for you. *Fairlight*™ combines magic, adventure, strategy, and action as you meet the challenge to find the Book of Light. You're in a far away and ancient land. But 3-dimensional high-resolution graphics make it all seem real. This epic will grab you. Only the strong and resourceful will conjure up the courage it takes to survive.

Mindscape
Software that challenges the mind.
MINDSCAPE INC

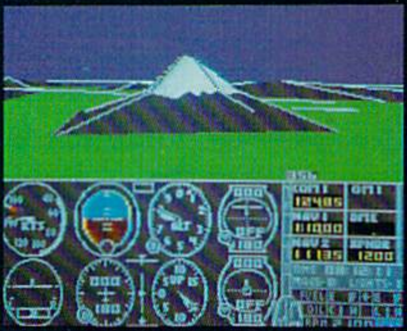
Visit your retailer or call 1-800-443-7982 (in Illinois 1-800-654-3767) for VISA or MasterCard orders. To purchase by mail, send VISA or MasterCard number with expiration date, check or money order to Mindscape, P.O. Box 1167, Northbrook, IL 60065. Add $3.00 for shipping and handling. Allow 3-5 weeks for delivery. All of the above C64/128 products are $29.95 each.

If you're an attorney read this: Commodore is a registered trademark of Commodore Electronics Ltd. Copyright © 1986 Mindscape, Inc. Mindscape is a trademark of Mindscape, Inc. *Spell of Destruction* copyright © 1985 Softek International, Ltd. *Infiltrator* software copyright © 1986 Chris Gray Enterprises, Inc. Licensed in conjunction with International Computer Group. *Bop 'n Wrestle* copyright © 1986 Beam Software. All rights reserved. Licensed in conjunction with International Computer Group. *The American Challenge: A Sailing Simulation* is a trademark of Mindscape, Inc. *Fairlight* copyright © 1985/1986 *Bo Jangeborg and the Edge*. Licensed in conjunction with International Computer Group.

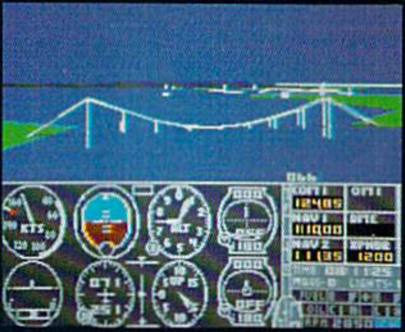
Tokyo for $19.95

東京

In the continuing evolution of scenery, SubLOGIC introduces the Japan and San Francisco Bay Area Scenery Disks for Flight Simulator II and Jet.

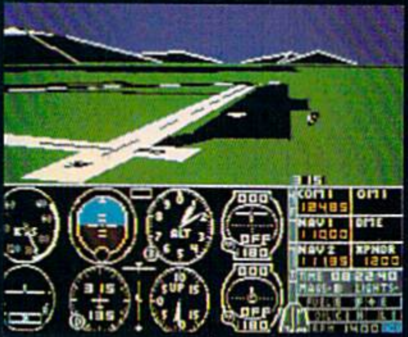


- Tokyo to Osaka is a comfortable 240-mile flight. The natural beauty of the Japanese coastline and mountain ranges complement the standard cross-country details.
- The beautiful San Francisco Bay Area "Star" Scenery Disk is perfect for concentrated sight-seeing.



- For the cross-country adventurer, our standard Western U.S. scenery (Disks 1-6) contains major airports, nav-aids, cities, highways, rivers, lakes, and mountains.

SubLOGIC Scenery Disks are available individually for $19.95. The six-disk Western U.S. set is available for $99.95. See your dealer, or write or call SubLOGIC for more information.



subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Order Line: (800) 637-4983
(except in Illinois, Alaska, and Hawaii)

GAZETTE contents

October 1986 Vol. 4, No. 10

features

reviews

games

education/home applications

programming

departments

program listings

*=General, **V**=VIC-20, **64**=Commodore 64, **+4**=Plus/4, **16**=Commodore 16, **128**=Commodore 128

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 application pending at Greensboro, NC 27403 and additional mailing offices. Entire contents copyright ©1986 by **COMPUTE!** Publications, Inc. All rights reserved, 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.

editor's notes

Since last issue, when we printed a rare combined COMPUTE!/GAZETTE editorial, we've had several interesting comments from readers. Some apparently took our comments regarding the Amiga as "Amiga-negative." We'd like to clarify. We feel, without reservation, that the Commodore Amiga is one of the most technically superior personal computers available on the market today. We also regret that the computer has not sold in greater quantities. We feel that the fact that the Atari ST has sold in greater quantities is indicative that there is a market out there to be exploited. We simply think that Commodore should (and could) have done—and hopefully will do—a better job of selling the computer. As many of you are aware, we have been steadfast supporters of these new milestones in personal computing. The COMPUTE! Books division has more titles in support of both the ST and the Amiga in print—and gets them there sooner—than any other book publisher. Our support remains strong and consistent with our reputation for quality. The fact is that to date all indications have been that the ST is outselling the Amiga by significant quantities, and that fact is not something we can alter. Frankly, we'd like to see them both selling quite well, and we continue to believe that the marketplace, properly addressed, can support such a notion. As Nigel Shepherd points out in a very excellent interview in this issue, previous sales comparisons have been comparing Atari worldwide to Commodore North America. Now that Commodore has actively begun to promote and market the Amiga in Europe, we'll see a greater increase in the installed Amiga base.

A Note For Mr. Brannon

Whether you're a recent reader or a long-established one, you've encountered the exceptional efforts of our Charles Brannon. From *SpeedScript* to *Ultrafont +*, Charles has traditionally been dedicated here to the quite supportive role of developing some of our major software applications. In a staff unique for its overall ability, Charles, as Program Editor, has been a special contributor. He first came to work with COMPUTE! as a high school student, working after school doing program listings. His first published work appeared during the fall of 1982. Since that time, Charles has been a columnist, program designer and developer, feature writer, and general chief contributor to the variety of things that we do here. Charles has provided a unique and important service to hundreds of thousands of personal computer owners through his support of our goal of offering the very best to our readers. You've enjoyed his efforts in literally dozens of contributions, and while we wish Charles well in his new employment, we want to take a moment here to thank him for those contributions.

Applications Needed; Apply Here

We're constantly seeking good material for our publications. As you know, much of what we publish is contributed by you, our readers. At the moment, we continue to be particularly interested in good Amiga material for COMPUTE!, as well as *GEOS* material for both the GAZETTE and COMPUTE!. We provide a very good payment package to our authors. See the "Author's Guide" in a recent issue.

The Nigel Shepherd Interview

We were pleased with the candor and openness exhibited by Mr. Nigel Shepherd, North American Marketing Manager for Commodore, in his interview in this issue. His comments are well worth reading, and bode well for the company and the industry.

Robert Lock

Robert C. Lock
Editor in Chief

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

Publisher James A. Casella
Founder/Editor in Chief Robert C. Lock
Senior Editor Richard Mansfield
Managing Editor Kathleen Martinek
Executive Editor Selby Bateman
Editor Lance Elko
Assistant Editor Todd Heimarck
Production Director Tony Roberts

Editors
Tom R. Halfhill, Editor, COMPUTE!'s Atari ST Disk & Magazine; Stephen Levy, Editor, COMPUTE! Books Division; Ottis R. Cowper, Technical Editor

Assistant Editors
Gregg Keizer, Ann Davies (Books); George Miller, Dale McBane, (Technical); Philip Nelson (COMPUTE! Magazine); Kathy Yakal, Assistant Features Editor; Tammie Taylor, Copy Editor; Mark Tuttle, Submissions Reviewer

Editorial Programmers
Patrick Parrish (Supervisor), Tim Victor, Kevin Mykytyn, Tim Midkiff

Programming Assistants
David Florance, David Hensley, Troy Tucker

Administrative Staff
Executive Assistant, Debi Nash; Julia Fleming, Iris Brooks, Mary Hunt, Sybil Agee

Production
Irma Swain, Production Manager; Janice Fary, Art & Design Director; Lee Noel, Assistant Editor, Art & Design; De Potter, Mechanical Art Supervisor; Terry Cash, Carole Dunton, Typesetting

Artists
Dabney Ketrow (Publications); Harry Blair, Illustrator

Associate Editors
Jim Butterfield (Toronto), Fred D'Ignazio (Alabama)

Customer Service
Diane Longo, Customer Service Manager; Orchid Tamayo, Dealer Sales Supervisor; Cassandra Greene, Customer Service Supervisor

Receptionist, Anita Armfield

Warehouse Manager, John Williams

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

Promotion
Caroline Dark, Promotion Assistant

Advertising Sales
Richard J. Marino, Vice President, Advertising Sales; Peter Johnsmeyer, Director of Advertising Sales; Bernard J. Theobald, Jr., Associate Advertising Director; Kathleen Hanlon, Production Coordinator

Sales Representatives

Jerry Thompson	415-348-8222
Phoebe Thompson	408-356-4994
Lucille Dennis	415-348-8222
Andrew Eisenberg	212-887-8474
Harry Blair	919-275-9809

Address all advertising materials to:
Kathleen Hanlon, COMPUTE!'s GAZETTE
324 West Wendover Ave., Suite 200, Greensboro, NC 27408

Sales Offices

Northeast	212-315-1665
Mid-Atlantic	212-315-1665
Southeast	212-275-9809

Sales Offices, Jules E. Thompson, Inc.
1290 Howard Avenue
Suite 303
Burlingame, CA 94010

Midwest	312-726-6047
Texas	713-731-2605
Pacific Northwest	415-348-8222
Northern CA	415-348-8222
Southern CA	213-378-8361
Arizona	213-378-8361
New Mexico	213-378-8361
Colorado	303-595-9299

Member
ABC
Audit Bureau of Circulations

COMPUTE! Publications, Inc., publishes
COMPUTE! **COMPUTE! Books** ***COMPUTE!'s GAZETTE***
COMPUTE!'s GAZETTE Disk ***Apple Applications***
COMPUTE!'s Atari ST Disk & Magazine

Editorial Office:
324 West Wendover Ave., Suite 200, Greensboro, NC 27408

Corporate Offices:
825 7th Avenue, New York, NY 10019

Customer Service:
P.O. Box 5038, F.D.R. Station, New York, NY 10150
Telephone: (In NY) 212-887-8525;
(In U.S.) Toll free 1-800-346-6767
Office Hours: 8:30 AM to 4:30 PM Monday–Friday

Subscription Orders
COMPUTE!'s GAZETTE
P.O. Box 10957, Des Moines, IA 50340

TOLL FREE
Subscription Order Line
1-800-247-5470
In IA 1-800-532-1272

COMPUTE!'s GAZETTE
Subscription Rates

(12 Issue Year): US (one year) $24. Canada, Mexico and Foreign Surface Mail $30. Foreign Air Mail $65.

The COMPUTE!'s GAZETTE subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE!'s GAZETTE, P.O. Box 10958, Des Moines, IA 50950. Include a note indicating your preference to receive only your subscription.

Authors of manuscripts warrant that all materials submitted to COMPUTE!'s GAZETTE are original materials with full ownership rights resident in said authors. By submitting articles to COMPUTE!'s GAZETTE, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **COMPUTE!** Publications, Inc. No portion of this magazine may be reproduced in any form without written permission from the publisher. Entire contents copyright © 1986 **COMPUTE!** Publications, Inc. Rights to programs developed and submitted by authors are explained in our author contract. Unsolicited materials not accepted for publication will be returned if author provides a self-addressed, stamped envelope. Where programs are included in an article submission, a tape or disk must accompany the submission. Printed listings are optional, but helpful. Articles should be furnished as typed copy (upper and lowercase, please) with double spacing. Each article page should bear the title of the article, date, and name of the author. **COMPUTE!** Publications, Inc., assumes no liability for errors in articles or advertisements. Opinions expressed by authors are not necessarily those of **COMPUTE!** Publications, Inc. **COMPUTE!** Publications assumes no responsibility for damages, delays, or failure of shipment in connection with authors' offer to make tape or disk copies of programs published herein.

PET, CBM, VIC-20, Commodore 64, Plus/4, 16, and 128 are trademarks of Commodore Business Machines, Inc., and/or Commodore Electronics Limited. Other than as an independent supplier of quality information and services to owners and users of Commodore products, **COMPUTE!** Publications, Inc., is in no way associated with Commodore Business Machines, Inc., or any of its subsidiaries.

COMMODORE 64 BEST SELLERS

from COMPUTE! Books

You can depend on COMPUTE! for clearly written, easy-to-use books for your Commodore 64. This assortment of titles includes some of our most frequently requested books containing many of our best ever applications, games, utilities, tutorials, and programming hints, all ready to type in and use on your Commodore 64.

All About the Commodore 64
V·O·L·U·M·E O·N·E
COMPUTE!'s FIRST BOOK OF COMMODORE 64 SOUND AND GRAPHICS
CREATING ARCADE GAMES ON THE COMMODORE 64
COMPUTE!'s Commodore Collection
VOLUME TWO
COMPUTE!'s Commodore Collection
VOLUME ONE
COMPUTE!'s Machine Language Routines for the Commodore 64

Look over this list of backlist bestsellers, and choose the titles you need to complete your library of first-rate Commodore 64 books from COMPUTE!.

COMPUTE!'s First Book of Commodore 64 Sound and Graphics

Edited, 275 pages
Clear, useful explanations of the 64's sound and graphics capabilities including tutorials and example programs.
$12.95 ISBN 0-942386-21-3

Creating Arcade Games on the Commodore 64

Robert Camp, 357 pages
A guide to creating arcade games on the 64, plus finished games to play and study.
$14.95 ISBN 0-942386-36-1

COMPUTE!'s Commodore Collection, Volume One

Edited, 208 pages
An anthology of 28 practical programs in easy-to-use form for the Commodore 64 and VIC-20.
$12.95 ISBN 0-942386-55-8

COMPUTE!'s Machine Language Routines for the Commodore 64

Edited, 255 pages
Complete machine language programs and easy-to-use routines help to make the Commodore 64 more powerful and versatile.
$14.95 ISBN 0-942386-48-5

COMPUTE!'s Commodore Collection, Volume Two

Edited, 270 pages
This second volume in COMPUTE!'s Commodore Collection series includes exciting games, sophisticated applications, versatile educational routines, and helpful programming aids for VIC-20 and Commodore 64 users.
$12.95 ISBN 0-942386-70-1

All About the Commodore 64, Vol. One

Craig Chamberlain, 289 pages
For beginning to intermediate programmers who want to develop the full potential of their Commodore 64 computers.
$12.95 ISBN 0-942386-40-X

Visit your local book or computer store for these titles. Or order directly from COMPUTE!. To order, call toll-free 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 allow 4–6 weeks for delivery.

When ordering, please include $2.00 shipping and handling per book in U.S. and surface mail or $5.00 airmail. North Carolina residents add 4.5 percent sales tax. New York residents add 8.25 percent sales tax. U.S. funds only.

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

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

feedback

Editors and Readers

Do you have a question or a problem? Have you discovered something that could help other Commodore users? We want to hear from you. Write to Gazette Feedback, COMPUTE!'s GAZETTE, P.O. Box 5406, Greensboro, NC 27403. We regret that due to the volume of mail received, we cannot respond individually to programming questions.

Almost A Command

I was experimenting with different commands on my 128 and accidentally typed QUIT. The computer responded with UNIMPLEMENTED COMMAND. Could you please explain what this means?

Bob Uhrick

When you type a BASIC keyword, the computer looks through the list of commands it knows, trying to find a match. When the matching word is found, the computer executes the command. If you misspell it or type an illegal command, the computer responds with SYNTAX ERROR.

QUIT is a special case. It's on the list of BASIC commands, but it doesn't do anything (except to trigger the UNIMPLEMENTED COMMAND error). Commodore's engineers apparently dropped whatever plans they had for QUIT. There's one other command that causes this error: OFF. We can only speculate about what these two words were supposed to do.

Don't Save To Commercial Disks

On some of the software I bought, the disks have about 400 blocks free. I'd like to know if it is safe to save programs on the empty space.

Michael Knol

We recommend against it. It might seem that there's empty space on the disk, but the software company could be using some of the free disk blocks for important information that's needed by the program (there are various ways of writing directly to disk sectors). If you saved a program over this information, you could ruin the commercial software. Also, many software companies deliberately put error sectors on the disks they sell, to prevent users from making copies. An error-ridden disk wouldn't be very reliable.

New disks cost only $2 or less, while commercial programs often cost $20–$30. Using the free space on a commercial disk wouldn't save you much money, especially when you consider that you might lose the software you paid for.

Modifying The 1520

I own the Commodore 1520 Printer/Plotter. Its device number is 6, and most programs expect the printer to be device 4. I would like to know if I can change the device number to 4, so I can use the 1520 with these programs.

Anthony J. Wajda

We have seen instructions for modifying the 1520's device number to 4. The process involves cutting a trace inside the printer (it's similar to the way you change device numbers on a disk drive). Unfortunately, we can't find the instructions, which were printed in a user group newsletter about a year ago. If any readers know the details, please reply. We'll publish the answer in an upcoming issue (please include a photograph or diagram pointing out the traces to be cut).

True, Not True

How does NOT work?

Uzair Hameed Ismail

Logically, NOT changes true statements to false, and vice versa. For example, IF NOT(A = 5) THEN GOTO 912 means if it's not true that A = 5, then go to line 912. This is equivalent to IF A<>5 THEN 912, which checks whether A is less than or greater than 5.

Occasionally you'll see NOT operating on a numeric variable, usually within a POKE statement. What NOT does to numbers is to reverse the sign and subtract one. PRINT NOT 9 will show you a result of −10. In machine language terms, this is a ones-complement operation. This is useful in machine language, but rarely needed in BASIC.

Joystick Reader For 64

I am writing a game which requires joystick input. Is there any BASIC routine or program to read the input of the joystick? I own a Commodore 64.

David Carlson

The following program reads a joystick plugged into port 2:

```
CR 1Ø DIMJ(1Ø):B=5632Ø:FORA=1T
      O1Ø:READJ(A):NEXT
AM 2Ø DATA 1,5,Ø,7,8,6,Ø,3,2,4
DJ 3Ø J=J(15-(PEEK(B)AND15))+1
      28-8*(PEEK(B)AND16)
DP 4Ø PRINTJ:GOTO3Ø
```

Lines 10 and 20 initialize the routine; they should be put at the start of the program. Line 30 reads the joystick value into the variable J. The following table shows the possible values of J and the corresponding directions:

Value of J	*Joystick Direction*
0	*center*
1	*north*
2	*northeast*
3	*east*
4	*southeast*
5	*south*
6	*southwest*
7	*west*
8	*northwest*

If the joystick button is pressed, 128 is added to the value of J (for example, if the firebutton is down and the joystick is pointing south, then J will have a value of 133). To read a joystick in port 1, change B=56320 to B=56321 in line 10. Incidentally, the value of J in this program is exactly the same as the value returned by the function JOY on the Commodore 128.

A Hidden Signature

Plus/4 and 16 owners who would like to see the names of the people who designed their computer should type SYS 52651. Four names appear—one flashing, one in reverse [and two normal].

Paul Arsenault

Thanks for the information. You'll occasionally find messages like this hidden in software.

Faster Printouts

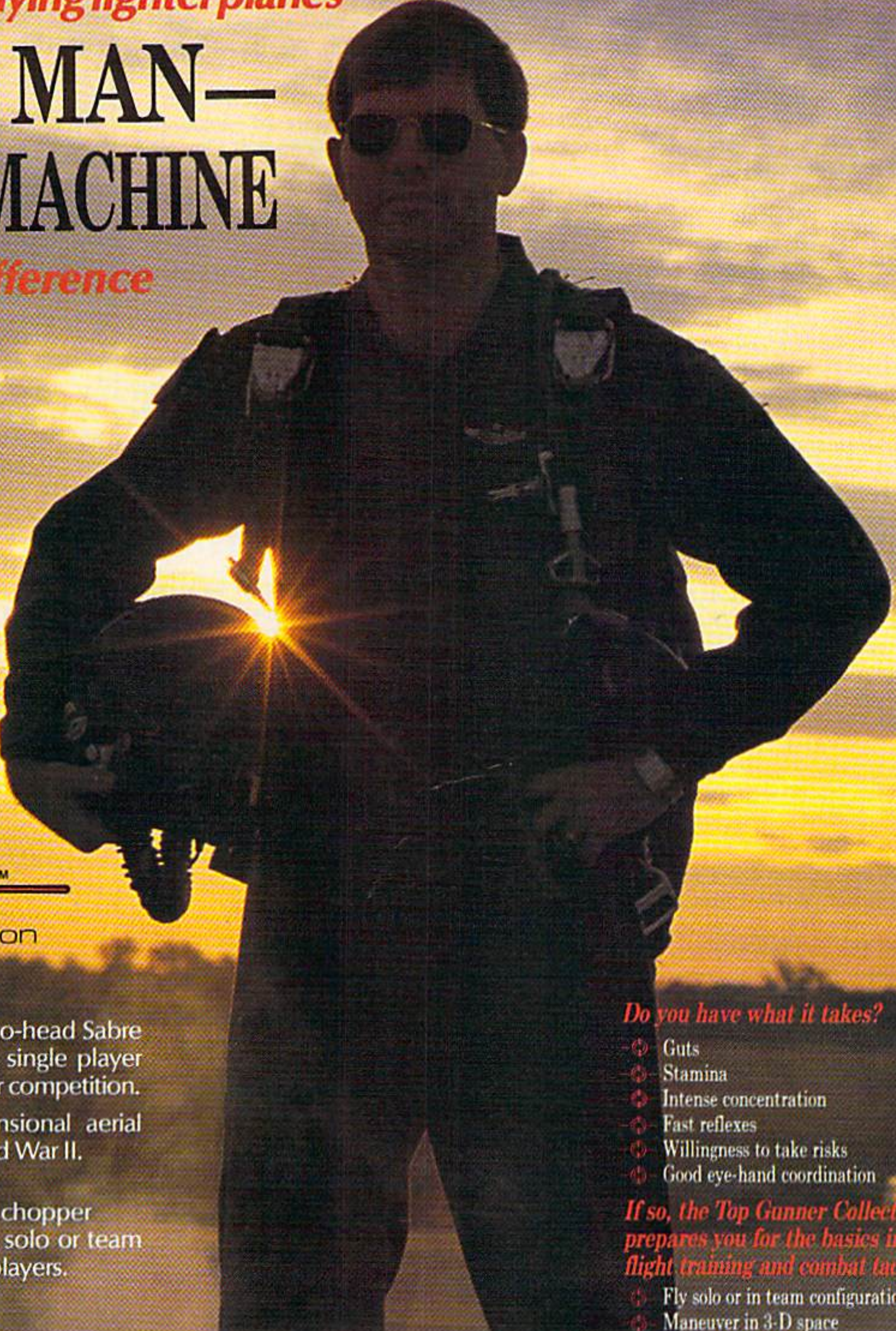
I've heard about printer buffers that are supposed to decrease the time it takes a printer to print, and also free up the computer while the printer is working. Could you explain how buffers work?

Mark R. Vogel

Printers are mechanical devices; adding a buffer won't increase the speed at which the printer runs. Buffers can save time, *When it comes to flying fighter planes*

IT'S THE MAN— NOT THE MACHINE

that makes the difference



TOP GUNNER™ Collection

MIG Alley Ace: Thrilling head-to-head Sabre Jet aerial dog fighting includes single player flying mode, as well as two player competition.

Hellcat Ace: Exciting 3-dimensional aerial combat over the Pacific in World War II. For one to four players.

Air Rescue: Sensational assault chopper rescue raids for daring pilots in solo or team configuration. For one to eight players.

Do you have what it takes?

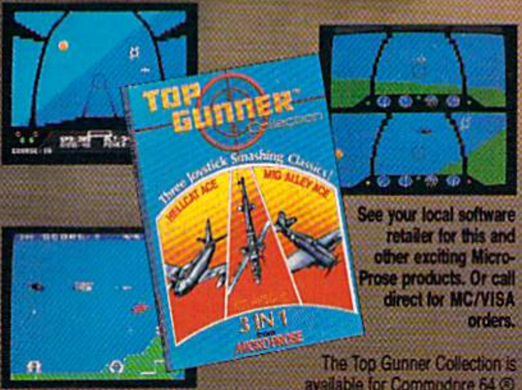
- Guts
- Stamina
- Intense concentration
- Fast reflexes
- Willingness to take risks
- Good eye-hand coordination

If so, the Top Gunner Collection prepares you for the basics in flight training and combat tactics:

- Fly solo or in team configuration
- Maneuver in 3-D space
- "See and avoid" techniques
- Advanced aerobatics: loops, rolls, G's, split S's
- Outfly the enemy

"You'll experience the wind in your face and the intense dog fighting action of some of the world's most honored and respected combat aviators: the original stick-and-rudder fighter pilots!"

MAJOR BILL STEALEY,
U.S.A.F. Reserve
President, MicroProse
(over 3,000 flying hours)



See your local software retailer for this and other exciting MicroProse products. Or call direct for MC/VISA orders.

The Top Gunner Collection is available for Commodore 64 ® and Atari XL/XE computers at a suggested retail of only $24.95.

Screen shots from Commodore 64

Commodore 64 and Atari are registered trademarks of Commodore Electronics, Ltd., and Atari Inc.

MICROPROSE
SIMULATION • SOFTWARE

120 Lakefront Drive, Hunt Valley, MD 21030 (301) 667-1151

though, especially if you do a lot of printing.

If you want to understand how a buffer works, it helps to know what happens when the computer tells the printer to start printing. The individual character codes are sent one by one to the printer. A computer can output the characters much faster than the printer can receive and print them, so at regular intervals the printer has to send a signal asking the computer to stop for a moment. When the printer gets caught up, it tells the computer to resume. During this series of stop-and-go communications, the computer is busy; you can't use it until the printing is finished.

A printer buffer is a section of memory that connects between the computer and the printer. You might install it inside the printer or it might be in a separate box. When the computer transmits the various characters, the buffer receives and stores them in memory as fast as the computer can send them. The process doesn't take very long, because there are no pauses. The buffer then starts feeding the characters to the printer. In the meantime, you've got control of the computer, because as far as the computer knows, it has already printed out the whole file.

Out Of Paper

I have a 64, a disk drive, and an Okidata 120 printer. When I turn on the power to all three devices and there is no paper in the printer, the disk drive does not respond to any commands. Why does this happen?

Mark Hlavac

Many printers have a built-in paper sensor that can tell when you've run out of paper. If the paper runs out in the middle of printing a document or if you're using single sheets of paper, the sensor stops the printer, sending a signal to the printer to stop output. Some printers may make a beeping noise or start blinking a light when the paper is gone.

This feature is designed to prevent the printer from printing directly on the platen. It's also useful when you accidentally run out of paper; you can add new paper and continue where you left off.

Since printers and disk drives share the serial bus, the printer's request to stop sending data affects other devices on the bus. There's usually a way to turn off the paper sensor, either by sending a command to the printer or by changing a DIP switch. Check your manual for details.

SpeedCheck Alphabetizer

In the December 1985 issue, you published "SpeedCheck," a spelling checker for the *SpeedScript* word processor. After using it for a few months, I decided to check the words against a dictionary (the book kind). But the words are arranged by the order they're entered in; they're not in alphabetical order.

I would appreciate an explanation of how to read the dictionary files, and, if possible, a program to alphabetically sort the entries. This would aid in looking for duplicate entries and checking for correct spelling.

John Edward

First a few notes about how "SpeedCheck" stores words in the files. Twenty-six files hold the words, one file for each letter of the alphabet. The filenames range from "A" to "Z". Every word in a file starts with the same letter, so the first letter is omitted. For example, if the file named "D" contains the words delete *and* delicious, *they appear in the file as* elete *and* elicious, *because the first letter is already known.*

SpeedScript *stores text as screen codes, so SpeedCheck follows this convention. The letters* a, b, *and* c *have screen codes 1, 2, and 3, rather than ASCII codes 65, 66, and 67. Some characters use the same number for both screen and ASCII codes (the character 5 is a 53 in both lists, for example). Before alphabetizing or printing to the screen, you'd have to convert the screen codes to ASCII.*

Also, there are no separators between words in the dictionary. SpeedCheck adds the number 128 to the screen code for the last character in each word. When you're reading through the file, you have to watch for numbers higher than 127. One final note: The first character in every word file is a 13, which should be ignored (this is necessary for reasons we can't go into here).

The following program reads through a SpeedCheck word file, alphabetizes the words, and prints them to the screen:

```
ES 1Ø L$="A":PRINT"READING FIL
      E ";L$
HG 2Ø OPEN1,8,2,L$
FF 3Ø FORJ=ØTO1:GET#1,A$:J=ST:
      PRINT".";
FF 4Ø IFA$>CHR$(127)THENT=T+1
PA 5Ø NEXTJ:CLOSE1
AD 6Ø PRINT:PRINT,T;"WORDS FOU
      ND":DIMW$(T)
FK 7Ø PRINT"READING WORDS INTO
       MEMORY"
MR 8Ø OPEN1,8,2,L$:GET#1,A$
DC 9Ø FORJ=1TOT
CH 1ØØ W$(J)=L$
CJ 11Ø GET#1,A$:IFA$<CHR$(128)
       THENW$(J)=W$(J)+CHR$(64
       +ASC(A$)):GOTO11Ø
AX 12Ø W$(J)=W$(J)+CHR$(ASC(A$
       )-64)
DF 13Ø PRINTW$(J),:NEXTJ:CLOSE
       1
AR 14Ø PRINT:PRINT"***ALPHABET
       IZING***"
RJ 15Ø C=INT(T/2)
BS 16Ø FL=Ø:FORJ=1TOT-C
XX 17Ø IFW$(J)<=W$(J+C)THEN19Ø
QS 18Ø TEMP$=W$(J):W$(J)=W$(J+
       C):W$(J+C)=TEMP$:FL=1
AH 19Ø NEXTJ:PRINT"*";:IF FL T
       HEN FL=Ø:GOTO16Ø
SP 2ØØ C=INT(C/2):IF C THEN 16
       Ø
MJ 21Ø PRINT:PRINT"DONE"
BC 22Ø FORJ=1TOT:PRINTW$(J),:N
       EXT
```

To read files other than "A", change the variable L$ in line 10. Lines 30–50 determine how many words are in the file by counting any character higher than 127 (note that the system's STatus variable will equal zero as long as there's more in the file, so the FOR-NEXT index variable J is continually set back to zero as long as there is more to read).

Lines 90–130 READ the words into an array, converting them to ASCII in the process. Lines 150–200 then alphabetize the list, using a shell sort, which is faster than a simple bubble sort and simpler than the faster quick sort. Line 220 prints the alphabetized list to the screen. If you prefer a printed list, add ***OPEN 4,4: CMD4*** *just before line 220; then* ***PRINT#4: CLOSE4*** *at the end of the program.*

Note that the program simply prints a list of the words in alphabetical order; it doesn't actually alphabetize the words in the disk file. Although alphabetizing the file might seem a logical thing to do, it won't improve the performance of SpeedCheck. In fact, for large dictionary files, alphabetizing the entries would almost certainly slow down checking. When checking words, SpeedCheck must read sequentially thorough the file searching for matches. Thus, the optimum arrangement for a SpeedCheck dictionary is to have the most commonly used words near the beginning of each file so that they will be found faster.

Scrambling Letters

I'm a brand-new subscriber with a 128, 1902 monitor, and 1571 drive. Would you please publish a short randomizing letters program? I like to work Scramlets (for example, CLORLS = SCROLL, or KEBRAM = EMBARK).

Robert Wessel

The program below works on all Commodore computers. It asks you to input a word and then prints ten scrambled versions. If you're trying to unscramble, enter the scrambled word and ten other scrambled versions will appear. You can then see if any look like normal English words.

```
SF 1Ø PRINT"ENTER WORD TO BE S
      CRAMBLED."
MF 2Ø INPUTW$:SZ=LEN(W$):DIM L
      $(SZ)
XH 3Ø FORJ=1TOSZ:L$(J)=MID$(W$
      ,J,1):NEXT
PM 4Ø FORJ=1TO1Ø
EG 5Ø FORK=1TOSZ:R=INT(RND(1)*
      SZ+1)
AX 6Ø TEMP$=L$(K):L$(K)=L$(R):
      L$(R)=TEMP$
AK 7Ø NEXTK
BD 8Ø FORK=1TOSZ:PRINTL$(K);:N
```

"*"BEST ADVENTURE GAME 1985" COMMODORE COMPUTING INTERNATIONAL

CHARTBUSTERS

FRANKIE GOES TO HOLLYWOOD

FRANKIE

the computer game

Frankie have sent you over 60 tasks in your journey from Mundanesville through the Pleasure Dome. Tasks ranging from the trivial, to heroic feats of skill and intelligence. Whenever you complete these challenges a bar-chart will show your increase in the various elements of your personality and Pleasure points will be awarded.....

This combination awards you the minimum requirement (99% a *complete person*) to search for the Special Door – the Door to the Ultimate Experience – the heart of the Pleasure Dome. Your goal is to become a *complete person* and to achieve this you must reach the top on the Bar Charts (when the word BANG will light up above the personality factors) and achieve a score of 87,000 Pleasure Points.

"Best adventure game 1985" Commodore Computing International

GOLD DISK COMMODORE 64/128K

FIREBIRD LICENSEES INC. P.O. BOX 49, RAMSEY, N.J. 07446 FACTSHEET/ORDER HOTLINE: 201.934.7373

COMMODORE 64/128 ARE TRADE MARKS OF COMMODORE BUSINESS MACHINES.

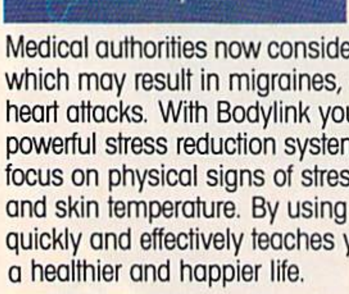
BODYLINK™

CONVERTS YOUR COMMODORE 64/128 INTO A HEALTH AND FITNESS SYSTEM.

Sports Training- A More Competitive You.

No matter what the sport, we have become a nation conscious of performing to win. We spend hours training, practicing, competing. Bodylink gives you the individual attention of the best pro—anytime you desire for as long as you wish—and at no extra cost. The Bodylink System provides instantaneous feedback on the exact manner in which your muscles are performing, enabling you to correct your swing or refine your pitch while still in motion. Bodylink allows a true two-way conversation between you and your body, giving you the winning edge.

MUSCLE COORDINATION PACKAGE—$139.95
This package includes Bodylink, Standard EMG Sensor and Lead Set, Head-Band, three electrodes and two software cartridges containing several programs.



Reducing Stress. A Healthier and More Productive You

Medical authorities now consider stress a major health risk which may result in migraines, ulcers, back pain and heart attacks. With Bodylink you have within reach a powerful stress reduction system. Bodylink allows you to focus on physical signs of stress such as muscle tension and skin temperature. By using this feedback, Bodylink quickly and effectively teaches you to reduce stress for a healthier and happier life.

STRESS REDUCTION PACKAGE—$209.95
This package includes Bodylink, Biofeedback EMG Sensor and Lead Set, Head-Band, three Electrodes, Biofeedback Temperature Sensor, and two software cartridges containing several programs.

Bicycle not included

Getting in Shape... Easier... Safer... More Effectively

Home exercise now becomes exciting and more effective. With Bodylink you can be sure you are exercising for maximum benefit. While using the aerobic exercise package, Bodylink monitors your heart rate and helps you determine the ideal level you need for aerobic gain. With the isometric muscle exerciser and training package, Bodylink guides you to do muscle developing exercises correctly and effectively. Bodylink motivates you to work harder if you are not reaching your target level, or helps you slow down if you are working too hard.

AEROBIC EXERCISE PACKAGE—$199.95
This package includes Bodylink, Standard EMG Sensor and Lead Set, Leg-Band, Pulse Rate Sensor, and two software cartridges containing several programs.

ISOMETRIC MUSCLE EXERCISE & TRAINING PACKAGE—$139.95
This package includes Bodylink, COMET, and two software cartridges containing several programs.

BODYLINK is a peripheral that plugs into the cartridge slot of the Commodore 64/128 computer. *Knowledge of computers or computer programming is NOT necessary to use BODYLINK.* You don't even need a disk drive to save your data. Various sensors are used to record internal signals from your body and relay them to BODYLINK. You simply place the sensor against the part of your body to be monitored and watch the result on your TV screen.

Commodore is a registered trademark of Commodore Electronics Limited.

DEALERS, PROGRAMMERS AND USER GROUPS
INQUIRIES WELCOME

With the BODYLINK "COMET"™ (COmputerized Muscle Exerciser and Trainer) attachment, you can build and tone the muscle groups of your stomach, chest, back, legs, and arms. COMET is a electronic muscle builder that is connected to BODY-LINK and sends a message to your TV screen when it's compressed or pulled.

Introductory Offer
Order Now and Save 20%

Offer expires September 30, 1986

DESCRIPTION	UNIT PRICE	QUANTITY	TOTAL PRICE
Muscle Coordination Package	$139.95		
Isometric Muscle Exercise & Training Package	$139.95		
Aerobic Exercise Package	$199.95		
Stress Reduction Package	$209.95		

Total Merchandise	
Deduct 20% if postmarked before end of month.	
N.Y. Residents please add applicable sales tax.	
Subtotal	
Shipping (Below $200 Add 5% of Subtotal (Over $200 Add 2.5% of Subtotal) OVERSEAS OR APO ADD 5% EXTRA	
TOTAL AMOUNT DUE	

MAIL ORDER TO:

BODYLOG, INC.
34 MAPLE AVENUE
ARMONK, N.Y. 10504

SATISFACTION GUARANTEED!
If you are not completely satisfied, you may return the products within 15 days for a full refund.

To purchase additional sensors and software separately

Call for More Information and our product catalogue

914-273-6480 or 1-800-233-2911.

Make checks payable to Bodylog, Inc.

Charge to my (   ) VISA   (   ) DISCOVER or (   ) MASTERCARD

Name ____________________

Address ____________________

City ____________ State ________ Zip ________

Signature ____________________

□□□□□□□□□□□□□□□□□

Payment must accompany order. Expires ________

All prices and specifications are subject to change without notice. Not responsible for typographic errors.

CG 9-1-86

```
      EXT:PRINT
QC 9Ø NEXTJ:PRINT:RUN
```

Scrambling is the opposite of sorting (see the previous letter), but the technique is basically the same. Instead of an IF-THEN to compare two members of a list and put them in order, you use the RND function to randomize two elements of the array, putting them in a random order.

Connecting A VCR And Computer Monitor

Could you please give me a complete set of directions on how to hook up a VCR to my 1702 monitor? Do I need special cables? Can I record and play tapes using the VCR and monitor?

Dan Andrichuk

It's quite easy to use a computer monitor with a video cassette recorder. There should be a jack on the back of your VCR labeled Video Out and another labeled Audio Out. These two outputs can be directed to the jacks called Video and Audio on the front of your 1702. Check the switch on the back to make sure it's set to the front (not the rear) connections.

The cable you use should have two RCA plugs on one end and two RCA plugs on the other. Look for the cable at a video or electronics store. The cable can also connect two VCRs together, to copy from one to another.

Translating Between Word Processors

I just upgraded from a word processor which used program files to one that uses sequential files. I would like to use my old files in the new program. Is there a program that can change them?

Kevin L. McClintock

Try running your new word processor and when you're prompted for a filename to load, type filename,p. *If that doesn't work, run your old word processor, load the file you want, and then save it with the name* filename,s. *Adding the ,p and ,s (for PRG and SEQ files) sometimes allows you to get around the incompatibility between files. You may see two additional characters at the beginning of the file when it's loaded into the new program.*

If you succeed in loading files from your old word processor, you may encounter more serious problems. Some word processors, especially ones that use PRG files, save text as screen codes, the numbers you use for POKEing to the screen. Others, especially ones that use SEQ files, use Commodore ASCII codes, the CHR$ numbers for PRINTing to the screen. If the file loads, but the characters look strange, this may be the situation. You may have to write a program that translates from one format to the other. Or check the documentation; if your old word processor saves files as screen codes, there's probably a way to create an ASCII file from within the program. You'd have to load a file from the old program and then resave it as an ASCII file.

There's one further problem you may run into, but it's minor. There's no standard for printer-formatting codes. If you use underlining, superscripts, or other such features, the codes that turn the functions on and off may turn into unpredictable characters when you transport the file from one word processor to another.

Teaching Grammar To A Computer

Could you please tell me how an adventure game parser works, and write a short demo program?

Rick Meyer

The following statements might be used in an adventure game:

GET RING
OPEN RED DOOR
ASK FRIENDLY ELF ABOUT MAP

The first has the simplest structure—a verb and an object. The second is a little more complicated because of the adjective. If a room has more than one door and the player says OPEN DOOR, the program has to be able to ask WHICH DOOR? The third statement is the most complex. ASK applies only to characters in the game (you wouldn't allow something like ASK DOOR ABOUT WIZARD, unless your game permits conversations with doors). Another consideration is that a friendly elf might be willing to talk, but a hostile elf, while able to talk, would not be amenable to chatting.

When you're writing your own adventure, it might be best to limit sentences to two words—a verb followed by an object—at least until you get the parser working. Before you start programming, make a list of verbs and objects you want to allow, and give each a number. Let's say the verbs will be GET, OPEN, and EXAMINE. At the start of the program, READ these three words into a string array:

```
10 DIM VB$(3)
20 FOR J = 1 TO 3: READ VB$(J): NEXT
30 DATA GET, OPEN, EXAMINE
```

In the main loop of the program, you'd input a string from the player, parse it (break it into individual words), and then act on the instructions:

```
400 PA$ = "": INPUT PA$: LP=
    LEN(PA$): IF LP = 0 THEN 400
410 B = 0: FOR J = 1 TO LP
420 IF MID$(PA$,J,1) = CHR$(32) THEN
    B = J - 1: J = LP + 1
430 NEXT J
440 IF B = 0 THEN PRINT "PLEASE
    ENTER A VERB AND AN OBJECT":
    GOTO 400
450 PV$ = LEFT$(PA$,B)
460 M = 0
470 FOR J = 1 TO 3
480 IF PV$ = VB$(J) THEN M = J: J =
    1000
490 NEXT J
500 IF M = 0 THEN PRINT "I DON'T
    UNDERSTAND": GOTO 400
510 ON M GOSUB 800, 900, 1000
520 GOTO 400
```

Line 400 inputs the player action (PA$) and tries again if nothing was typed. Lines 410–430 then try to find a space (CHR$(32)), because a space will mark the boundary between the verb and object (on the 128, you could substitute the INSTR$ function, which searches for the occurrence of a substring). If no space character is found, the program goes back to the INPUT line. Otherwise, the LEFT$ in line 450 extracts the verb PV$ from PA$. Finally, the verb is checked against the list previously set up (lines 470–490). If a match is found, line 510 sends the program to the appropriate routine (GET at line 800, OPEN at 900, or EXAMINE at 1000).

Line 800 will parse the remainder of the string and respond appropriately. GET DOOR is impossible. GET RING would be fine if there was a ring inside the room (you'd use another array to keep track of which objects are in which rooms). GET SNAKE might have unpleasant consequences. And so on.

Adventure games rely heavily on arrays, as you may have noticed. If you can manage the verb-object parser, you can later expand the program to include adjectives and more complex statements.

Medical Computing

I am a practicing physician and have a Commodore 128. I would be very interested in knowing if there are any programs for physicians in the way of drug use or drug interaction. I would also be interested in any kind of program for physicians in office management.

Susan K. Pyle, M.D.

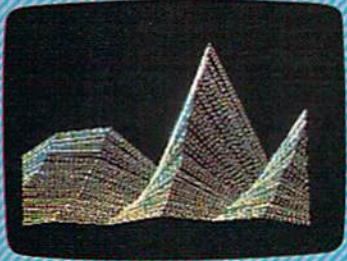
We have seen information about a service called MEDLINE, which is available on CompuServe (GO PCH). It contains indexes and abstracts from 3400 journals, with over 3.4 million references on drugs, diagnosis, surgery, and treatment programs. The fee for accessing this service is $24 per hour. There are other services of this kind as well.

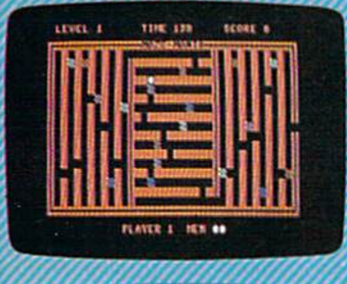
Even if the 3.4 million references were only 100 words (500 characters) apiece, around 1.7 billion bytes (1.7 gigabytes) of storage would be required. That much information would fill up more than 5000 double-sided 1571 disks, or 10,000 disks for the 1541. There may someday be a compact disc reader (CD-ROM player) which can handle such a large amount of data, but for now you can only access this kind of database by using a modem to connect with a mainframe.

A single disk would probably not hold enough information to provide a

GET UP TO 200 FUN-FILLED PROGRAMS EACH YEAR—

when you subscribe now to COMPUTE!'s Gazette





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

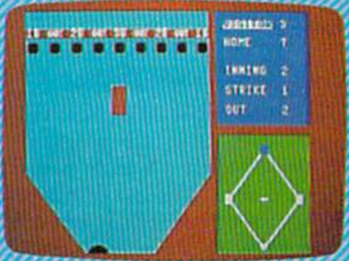
Subscribe now and you can depend on a steady supply of high quality, fun-filled programs for your Commodore computer—programs like Disk Editor, Mini-Filer, Arcade Baseball, Budget Planner, Sound Designer, *SpeedScript*, and hundreds of other educational, home finance and game programs the entire family can use all year long.

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

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

Whether you're a novice or an experienced user, COMPUTE!'s Gazette is the magazine for you. So subscribe today. Return the enclosed card or call 1-800-247-5470 (in Iowa, 1-800-532-1272).

Do it now.



ACT NOW AND SAVE!

COMPUTE!'s GAZETTE
FOR COMMODORE PERSONAL COMPUTER USERS
Computers And MIDI
Vampyre Hunter
Arcade Baseball
Super Synth
Make great sounds on your 64—the easy way.

COMPUTE!™ Publications, Inc. abc

One of the ABC Publishing Companies

If attached order card is missing, write: COMPUTE!'s Gazette P.O. Box 10955, Des Moines, IA 50950 *comprehensive and useful list of drug interactions.*

There are a couple of choices you have for medical office management: general-purpose software or vertical-market software. For word processing, you won't need a specialized program; most word processors can handle office correspondence, regardless of whether it's for a doctor's office or an art supply store. For billing and insurance claims, you could use a standard accounting package, although you might have to adjust your procedures to fit the software. For something more specific, like software written for medical offices, you'll be more likely to find a package for CP/M mode than for either 64 mode or 128 mode.

128 Text Screen Dump

Included on the Test/Demo disk for the 1571 disk drive are programs entitled "PRINT.UTIL" which provide users with a screen dump of the text screen whenever the SHIFT and CTRL keys are pressed simultaneously. Unfortunately, this program only has versions for the Commodore 64, Plus/4, and 16. Could you please provide a version for the 128?

Philip Russo

The program below prints a text screen dump of most 128 programs. The program does not print hi-res screens or sprites.

After you've entered the program, save a copy to tape or disk; then load it and type RUN. To activate the screen dump utility, type SYS 3072. Once this has been done, any time the CONTROL and SHIFT keys are pressed simultaneously, the screen is dumped to the printer. Make sure that the printer is turned on before you try to print the screen.

To deactivate the screen dump utility, press RUN/STOP–RESTORE.

```
FM 10 FORA=3072TO3238:READB:C=
      C+B:POKEA,B:NEXT:IFC<>20
      513THENPRINT"DATA ERROR"
      :END
XA 20 PRINT"{CLR}SYS 3072 TO A
      CTIVATE{20 SPACES}PRESS
      {SPACE}SHIFT+CONTROL TO
      {SPACE}PRINT"
GM 30 DATA 120,169,13,141,20,3
      ,169,12,141,21,3,88,96,1
      65,211,201,5,208,89
PA 40 DATA 216,169,128,141,26,
      208,160,255,169,4,170,32
      ,186,255,32,192,255,162,
      4
RQ 50 DATA 32,201,255,169,0,13
      3,251,141,168,12,141,166
      ,12,169,4,133,252,162,3
PE 60 DATA 160,0,177,251,32,11
      1,12,200,204,168,12,208,
      245,230,252,202,48,9,208
MG 70 DATA 238,169,232,141,168
      ,12,208,231,169,13,32,21
      0,255,169,13,32,210,255,
      169
GQ 80 DATA 4,32,195,255,32,204
      ,255,120,169,129,141,26,
      208,76,101,250,142,167,1
      2
SA 90 DATA 141,169,12,41,63,14
      ,169,12,44,169,12,16,2,9
      ,128,144,4,166,244
QR 100 DATA 208,4,112,2,9,64,3
       2,210,255,238,166,12,17
       3,166,12,201,40,208,10
KK 110 DATA 169,0,141,166,12,1
       69,13,32,210,255,174,16
       7,12,96,0
```

POKEing Around

I need to learn a lot more about my 64 to use it efficiently. Any suggestions on reading material that will explain PEEKs and POKEs?

Dan Lemke

I've been trying to write some machine language programs. Instead of having to create operations for addition, subtraction, multiplication, division, and exponentiation, I tried disassembling BASIC to find out what happens when the +, −, *, /, and ↑ keys are pressed. I've had no luck. Is it possible to use those parts of BASIC in a machine language program? If not, why?

Joseph N. Cox

What both of you are looking for is a memory map. Over the past few years, abbreviated memory maps have been published in COMPUTE!'s GAZETTE *and elsewhere. For more detailed maps, we'd suggest* Mapping the 64 *and* Mapping the VIC *from COMPUTE! Books, both of which provide a comprehensive explanation of all memory locations (*Mapping the 128 *will be available soon). Each location in RAM is described, often with PEEKs and POKEs you can try out or a short program that illustrates a programming technique in BASIC or machine language.*

In addition to the PEEKs and POKEs, you'll find information about the ROM routines that perform operations like addition, multiplication, and so on, with an explanation of setup routines that should be called first. The ROM routines are most useful in machine language programs, although BASIC programmers will find many handy SYSes which can be used in a program.

Upgrading CP/M

How can 128 owners obtain the latest revision of the CP/M system disk? I have not been able to get any information from Commodore about this.

David L. Heritage

When you boot the CP/M disk, a version date appears on the screen. The latest revision we know of is 6 DEC 1985. Other versions have a June or August release date.

The December 6 CP/M version can communicate with a modem; earlier releases did not allow you to use a modem. It also recognizes the memory expander as drive m:. If you don't use your modem in CP/M mode and don't own a memory expander, you can probably get by with the older CP/M. The memory expander comes with the latest version of CP/M, so that's one way to obtain it. Also, since the beginning of the year, the December 6 version has been included with the 128.

If you want to upgrade, there are several ways to obtain the newer CP/M. You need a CP/M program called NEWSYS.COM. When you run it, it rewrites parts of the system files CPM+ and CCP to transform them into the December 6 CP/M. NEWSYS.COM is in the public domain, and is available from many Commodore user groups and from groups that distribute public-domain software.

In addition, NEWSYS.COM is available for downloading from CompuServe and QuantumLink. Paradoxically, you can't download NEWSYS unless you have the new CP/M—and you can't have the new CP/M until you download NEWSYS. There are three ways to work around this problem. You can download in 64 or 128 mode to a Commodore formatted disk and then download a second program that reads a file from a Commodore disk and writes it to a CP/M disk. Or you can download a terminal program that runs in 128 mode, but can write to CP/M disks. Finally, if you own a 1571 disk drive, you can try to find someone who owns a modem and a Kaypro, Osborne, or Epson computer and who's willing to download NEWSYS from CompuServe (QuantumLink is Commodore-specific, and won't work with these other computers). The 1571 is able to read disks from these computers.

Character Set Mover

I am writing a program for my Commodore 64 and 128 that uses custom characters. But it takes too long to copy the character set in BASIC. Could you provide me with a machine language routine that will speed up this process? I want to move the character set to location 12288.

John Hollis

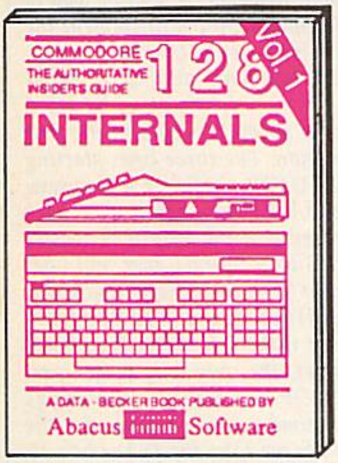
We've written two routines—one for the 64, the other for the 128—to copy the character set from ROM to RAM at 12288. To change where the character set is copied to, change the number 48 in line 30 to the page number of the character set (the memory location divided by 256).

Line 20, in both programs, contains the necessary POKEs to set up an alternate character set at location 12288, then redefines one character (the @) to become a reversed space.

Character Mover For 64

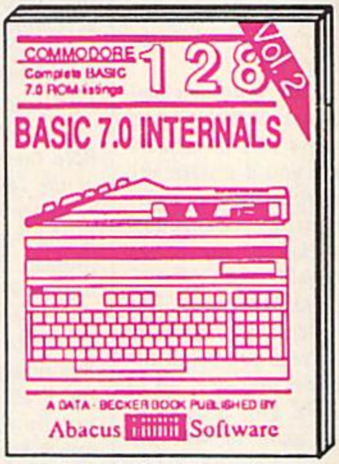
```
EB 10 FORA=828TO859:READB:POKE
      A,B:NEXT:POKE56333,127:P
      OKE1,51:SYS828:POKE1,55
```

REQUIRED READING for the Commodore 128™



COMMODORE 128 Vol. 1
THE AUTHORITATIVE INSIDER'S GUIDE
INTERNALS
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

Detailed guide presents the 128's operating system, explains graphic chips, Memory Management Unit, 80 column graphics and commented ROM listings. 500pp $19.95

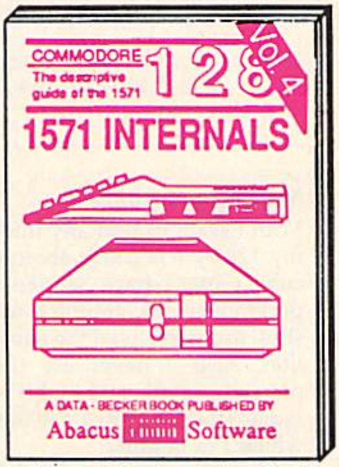


COMMODORE 128 Vol. 2
Complete BASIC 7.0 ROM listings
BASIC 7.0 INTERNALS
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

Get all the inside information on BASIC 7.0. This exhaustive handbook is complete with commented BASIC 7.0 ROM listings. Coming Summer '86. $19.95

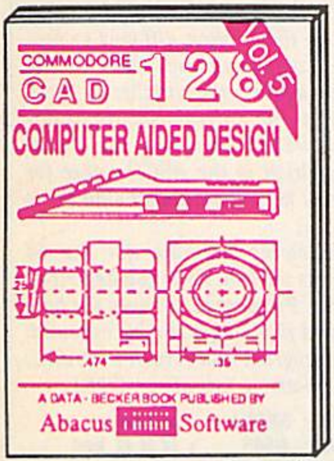
COMMODORE 128 Vol. 3
Practical & easy-to-use techniques for C-128
TRICKS & TIPS
BEST SELLER
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

Filled with info for everyone. Covers 80 column hi-res graphics, windowing, memory layout, Kernal routines, sprites, software protection, autostarting. 300pp $19.95

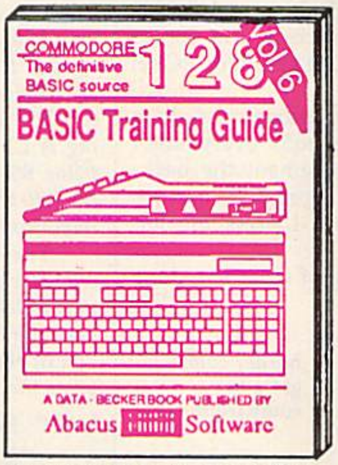


COMMODORE 128 Vol. 4
The descriptive guide of the 1571
1571 INTERNALS
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

Insiders' guide for novice & advanced users. Covers sequential & relative files, & direct access commands. Describes DOS routines. Commented listings. $19.95

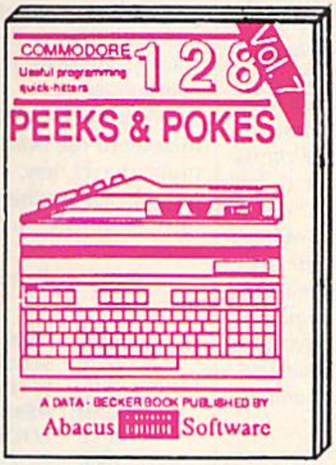


Learn fundamentals of CAD while developing your own system. Design objects on your screen to dump to a printer. Includes listings for '64 with Simon's Basic. 300pp $19.95



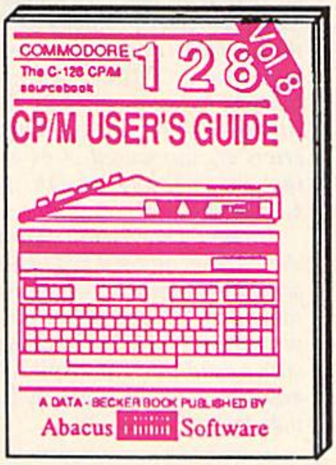
COMMODORE 128 Vol. 6
The definitive BASIC source
BASIC Training Guide
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

Introduction to programing; problem analysis; thorough description of all BASIC commands with hundreds of examples; monitor commands; utilities; much more. $16.95



COMMODORE 128 Vol. 7
Useful programming quick-hitters
PEEKS & POKES
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

Presents dozens of programming quick-hitters. Easy and useful techniques on the operating system, stacks, zero-page, pointers, the BASIC interpreter and more. $16.95



Essential guide for everyone interested in CP/M on the 128. Simple explanation of the operating system, memory usage, CP/M utility programs, submit files & more. $19.95

Commodore 128™ is a trademark of Commodore Electronics, Ltd.

Abacus Software

P.O. Box 7219 Dept. GA Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Optional diskettes available for all book titles - $14.95 each. Other books & software also available. Call for the name of your nearest dealer. Or order directly from ABACUS using your MC, Visa or Amex card. Add $4.00 per order for shipping. Foreign orders add $10.00 per book. Call now or write for your free catalog. Dealer inquires welcome - over 1400 dealers nationwide.

```
SF 2Ø POKE56333,129:POKE53272,
      28:FORA=12288TO12295:POK
      EA,255:NEXT
CR 3Ø DATA169,Ø,133,251,133,25
      3,168,169,48,133,252,169
      ,2Ø8,133,254,162,7,177,2
      53
PC 4Ø DATA145,251,136,2Ø8,249,
      23Ø,252,23Ø,254,2Ø2,16,2
      42,96
```

Character Mover For 128

```
FH 1Ø FORA=3Ø72TO31Ø3:READB:PO
      KEA,B:NEXT:BANK14:SYS3Ø7
      2:BANK15
XC 2Ø POKE26Ø4,28:POKE217,4:BA
      NKØ:FORA=12288TO12295:PO
      KEA,255:NEXT
CR 3Ø DATA169,Ø,133,251,133,25
      3,168,169,48,133,252,169
      ,2Ø8,133,254,162,7,177,2
      53
PC 4Ø DATA145,251,136,2Ø8,249,
      23Ø,252,23Ø,254,2Ø2,16,2
      42,96
```

Math Conversions

I can't seem to find any information in my *128 System Guide* about calculating cube roots. I have written about 100 programs with graphics for machine shop use, but I need the cube root function. Also, I never use the sine expressed in radians; I find if I divide the radians by 57.2958 I get the sine with which I'm familiar.

Edward Ellis

Just to the left of the RESTORE key is the up-arrow key (↑) which performs the function of raising a number to the given power. To find the cube of a number—five, for example—you can enter PRINT 5 ↑ 3. For the cube root, raise the number to the one-third power: PRINT 125 ↑ (1/3). This function operates indirectly through logarithms, so you'll find that slight rounding errors are introduced. A 64 or 128 thinks that the cube root of 216 is not 6, but 6.00000001.

Your conversion factor for radians should work. But if you'd like a little more accuracy, multiply the measurement in radians by 180 and then divide by pi. The pi character (SHIFT-↑) acts as a variable with a constant value of pi. To convert degrees into radians, multiply by pi and divide by 180.

Hi-Res Text

Besides the CHAR command, is there any other way to print text to the bitmapped graphics screen on the 128?

Greg Bennett

The CHAR command is probably the easiest way to put text on the 128's hi-res screen, but there are a couple of other techniques available. (See "Commodore 128 Hi-Res Text Manipulation" in last month's issue.)

You could store the shapes in a shape table, a list of lines and circles that define the various characters of the alphabet. You might begin with a 100 × 100 grid and, for the letter X, decide to use a line from (0,0) to (95,95) and another line from (0,95) to (95,0). The numbers in parentheses are x- and y-coordinates.

The various lines and other shapes would have to be stored in an array indexed by the ASCII value of the character. To print a word like cat, *you'd use MID$ to pick out each letter, use ASC to find its ASCII code, and then look up the shape. The DRAW and CIRCLE commands would put the characters on the screen. The advantage of using such a table is that you can easily scale the text. If you wanted the characters to fit in an 8 × 12 grid, you'd just multiply the x-coordinate by .08 and the y-coordinate by .12.*

Another way to create your own custom characters is to use GSHAPE and SSHAPE, which act as a sort of rubber-stamp function. First you'd create the character set (either with a drawing utility or with the graphics commands). SSHAPE saves a portion of the hi-res screen into a string variable and GSHAPE copies the shape back to the screen. The characters wouldn't necessarily measure eight pixels by eight pixels; you could invent characters of almost any size, up to the limits of GSHAPE and SSHAPE. Again, you'd probably want to save the shapes in an array.

Debugging Machine Language

I am trying to write a machine language program that would allow me to SYS 828 and change the screen and border colors, rather than use standard POKE commands.

What I want the program to do is increment the border color every time I press the f1 key, increment the background color when I press f3, and return to BASIC when I press the left arrow.

Here is a listing of what I've done so far:

```
      LDY  #$00
      STY  $D020  ; border color
LOOP  JSR  $FFE4  ; get a key
      CMP  #$85   ; compare to f1
      BNE  LOOP
      INY
      STY  $D020
      CMP  $5F    ; left arrow
      BNE  LOOP
      RTS
```

Dennis E. Smith

You've got the right idea, but there are a few mistakes which will prevent the program from running correctly. Let's start with GETIN, the Kernal routine for getting a keypress, located at $FFE4. The program increments the Y register (INY) to change the border color. But GETIN corrupts the Y register. The value in Y before JSR $FFE4 isn't guaranteed to be there when the computer returns, so your program may be storing some unknown value in $D020. Many of the Kernal routines affect the A, X, and Y registers. For a complete list of which routines affect which registers, see the Programmer's Reference Guide.

The following lines could be used to save the value of the Y register (TEMP is any free memory location):

```
STY TEMP
JSR $FFE4
LDY TEMP
```

For your purposes, you shouldn't have to bother with temporarily saving the Y register, however. A shorter way to accomplish the same thing (increment the border color) is to use the INC instruction: ***INC $D020****. You don't need the Y register at all.*

A second problem occurs in the first BNE instruction. The three lines starting at the label LOOP check for a keypress, compare it to the f1 key, and branch back to the beginning of the loop if f1 wasn't pressed. This loop repeats over and over until the user presses f1. So far, so good.

A few lines down, the program checks to see if the left arrow is pressed. But remember, the only way to get past the BNE LOOP line was to press the f1 key. By the time you're checking for the arrow key, it's too late; the f1 key has already been pressed. The first BNE instruction should point forward to the next CMP instruction instead of branching back to get a key. If it's not the f1 key, then see if it's the arrow key.

The line CMP $5F contains a common bug, one that's often difficult to recognize. It's comparing the value in the accumulator against the number in location $5F. What the program should be doing is CMP #$5F, comparing the actual value $5F, which is the ASCII value for the left-arrow key. The number sign (#) is important.

The following program (for the 64 only) performs the functions you described in your letter. Pressing f1 changes the border color, and f3 changes the background. To exit the program and return to BASIC, press the left-arrow key above CTRL.

```
LOOP   JSR  $FFE4  ; get key
       CMP  #$85   ; is it f1 key
       BNE  NEXT
       INC  $D020  ; increment the
                     border color
NEXT   CMP  #$86   ; is it f3
       BNE  NEXT2
       INC  $D021  ; increment the
                     background
                     color
NEXT2  CMP  #$5F   ; is it left arrow
       BNE  LOOP   ; not left arrow,
                     get another
                     key
       RTS
```

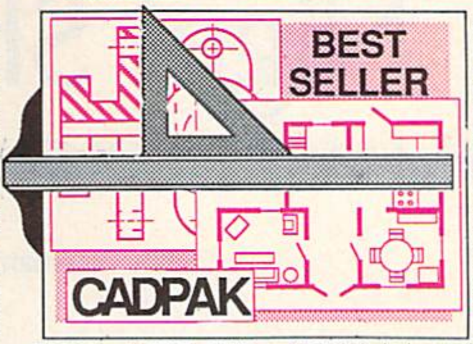
SUPER SOFTWARE

BASIC Compiler

Give your BASIC programs the speed and performance they deserve

The *complete* compiler and development package. Speed up your programs 5x to 35x. Many options: flexible memory management; choice of compiling to machine code, compact p-code or both. '128 version: 40 or 80 column monitor output and FAST-mode operation. '128 Compiler's extensive 80-page programmer's guide covers compiler directives and options, two levels of optimization, memory usage, I/O handling, 80 column hi-res graphics, faster, higher precision math functions, speed and space saving tips, more. A great package that no software library should be without.

128 Compiler $59.95
64 Compiler $39.95



Remarkably easy-to-use interactive drawing package for accurate graphic designs. New dimensioning features to create exact scaled output to all major dot-matrix printers. Enhanced version allows you to input via keyboard or high quality lightpen. Two graphic screens for COPYing from one to the other. DRAW, LINE, BOX, CIRCLE, ARC, ELLIPSE available. FILL objects with preselected PATTERNS; add TEXT; SAVE and RECALL designs to/from disk. Define your own library of symbols/objects with the easy-to-use OBJECT MANAGEMENT SYSTEM–store up to 104 separate objects.

C-128 $59.95
C-64 $39.95

Super C Language Compiler

On your C128 or C64

The language of the 80's and beyond

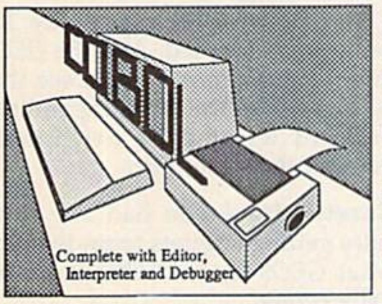
For school or software development. Learn C on your Commodore with our in-depth tutorial. Compile C programs into *fast* machine language. C-128 version has added features: Unix™-like operating system; 60K RAM disk for fast editing and compiling Linker combines up to 10 modules; Combine M/L and C using CALL; 51K available for object code; Fast loading (8 sec. 1571, 18 sec. 1541); Two standard I/O librarys plus two additional libraries—math functions (sin, cos, sqrt, etc.) & 20+ graphic commands (line, fill, dot, etc.).

C-128 $59.95
C-64 $59.95

Let your '128 or '64 speak Pascal

Not just a compiler, but a complete system for developing applications in Pascal with graphics and sound features. Extensive editor with search, replace, auto, renumber, etc. Standard J & W compiler that generates fast machine code. If you want to learn Pascal or to develop software using the best tools available–**SUPER Pascal** is your first choice.

C-64 $59.95



Complete with Editor, Interpreter and Debugger

Now you can learn COBOL, the most widely used commercial programming language, on your 128 or 64. COBOL Compiler package comes complete with syntax-checking editor, interpreter and symbolic debugging aids. New '128 version works with 40/80 column monitors and is faster than the '64 version.

C-64 $39.95
New! C-128 $59.95

Speedterm-128

Flexible, command driven terminal software package. Supports most modems for the C-128. Xmodem and Punter file transfer protocol. VT52 and VT100 terminal emulation with cursor keys, large 45K capture buffer and user definable function keys.

C-128 $59.95

Technical Analysis System for the C-64		$59.95
Technical Analysis System for the C-128	Sept. '86	$59.95
Ada Compiler for the C-64		$39.95
VideoBasic Language for the C-64		$39.95

OTHER TITLES AVAILABLE:

Chartpak

Easily create professional high-quality charts and graphs without programming. Immediately change the scaling, labeling, axis, bar filling, etc. to suit your needs. Accepts data from CalcResult and MultiPlan. '128 version has 3X the resolution of the '64 version.

C-128 $39.95
C-64 $39.95

Personal Portfolio Manager

Complete portfolio management system for the individual or professional investor. Easily manage your portfolios, obtain up-to-the-minute quotes and news, and perform selected analysis. Enter quotes manually or automatically through Warner Computer Systems. **Oct. '86**

C-128 $59.95
C-64 $39.95

Xper

XPER is the first "*expert system*" for the C-128 and C-64. While ordinary data base systems are good for reproducing facts, XPER can derive knowledge from a mountain of facts and help you make expert decisions. Large capacity. Complete with editing and reporting.

C-64 $59.95

Commodore 64® and Commodore 128™ are trademarks of Commodore Electronics, Ltd.
Unix™ is a trademark of Bell Laboratories

Abacus Software

P.O. Box 7219 Dept. GA Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Call **now** for the name of your nearest dealer. Or to order directly by credit card, MC, AMEX of VISA call **(616) 241-5510**. Other software and books are available–Call and ask for your **free** catalog. Add $4.00 for shipping per order. Foreign orders add $12.00 per item. Dealer inquires welcome–1400 nationwide.

An Interview With

Nigel Shepherd

General Manager Of Commodore North America

Selby Bateman, Features Editor

For more than nine months, Nigel Shepherd has headed all Commodore operations in the United States and Canada, a period of both ups and downs for the West Chester, Pennsylvania–based computer company. But Shepherd is no newcomer to the Commodore fold. For the seven years preceding 1986, he was the highly successful general manager of Commodore Australia, with responsibility for the international company's business in Australia and Southeast Asia. Born and educated in Scotland, Shepherd joined Commodore Australia after serving as general manager of marketing for Bell & Howell in Australia.

With extensive background in the sales and marketing of both business systems and personal computers, Shepherd has had a crucial role this year in helping Commodore's president, Thomas Rattigan, weather tough financial problems, sizeable employee layoffs, slow-starting Amiga sales, and other corporate setbacks. At the same time, he's had a hand in promoting the very successful Commodore 128 computer and the continuously phenomenal Commodore 64 in its new 64C incarnation. Those who've met and worked with him describe Shepherd as both knowledgeable and candid, with a good sense of humor.

When COMPUTE!'s GAZETTE *spoke with Shepherd, Commodore had recently announced that 128 sales were over the 600,000 mark. The new 64C, a recased 64 bundled with the icon-based GEOS operating system and QuantumLink telecommunications software, had been on the market just over a month. And Commodore officials, including Shepherd, were optimistic about increased sales of the Amiga during the third and fourth quarters of 1986.*

Gazette: As far as you can tell at this point, what's been the initial response to the new 64C among consumers and retailers?

Shepherd: I think that there's probably a two-step answer to that. If you go back perhaps two and a half to three months ago [April and May], when we were first talking about a restyled Commodore 64, I think that they were fairly blasé about it. It was almost a so-what type attitude. Even though we did talk to them about packing in with that both *GEOS* and QuantumLink, there was a fair degree of cynicism. And then, of course, as you're probably aware, we increased the price of that product.

From that until today, we've had a fairly universal acceptance now of the higher price. That earlier degree of cynicism is turned into something very positive. I think we've had a very positive response to *GEOS* and to QuantumLink.

Gazette: When did the units first begin to be available?

Shepherd: We started shipping them right in the middle of June.

Gazette: Have you received any major indication of how things are moving?

Shepherd: We certainly had a positive response from our customers in terms of buying them in. It's probably too early to say how well they're selling out to the consumer because these same customers [retailers], of course—at least some of them—were still sitting on some stocks of the previous 64. And, of course, they'd be clearing them out first.

Certainly, in things like shows where we've exposed it, we've had people come to us and say, "Hey, can we only buy the case because we've got a 64?" [Laughs.] And "We'd love to swap cases," and that type of thing.

So, in terms of appearance, it's being well accepted. And the *GEOS* thing, I think most people see that as extending the 64's capabilities beyond what they originally thought they could do.

Gazette: Have you had any difficulty getting retailers to understand what *GEOS* is, and what it potentially offers?

Shepherd: That's a bit of a 50/50. I guess, to be quite frank, we don't necessarily expect the individual guy behind the counter at [a mass market retailer] to realize the implications of *GEOS*.

However, even these major outlets—a buyer—you know, we've taken the time and the effort to explain the implications of *GEOS* to them. And the buyers have been very positive about it. And they certainly see it being plus-value for the 64.

We've also had a pretty good response from industry magazines who, again, were a little bit like our

Commodore's head of North American operations talks to the Gazette about the new 64C, *GEOS*, memory expansion for the 64, QuantumLink, prospects for the Amiga computer, and the company's financial situation.

customers, who initially started off with a fair degree of cynicism, saying we're putting a new look on the venerable Commodore 64. But I think most of the industry analogies of *GEOS*, the ones I've seen so far, have been very positive.

Gazette: What were the deciding elements in your decision, and Commodore's decision, to bundle *GEOS* with the 64C and to come out with a recased 64?

Shepherd: First, in terms of the recase—because that was really decided before the *GEOS* exercises—when we produced the 128, we felt that the 64, being that certainly it was a machine that was four-and-a-half years old—it needed a bit updating. What was a good-looking machine four years ago wasn't necessarily a good-looking machine today. We decided that there should be in appearance more synergy between the 64 and the 128. Previously, of course, there were different colors and different shapes. We were looking for a synergy so that also, with accessories, it would be possible to mix and match disk drives and printers. And whether you had a 64 system or a 128 system, you would have a color coordination.

So that was part of it. The new 64 case doesn't cost us any more, nor does it cost us any less. So there wasn't any savings in that area. It was really a question of styling. Some people do want to put a 1571 disk drive on a 64 for example, or an MPS-1000 printer or an 803

printer on a 128. So we felt that synergy would do something for us in terms of appearance, in terms of display. So that was the major reason for the new-look 64.

Gazette: How about for *GEOS* itself? That was a decision made after the case. How did that come about?

Shepherd: That was developed, of course, by Berkeley Softworks. And they came to us initially with it as an alternative operating system that they intended to market themselves. The more we looked at that, we saw that as being a fairly substantial offering for the 64. And we felt that, with the Commodore 64, if we could encourage software houses to put a front-end on the programs for *GEOS*, we could offer on the 64 icons and a mouse capability for the price of under $200 for the CPU. That was something substantial. So our decision to bundle it was really aimed at convincing the software community out here, "Look, there are substantial volumes of *GEOS* out there." And people would understand what *GEOS* was and that it would certainly be in the best interest of the software community to put an interface on the software for *GEOS*.

So that was the decision. We saw it as revitalizing the Commodore 64. It doesn't replace the Commodore 64 operating system, because you wouldn't put a *GEOS* front-end on games, for example—there's no advantage.

We saw people who wanted to get access to the current 64 operating system; of course, they still can do that. But we can transform the 64 with greater capabilities, specifically in graphics and productivity, than it had previously.

Gazette: *GEOS* is now on disk. Any chance that it will be put on a ROM chip in the future?

Shepherd: Not at this stage. It's a fairly common question. It's something at this stage that we haven't looked at. As I understand it, the only way to do it with the current board layer—but, of course, that could be changed—was really to supplant the 64 operating system with the *GEOS* operating system. And we think at this stage, we just couldn't sacrifice the operating system.

Gazette: You had mentioned that at first you received—if not negative reactions—at least some raised eyebrows about the price. How is that working out?

Shepherd: That's basically totally dissipated. Sure, there's a strong argument to say that if the street price for the 64 goes from $159 to about $199, that there has to be some form of pro rata fewer purchases. Fewer people can afford $199 than $159. It's still early to say, but that doesn't look as if that's going to materialize.

"We're looking at...RAM expansion capability on the 64, which can be accessed through *GEOS*...at the 128-mode version of *GEOS*."

Our research—and we did considerable research on this, not only on *GEOS*, but on the pricing—indicated to us that people who at this stage don't have a home computer, but in the back of their mind thought, "One of these days I'm going to get a home computer, and it's going to be good for the kids"—when we ran that survey, they indicated they thought a decent home computer would cost about a thousand dollars. Now sure, they didn't always segregate that into a CPU [central processing unit], a disk drive, and a printer. At the end of that research, when we said, "Hey, the 64 has a street price of $199; you get a disk drive for approximately the same price," they were a bit surprised. So I think the people who were going to buy a fortnight ago were the people who were conscious of $159. The people who are going to buy for the rest of this year, I don't think are really alert to the fact that the price has been increased. They just look at what's value for money.

And I think the other thing is that the retailers have never seen Commodore increase prices for something like five years. So, it was mind-boggling [Laughs]. They never thought they'd see the day Commodore would increase prices.

Gazette: We understand that there'll soon be a 128 version of *GEOS*. When might that be available? And we assume that it won't be in ROM either.

Shepherd: The 128 version of *GEOS* just now is approximately three months.The reason I say approximately is that we've got a number of things we're looking at with *GEOS*. We're looking at the possibility of putting a RAM expansion capability on the 64, which can be accessed through *GEOS*. We're looking at the 128-mode version of *GEOS*. Now what we have to determine as a company is what our priorities are. And indicate to Berkeley Softworks that we want the interface to the RAM disk first, or we want the 128 mode.

We don't see it really as being as crucial for the 128 inasmuch as the 128 already has 80-column capability. So it doesn't need *GEOS* to give it that. You can use the 64 version of *GEOS* on the 128, anyway, in 64 mode.

Nevertheless, to directly answer your question, you're looking roughly at three to four months from now [October-November]—a 128-mode version of *GEOS*.

Gazette: So the emphasis right now is memory expansion capability for the 64 through the interface from Berkeley.

Shepherd: It's that. It's also liaison with other software houses in terms of encouraging them to put *GEOS* on the front-end of their software.

Gazette: Do you have any idea when the memory expansion might be available for the 64?

UNREAL. UNSAFE.

UNBELIEVABLY FUN.

SHARD OF SPRING™, the new multiple-character role-playing game from SSI, promises an adventure unbounded by the mundane constraints of reality. It is set in a world where magic overrides the laws of physics, where monsters, gnomes and elves outnumber mere humans. Your quest is to recover the wondrous Shard, the giver of eternal Springtime. This precious crystal has been stolen by a demonic sorceress. Assemble five characters and endow them with different combinations of speed, intellect, strength, endurance, and warrior/wizardry skills. Then guide them wisely through evil dungeons and treacherous lands as they search to regain the Shard.

This fantasy game boasts a unique feature: During combat, the tactical screen is an exact zoom-in shot of where you are on the overall map.

On disk for 64K Apple® II series and C-64™. IBM® version coming soon.

In ROADWAR 2000™, you get to break a new set of rules, namely the law of safe driving. In the year 2000, bacteriological warfare has ripped apart the very fabric of American civilization. Cities have turned into gangland prizes; the highways, into battlefields. As the patriotic leader of a road gang, you are asked by what is left of the Federal Government to locate eight scientists and return them to a secret underground lab to develop a cure for the dreaded disease. As you crisscross the nation's highways on your desperate mission, you must constantly battle mutants, cannibals, and rival road gangs for new recruits, vehicles, supplies, food, gas, guns, ammunition, and medicine. In this brutal land, they are more precious than gold.

On disk for 48K Apple® II series and C-64™.

For excitement that is unreal and unbearable, check out these games at your local computer/software or game store today!

If there are no convenient stores near you, VISA & M/C card holders can order these $39.95 games directly by calling **toll-free 800-443-0100, x335.** To order by mail, send your check to: STRATEGIC SIMULATIONS, INC., 1046 N. Rengstorff Avenue, Mountain View, CA 94043. (California residents, add 7% sales tax.) Please specify computer format and add $2.00 for shipping and handling.

All our games carry a "14-day satisfaction or your money back" guarantee.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES TODAY.

Apple, Commodore, and IBM are trademarks of Apple Computer, Inc., Commodore Electronics, Ltd., and International Business Machines Corp., respectively.

SSI

©1986 by Strategic Simulations, Inc. All rights reserved.

tcp COMMODORE

ORDERS ONLY
CALL TOLL FREE **1-800-468-9044**

COMMODORE 128

ON SALE NOW!!!
1670 ... $134.95
MPS-1000 ...$247.95

C128
1571, 1572 DRIVES
1902 MONITOR
1350 MOUSE
1700, 1750 RAM EXPANSION

CALL NOW FOR LOWEST PRICES

SAVE A BUNDLE ON A BUNDLE
PACKAGE DEALS AVAILABLE
CALL FOR DETAILS

NEW COMMODORE 64C

Now bundled with GEOS and Quantrum Link!!
Still compatible with all C64 programs!!

CALL FOR OUR LOW PRICE!!

NEW 1541C DISK DRIVE!!
OUR PRICE....$CALL

CMS

General Acct System

for the C128 in 128 mode
Includes □ General Ledger □ Accts Receivable □ Billing Statements □ Accts Payable □ Check Writing □ Job Costing □ Payroll

TOTAL PRICE **$134.95**

We are a Commodore Authorized Service Center

Out of warranty repair rates:

C-64	$69.95
1541	$79.95
C-128, 1571	$CALL
1541, 1571 align	$29.95

prices include all parts and labor, 30 day warranty, and UPS return ground shipping

COMMODORE 128 SOFTWARE

For 128 in 128 or CPM modes

WORD PROCESSORS

JANE	$32.95
VIZAWRITE 128	$CALL
WORDPRO 128	$59.95
PAPERCLIP	$37.95
PAPERCLIP w/spellpack	$49.95
WORD WRITER 128 w/speller	$49.95
PERFECT WRITER	$CALL
PAPERBACK WRITER 128	$CALL
FLEET SYSTEM II w/speller	$47.95
FLEET SYSTEM III w/speller and thesaurus	$CALL
SUPERSCRIPT 128	$59.95

SPREADSHEETS

EPYX MULTIPLAN	$44.95
PERFECT CALC	$CALL
SWIFTCALC 128 w/sideways	$49.95
PAPERBACK PLANNER 128	$CALL

DATABASES

PROFILE 128	$59.95
PERFECT FILER	$CALL
DATE MANAGER 128	$CALL
PAPERBACK FILER 128	$CALL
SUPERBASE 128	$CALL

MISC 128 SOFTWARE

VIZASTAR 128	$CALL
MACH 128	$39.95
MATRIX from Prog. Periph	$47.95
DESK MANAGER 128	$34.95
C POWER 128 from Proline	$59.95

GAME GALLERY

Educational Software too!
All Games stocked for quick ship!!! Games work on 64 or 128 in 64 mode

BEST SELLING GAMES	
Leaderboard	$29.95
Bard's Tale	$32.95
Lord of Conquest	$24.95
Night Mission Pinball	$call
Hitchhiker's guide	$22.95
F-15 Strike Eagle	$21.95
Karateka	$19.95
Kung Fu (Spinnaker)	$call
Flight Simulator II	$32.95
FS II Scenery disks	$15.95
Jet	$29.95
Microleague Baseball	$27.95
Microleague general manager disk	$27.95
Microleague 1985 teams	$15.95
Hobbit	$call
Sargon III	$37.95
Murder by the Dozen	$24.95
ACCESS	
Beach Head	$24.95
Beach Head 2	$29.95
Leader Board Tourn. Disk	$16.95
ACCOLADE	
Dam Buster	$22.95
Fight Night	$22.95
Hard Ball	$22.95
Law of the West	$22.95
PSI 5 trading co	$22.95
ACTIVISION	
Alcazar	$19.95
Borrowed Time	$19.95
Computer Fireworks Celebration	$19.95
Countdown to shutdown	$19.95
Fast Tracks Slot Car Const	$19.95
Garry Kitchen's Gamemaker	$24.95
Ghostbusters	$24.95
Great Amer RR	$24.95
Hacker	$19.95
Little Computer People	$24.95
Master Of Lamps	$19.95
Mindshadow	$19.95
Pitfall II: Lost Caverns	$19.95
Space Shuttle	$19.95
AVALON HILL	
Superbowl Sunday	$26.95
Team Disk for SS	$16.95
BRODERBUND	
Championship Lode Runner	$24.95
Lode Runner	$24.95
Music Shop	$29.95
BLUE CHIP	
Baron	$24.95
Millionaire	$24.95
Tycoon	$24.95
COMMODORE	
Sky Travel	$27.95
ELECTRONIC ARTS	
Adventure Constr	$27.95
Archon	$18.95
Archon 2	$27.95
Boulder Dash	$18.95
Carriers at War	$37.95
Europe Ablaze	$37.95
Hard Hat Mack	$13.95
Heart of Africa	$24.95
Mail Order Monsters	$24.95
Movie Maker	$27.95
Music Const. Set	$17.95
Pinball Contr. set	$16.95
Racing Destr	$24.95
Reach for the Stars	$32.95
Realm of Possib	$16.95
Seven Cities of Gold	$24.95

Sky Fox	$24.95
Software Golden Oldies	$16.95
Touchdown Football	$22.95
Ultima IV	$45.95
1 on 1/Dr. J-Bird	$24.95
DATA EAST	
Karate Champ	$26.95
Kung Fu master	$26.95
EPYX	
Summer Games	$26.95
Summer Games II	$26.95
Temple Trilogy	$29.95
Winter Games	$27.95
World's Greatest Baseball	$24.95
World's Greatest Football	$28.95
World Karate Cha	$22.95

CALL FOR PRICES ON OTHER EPYX PRODUCTS

INFOCOM	
Cut Throats	$22.95
Deadline	$25.95
Enchanter	$25.95
Infidel	$29.95
Planetfall	$25.95
Sorcerer	$29.95
Spellbreaker	$29.95
Suspect	$29.95
Wishbringer	$25.95

Witness	$25.95
Zork I	$22.95
Zork II	$26.95
Zork III	$26.95
INVISICLUE BOOKS FOR ANY INFOCOM GAME	**$6.95**
MINDSCAPE	
Bank Street Music Writer	$27.95
Bank Street Storybook	$27.95
Castle Clobber	$18.95
Halley Project	$27.95
Indiana Jones	$22.95
Quake minus one	$17.95
Shadowfire	$17.95
The Perfect Score: Prep for Sat	$49.95
The Lords of Midnight	$17.95

CALL FOR PRICING ON OTHER MINDSCAPE PRODUCTS

MICROPROSE	
Acrojet	$24.95
Crusade in Europe	$27.95
Decision in the Desert	$27.95
Gunship	$24.95
Conflict in Vietnam	$call
Hellcat Ace	$21.95
Kennedy Approach	$24.95
NATO Commander	$24.95
Silent Service	$24.95

Solo Flight	$24.95
Spitfire Ace	$21.95
Top Secret	$24.95
RADARSOFT	
Maps USA	$34.95
Maps Europe	$34.95
World Maps	$34.95
RANDOM HOUSE	
Garfield	$17.95
Garfield Double Dares	$17.95
Snoopy's reading machine	$17.95
Snoopy's sky writer	$17.95
Snoopy's writer	$17.95
SIMON & SHUSTER	
G.I. Paper Airplane const	$call
JK Lasser's Income Tax guide	$39.95
Kermit's Story Maker	$19.95
NY Times Crossword Puzzle Vol. 1 or 2	$14.95
Spy Hunter	$31.95
Star Trek-Kobayashi alt	$29.95
Typing Tutor III	$29.95
SPECTRUM HOLOBYTE	
Gato	$call
SPINNAKER	
Adventure Creator	$22.95
Alphabet Zoo	$16.95
Cosmic Combat	$call
Delta Drawing	$19.95
Facemaker	$19.95
Hey Diddle Diddle	$16.95
Homework Helper	$call
In search of the most amazing thing	$18.95
Snooper Troops I or II	$18.95
SPRINGBOARD	
Early Games	$26.95
Easy As	$29.95
Piece of Cake Math	$26.95

CALL FOR PRICING ON ALL STRATEGIC SIMULATION GAMES

WICO	
Boss	$13.95
Bat Handle	$18.95
Three Way	$CALL
SPECTRAVIDEO	
Quick Shot	$6.95

JOY STICKS

C-64 BUSINESS AND PRODUCTIVITY

C-64 WORD PROCESSORS	
Paperback Writer	$call
Paperback Writer Dictionary	$19.95
Bank Street Writer	$call
Bank Street Speller	$34.95
Cut & Paste(EOA)	$17.95
Font Master II	$34.95
Wordpro 64	$36.95
Spellpro 64	$32.95
Wordpro 3 + /64	$14.95
Fleet System II	$call
Paperclip	$37.95
Paperclip w/spellpack	$49.95
Mirage Professional WP	$36.95
Trio	$call
Word Writer 64 w/speller	$34.95
Kid Pro Quo	$32.95
C-64 DATABASES	
Bank Street Filer	$34.95
Consultant	$39.95
Mirage Database w/Report gen	$36.95
Profile 64	$36.95
Data Manager	$19.95
C-64 SPREADSHEETS	
Paperback Planner 64	$call
Vizastar 64	$79.95
Pracitcal(d) or (t)	$call
Calkit	$39.95
Hes Multiplan	$29.95
PS, The Programmable Spreadsheet	$19.95

Swiftcalc 64 w/sideways	$39.95
Sideways	$19.95
Calc Result Adv (d,c)	$67.00
CARDCO	
Freeze Frame	$39.95
Numeric keypad	$34.95
5 slot cart. exp	$54.95
S'more Basic	$49.95
PROLINE	
GT4	$22.95
fast save, fast load cartridge	
C Power, C compiler	$call
CADPIC	$36.95
Cashbox	$36.95
Wordpro 64	$36.95
Spellpro 64	$32.95
Profile 64	$32.95
PAL 64	$32.95
POWER 64	$32.95
TOOLBOX 64	**$49.95**
UTILITIES	
Merlin 64	$34.95
Fast Load	$24.95
Vorpal fast loader	$call
Quest Stat manager	$34.95
Pal 64	$32.95
Power 64	$32.95
Toolbox 64	$59.95
CSM 1541 align	$34.95
Mach 5	$24.95
Mach 128	$call
C Power	$69.95
Simon's Basic	$39.95

C-64 INTEGRATED PACKAGES	
Vizastar 8K	$79.95
Trio	$call
Homepack	$39.95
MISCELLANEOUS	
Estate power supply for C-64	$54.95
Koalapad	$59.95
Koala Gibson Light pen	$49.95
Naverone 3 Slot expander	$27.95
GRAPHICS	
Print Shop	$25.95
Print Shop Companion	$24.95
Graphics Library I, II, or III	$16.95
Newsroom	$34.95
Clip Art I	$19.95
Clip Art II	$call
CADPIC	$32.95
Screen Dumper 64	$19.95
Financial & Accounting	
Softsync Pers. Acct	$32.95
Cont. Home Acct	$46.95
Timeworks General Ledger, A/R, A/P, Payroll, Inventory	$40.95 each
Cashbox	$36.95
Timeworks Electr. Checkbook	$19.95
Timeworks Money Manager	$19.95

tcp TUSSEY COMPUTER PRODUCTS

P.O. BOX 1006
STATE COLLEGE, PA 16804

CLASSIFIEDS

INFORMATION AND
PA ORDERS 814-234-2236

OPEN 9-8 Mon-Fri; 10-5 Sat EAST COAST TIME

GEOS
C64 COLOR VERSION
$49.95

PRINTERS

Epson printers
1 yr warranty

LX-80	$229
Homewriter 10	$CALL
FX-85	$359
FX-286	$CALL
RX-100	$299
DX-10	$CALL

PANASONIC
2 YR WARRANTY

1091	$229.00
1080	$199.00
1092	$CALL
1592	$CALL
3131 Daisywheel	$CALL
3151 Daisywheel	$CALL

JUKI

PRINTER/TYPEWRITER
2200 .... $264
Daisywheel Printer, Portable Typewriter, Built in Correction Tape

PRINTERS

6000P	$CALL
6100P	$344

star MICRONICS·INC

NX-10 call for price and details	
NL-10C	$CALL
SL-10C or SG-10C Commodore ready	$226
SG-15	$364
SD-10	$323
SD-15	$449
POWERTYPE 18cps daisywheel	$229
SR-10, SR-15	$CALL

PRINTER PACKAGES
all packages work with C64 or C128

star NX-10 & Xetec Supergraphix	$289.95
NX-10 & Xetec Supergraphix jr.	$275.95
PANASONIC 1091 & Xetec Supergraphix	$289.95
1080 & Xetec Supergraphix jr.	$239.95

Packages Available! All printers/interfaces. . .call for details!

SEIKOSHA
2 YR WARRANTY

SP-1000VC	$179.00
SP-1000A or SP-1000I	$199.00

Printer Closeouts!

***NEC* 8027A**
105 CPS, Friction and tractor feed Works with Print Shop!
$139

COMREX
COMREX 220 ...$119.95
Commodore Ready ... 3 month Warranty from Epson

PRINTER INTERFACES

G-WIZ	$49.95
MW-350 w/4K Buffer	$CALL
Cardco Super G...	$CALL
Xetec Supergraphix	$CALL
Xetec Jr.	$46.95
Grappler CD	$87.95
Tymac Connection	$67.95

xetec inc.
SUPER GRAPHIX interface w 8K buffer, down loadable fonts
$ lowest price
SUPERGRAPHIX jr printer interface . $46.95
Font Master II . $34.95

UW Unison World

Printmaster	$29.95
Art Gallery	$CALL

RELEVANT SOFTWARE

Icon Factory	$29.95
Billboard Maker	$29.95

ENHANCER 2000
$159 *DISK DRIVE*
1 YEAR WARRANTY

The Print Shop

Broderbund Print Shop	$25.95
Print Shop Companion	$24.95
Graphics Library I, II or III	$16.95
120 sheet color paper refill ⅓ red, ⅓ blue, ⅓ gold	$8.95

SAT Preparation
With HAYDEN
Prepared by Professionals

SAT Score Improvement System $59.95
Includes Preparation for Math, Reading comp., Vocabulary, Quantitative Comparisons, and Word problems
SAT Practice Test $17.95

DISKS
per box of 10

BONUS DISKS	
SS/DD	$7.45
DS/DD	$7.95
NASHUA DISKS	
SS/DD	$8.95
DS/DD	$9.45
TUSSEY DISKS	
SS/DD	$8.45
DS/DD	$8.95

DATA 20 XL-80

The XL-80 is an 80 column board that plugs into the back of your C-64. It's **compatible with the Mitey Mo** modem software to give you an 80 column terminal on your C-64. Included with the XL-80 is an 80 column word processor, spreadsheet, and mailing list program.

XL-80
$59.95
XL-80 & MITEY MO Package Deal
$107.95

CLOSEOUT SPECIAL!!
Wordpro 3+/64
$14.95
while supply lasts

MODEMS – MODEMS

VOLKS 6480 300/1200 BAUD	$124.95
WESTRIDGE 6420	$59.95
6480 300/1200 BAUD	$139.95
MESSENGER	$42.95
1670	$157.95
1660	$CALL
MITEY MO	$59.95
VIP TERMINAL	$34.95
VIDTEX TERMINAL	$23.95
Compuserve Starter Kit	$19.95
Playnet Starter Kit	$14.95
MASTER MODEM Includes Dow Jones, Compuserve Time	$39.95

MONITORS

COMPOSITE COLOR	
COMREX 6550 with sound & 2 yr warranty	$149.00
SAKATA SC-100	$149.00
THOMPSON 36632	$139.00
TAXAN 220 14" Separated Video, mono mode	$169.00
MONOCHROME	
ZENITH 1220 12" flat amber	$94.00
ZENITH 1230 GREEN	$94.00
GOLDSTAR 12' medium res amber or green	$79.95
GOLDSTAR 12" high res....	
PANASONIC 120 12" amber	$79.95
NEC1205 amber w/sound	$89.95
CARDCO 128/80 cable C128 to 80 composite monochrome adapter	$ 9.95
RGB/COMPOSITE	
MAGANAVOX 8562 2 yr warranty, monochrome mode	$287.00
THOMPSON 36512VC monochrome mode, separated video	$274.95
PANASONIC DT1300 Great price & Panasonic reliability	$259.95

Software orders over $50.00 will be shipped FEDERAL EXPRESS (yes, even with *these* prices)
You only pay TCP's standard shipping charge of $4.00 per order. This offer also valid on peripherals and accessories under 8 pounds. Orders arriving before 11:00 AM our time will be shipped out same day.*

Computerized order entry, processing and status allow TCP to serve you faster and better!

- If through some oversight we don't have the the lowest price, we would appreciate the opportunity to beat it. If we can, you will get the benefit of our Federal Express shipping on software orders over $100.00.
- We accept Mastercard, Visa, COD and mail orders.
- Purchase orders are accepted from qualified corporations and institutions. Minimum order of $500.00 required.
- No sales tax on orders outside of PA.
- Buy with confidence. We honor manufacturers warranty.

To order by mail: We accept money order, certified check, personal check. Allow 2 weeks for personal check to clear.
Shipping: $4.00 for software and accessories/$10.00 for printers and color monitors/$8.00 for disk drives and other monitors/Add $3.00 per box shipped COD. Call for other shipping charges. Additional shipping required on APO, FPO, AK, HI, and foreign orders.

Terms: ALL PRICES REFLECT CASH DISCOUNT, ADD 1.9% FOR MASTERCARD OR VISA. Manufacturer's warranty honored with copy of our invoice. **ALL SALES ARE FINAL.** Defective items replaced or repaired at our discretion. Pennsylvania residents add 6% sales tax. Prices and terms subject to change without notice.

ORDERS ONLY...CALL TOLL FREE **1-800-468-9044**

tcp

Shepherd: That is, at this stage, a September or October product.

Gazette: What have been the software publishers' responses to developing under *GEOS*?

Shepherd: The software publishers who are primarily interested in productivity software have been very encouraged by *GEOS* because to them, they see it almost as a second market.

They've already sold into the installed base word processors, databases, and spreadsheets. They're encouraged by the fact that apart from the new market out there, which they would have access to anyway, they believe that the implication is that *GEOS* is sufficiently strong for people who currently have word processing or other types of productivity software to buy a new version with *GEOS* capabilities. Whereas before they may have not been encouraged or enticed to buy new software. They think there are sufficient additional attributes offered by *GEOS* to encourage the people who're out there just now quite happily using some of that software to say, "Hey, I'll buy a new bit of software with the *GEOS* capabilities." So to them, it increases their overall market.

Gazette: Has Commodore stopped making the old 64 in favor of the new 64C with the case?

Shepherd: Yes, we have.

Gazette: When were the last old-style 64s produced?

Shepherd: Well, we really cut them out. There was a small overlap of no more than a week in the middle of June. So, up to about halfway through June we shipped out the last of the old 64s, and then cut over, virtually overnight, straight onto the new machines.

Gazette: So, there was really no downtime as far as production of 64s is concerned?

Shepherd: No, none at all.

Gazette: What's the status of 64C availability now and into the Christmas season? Will you have enough units to meet expected demand?

Shepherd: We don't see any problem at all. We did, but it was purely a temporary thing. And I guess, because of the pent-up demand, we totally sold out of the Commodore 64Cs by the end of June.

And we'll be steadily producing that machine right through Christmas, so we don't see a real problem with supply and demand.

Gazette: Commodore obviously views the QuantumLink telecommunications system as an important aspect of the marketing of the 64C. Do you have any idea how many subscribers there are, or how many you would ultimately like to target with this kind of system?

> **"...software publishers who are primarily interested in productivity software have been very encouraged by *GEOS* because ...they see it as a second market."**

Shepherd: In terms of QuantumLink?

Gazette: Yes.

Shepherd: Well, I guess it's their target more than ours, although we obviously benefit. The degree is an intangible. I guess we'll never know totally the answer to "Did you buy the Commodore 64 for itself; did you buy it for *GEOS*; did you buy it for QuantumLink?" We'll never have a tangible answer to that, but I think there's the possibility that we'll be looking at somewhere close to one in every ten purchasers of the 64 actually going ahead and utilizing QuantumLink.

Gazette: Does Commodore plan any arrangements similar to this with other telecommunications services?

Shepherd: This is the primary one in terms of what QuantumLink offers us. It doesn't preclude us, however, from getting involved with other telecommunications services. So, there's an opportunity that's a mutual one in terms of assisting them in terms of subscribers, but us believing that it also helps us to sell more Commodore machines. We'll certainly look at that case by case.

Gazette: How do sales of the 64C have an impact on sales of the 128? Is there an overlap or are they in different market niches?

Shepherd: There's a definite overlap, which is sort of for and against. There's a number of 64 owners—we estimate possibly 20 to 25 percent of 128 purchasers are people upgrading from a 64. The reason for that, I guess, is that these people have invested quite a bit of money in disk drive, printer, monitor, software, and when they transfer across to the 128, they can keep the disk drive, keep the printer, and keep the software. And slowly but surely they're buying a 128 disk drive or buying a printer. But their up-front expense, really, is only for the CPU.

So, on the one hand we expect that to continue—that there will still be people upgrading from the 64 to the 128. On the other hand, because we're essentially making the 64 a more powerful machine, then that presents some opposition to the 128. If you say, "Why would somebody buy a 128 rather than a 64?" I guess there are several reasons. The CP/M mode—and that's stronger than most people think. If you look at a lot of what's going on on the telecommunications services on the 128, you'll find a healthy report going back and forth on CP/M. But, I guess the major reason is to get access to 80-column and productivity software. That's offset to a degree, of course, by what we're doing with *GEOS*.

There's a definite overlap. I guess it gets down to how much money someone is prepared to invest.

Gazette: At the Summer Consumer Electronics Show (CES), Commodore announced that more than

IF YOU CAN FIND A BETTER C128 PROGRAM WE'LL BUY IT FOR YOU!*

Now includes THESAURUS and OUTLINER
TIMEWORKS
128
Word Writer with Spell Checker 128
Data Manager with Report Writer 128
SwiftCalc with Super Graphics 128
SIDEWAYS

WORD WRITER 128 with Thesaurus

An 80-column professional word processing system for home and business use. You get:

- **An 85,000 word Spelling Checker**
- **An Integrated Thesaurus** with thousands of synonyms
- **A built-in, 5-function calculator**
- **On-Screen Highlighting** displays underlining, **boldface** and *italics* – as you write.
- **Headers and Footers**
- **Superscript and Subscript**
- **An Outliner Processor** that quickly organizes facts and ideas in outline format.

With Timeworks you get more than software. . .

You Get Our Customer Technical Support Team – free to all registered users.

DATA MANAGER 128 with Report Writer

A complete general information storage and retrieval system with report writing, graphics, statistics, and label making capabilities. Plus, you get:

- **Exclusive X-SEARCH, X-SORT, and X-CHART** features that allow you to cross-search any category of information; sort items alphabetically, numerically, or by date; break down statistical information into categories; and graphically view your results.
- **On-Screen Tutorial**
- **Large Capacity Record Size** includes up to 64 pages (4,096 characters) per record.

With Timeworks you get our Money Back Guarantee*
If you can find anything that works better for you – and it's available – we'll buy it for you. Details inside every Timeworks package.**

SWIFTCALC 128 with Super Graphics

A powerful, easy-to-use electronic spreadsheet designed especially for home and business use:

- **Super Graphics** display and print out numerical information using pie charts, vertical bar charts, scatter diagrams, line graphs, 3-D staggered bar charts.
- **Sideways** prints all your columns on one continuous sheet . . . sideways.
- **250 rows and up to 250 columns** provide up to 62,500 cells (locations) in which to place information
- **Performs mathematical and financial analysis functions.**
- **Speed Keys** • **SwiftSort** • **SwiftDOS**

You Get Our Liberal Upgrade and Exchange Policy – Details are inside every Timeworks package.

TIMEWORKS®

More power for your dollar

Timeworks, Inc., 444 Lake Cook Road, Deerfield, Illinois 60015
312-948-9200

For the Commodore 128™ Computer***

Suggested Retail List Price:
$69.95 each

Available now at your favorite dealer, or call Timeworks

**TO ORDER CALL:
1-800-535-9497**

These Programs INTERFACE with Each Other

** Offer expires 90 days after date of original purchase.
*** COMMODORE 128 is a trademark of Commodore Electronics, Ltd.

600,000 128s have been sold. Any estimates as far as the number of 128s out there by the end of the year?

Shepherd: That was a worldwide figure, that 600,000. That would have been up to, roughly, the end of April or maybe closer to the end of May. And that did include our launch in the selling period. Up to the end of the year, I think we'd have to be fairly close to the million mark worldwide.

Gazette: How many just in North America?

Shepherd: Roughly, it's about 50/50. We split right down the middle—50 percent North America and 50 percent the rest of the world. So, you'd be talking about 500,000 in North America.

Gazette: What's the biggest 128 market outside the U.S. and Canada?

Shepherd: Most definitely, that would be Germany; the European market, in general, but specifically Germany.

Gazette: Commodore has received a great deal of press during the past six months relating to the company's financial situation. You've done some belt-tightening. How do things look for the rest of 1986 and into 1987 for Commodore?

Shepherd: One of the reasons for the cynicism out there is that Commodore has over the last six months had a couple of false starts in terms of "We're climbing out of it; you're going to see a change," and then reversing our position. We did, at the end of the March quarter, indicate that we thought that expenses were under control to the extent that in the June quarter we'd be near a break-even situation. And that we'd return to profitability in the September quarter. All the June results are not in yet, and probably won't be for a couple of weeks. I'd say at this stage, we expect to be very close to a break-even at the June quarter. Whether that is plus or minus one percent or two percent, we don't know yet.

We think we'll break even at the June quarter, and that we will, in fact, be profitable for the second half of this year.

Gazette: Is that a function of 64 and 128 sales more than Amiga sales at this point?

Shepherd: It's really a combination. As I indicated, with the 64 we increased prices. We needed the extra margin. So, it's a combination of 64 and 128 sales; Amiga sales, while certainly not mind-boggling—we wouldn't try and pretend they were—they're steadier now. It's a combination of that, and cost reducing. There's a whole lot of factors that go into that total pie.

"We have our expenses under control to the extent that we can run the business profitably from here on in."

Gazette: So, as a company that has pared down the total number of employees and undertaken other belt-tightening measures, is the cost effectiveness better for Commodore during the next six months to a year?

Shepherd: Most definitely. I mean, we have gone through considerable pain to get it there. We're confident that there are no more sudden surprises. We have our expenses under control to the extent that we can run the business profitably from here on in.

Gazette: What's your estimate on unit sales of the Amiga now, and by the end of the year?

Shepherd: First, and I guess I'm almost in a defensive mood here, the reason for that is that the Amiga has been sold only in North America up to the middle of June. It was only launched in Europe—really they only got the first shipment about halfway through June. One of the reasons, when I say I'm defensive on that, is that everytime somebody looks at Amiga figures, they never qualify it by saying it's only in North America.

So now, on Amigas, we're looking roughly at a run rate of 10,000 to 15,000 a month on a worldwide basis.

Gazette: Any estimates of what the total will be by the end of the year? Or is that too difficult at this point to say?

Shepherd: It's too difficult. One of the things is that, we're committed to going out publicly and saying that. In fact, we've got some people in Europe just now and we feel that, whether that's good news, mediocre news, or bad news, that we have to come out with a statement that indicates the installed base of the Amiga computers and the run-rate we expect for the rest of the year. There's no sense in us being bullish and loving to quote our 64 and 128 figures, and hiding from Amiga figures. So, in the next couple of weeks, we should have all these figures together, and we'd be quite happy to give them out.

Gazette: There have been some press reports that the Amiga would be maintained in a vertical market rather than in a more broadly based market. Could you comment on those reports, and do you see the Amiga as a consumer machine?

Shepherd: I think to a degree there was a bit of misquoting [in those reports]. I certainly said in a couple of interviews that apart from some software development that's taking place, we'd have additional markets which would be in the vertical market areas. Not at the expense of the broad market.

We see ourselves in vertical markets—I don't know if you'd call desktop publishing a vertical market; that's probably a horizontal market. We see ourselves entering the desktop publishing market. We see ourselves in the graphics market, in the illustrators' market, using some things like the Genlocks and the Frame Grabbers, in addition to the broad base of Amiga buyers.

So, I think what you're going to see is the Amiga continue to be acceptable to the hobbyist, who's after the technology more than anything else, who's, I guess, in the higher income bracket for expensive home machines. And I can see them getting into the corporate market, not so much as an alternative to, let's say, corporate America's PC market; but in the corporate markets where the Amiga has capabilities that the PC doesn't have—that use the Amiga's graphics and color and sound capabilities.

So, yes, I've noticed that myself—(the reports) that Commodore's thrust was more into vertical markets. I think what we're saying is that there's an *additional* thrust going into vertical markets.

Gazette: Many people would like to see Commodore lower the price and make the Amiga competitive with the Atari ST and the PC clones by the end of this year. Or, on the other hand, have a different version of the Amiga at more of a mass-market level. Is either of those a possibility?

Shepherd: Well, I'd say that both are. When you come up with a base product, like the Amiga 1000, in terms of future development, we do two things: We look at cost reduction and we look at product enhancement. It's very difficult to do both. Sometimes it's an either/or: How can you enhance the product, but get some cost out of it.

So, we're certainly looking at just how we can cost-reduce the Amiga. And we're also looking at a more powerful Amiga. Neither of these would you see this year. And we haven't pushed the button on one or both of them. Certainly, our target in 1987 is to have an enhanced Amiga, and if possible—and it may be a separate machine—also to have a lower-cost Amiga.

Gazette: One last question. Some third-party software developers have told us that the 64C, and the large installed base of the 64, and sales of the 128 mean that their commitment to that market could be for at least another three years—and that it's open-ended after that. What's Commodore's view on commitment to the 64 and 128?

Shepherd: Essentially, they both come from the same operating system. And the 128 certainly has enhancements on that.

We believe that the 64 and 128 product line certainly goes into 1988. I think we'd be naïve to say that we'd be selling them in 1990. If we'd be selling them four years from now, they would need continued enhancements. It's quite possible. Two years ago, we never thought and neither did anybody else, that something like *GEOS* would come along. There are things that we're now capable of doing technology-wise that two years ago we didn't think we were capable of doing on the 64.

So, that basic 64 product line—and I include the 128 in that—we think lends itself to additional development. And we'd certainly see it through 1988. I'm not going to quote our friend from Apple who said that suddenly the Apple II is the machine of the 1990s [laughs].

But, yes, you're looking at a solid two to three years out of that [64 and 128] machine.

The Command Center consolidates all these peripherals.
Built-in features include power strip, fan and phone plug.

The NEW Commodore Educational Software

Selby Bateman, Features Editor

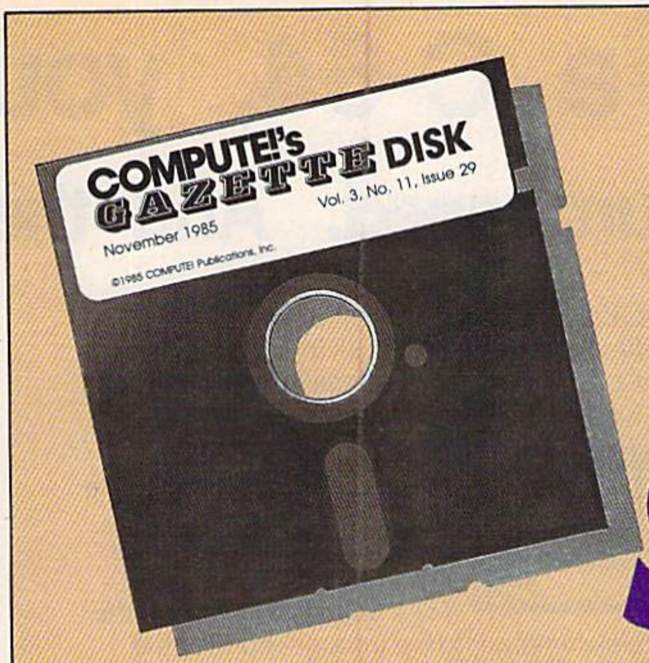
One of the toughest challenges faced by any software company is developing educational computer programs that can be simultaneously effective, engaging, and useful in the home and classroom. In addition, there have been too many companies chasing too few buyers, which has resulted in an industry shakeout among educational software producers. And yet, some companies have survived and are doing well by carefully targeting their customers and understanding what kinds of programs are needed in homes and schools. For Commodore owners, the result is a growing supply of good educational programs.

When purchasers of the new Commodore 64C computer open the package, one of the first things they'll find is an educational software program named *Odell Lake*, created by MECC, the Minnesota Educational Computing Corporation.

MECC is no newcomer to educational software, although home computer users may be less familiar with the company than those in education. The company was started in 1973, which is almost ancient history in microcomputer circles. And since that time, MECC has established a solid reputation in schools across the country.

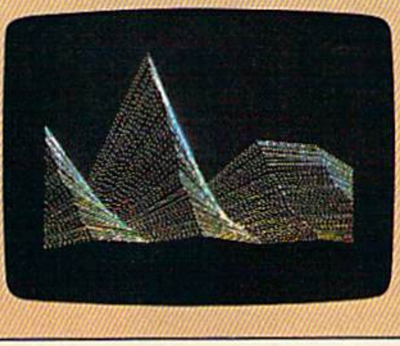
But this does mark the first time that Commodore has chosen to bundle an educational program with one of its computers. MECC's involvement and Commodore's effort both say a lot about many people's perception of the 64 as an inexpensive and versatile learning machine for both classroom and home environments.

Now, with *Odell Lake* and selected



BUILT FOR SPEED

Now Get Inside Your Commodore with *COMPUTE!'s Gazette Disk.*



Now there's a way to get all the exciting, fun-filled programs of ***COMPUTE!'s Gazette***—already on disk—with ***COMPUTE!'s Gazette Disk***.

Subscribe today, and month after month you'll get a new, fully-tested 5¼-inch floppy disk guaranteed to run on your Commodore 64, Commodore 128, or VIC-20 personal computer.

COMPUTE!'s Gazette Disk brings you all the latest, most challenging, most fascinating programs published in the corresponding issue of ***COMPUTE!'s Gazette***. So instead of spending hours typing in each program, now all you do is insert the disk...and your programs load in seconds.

RESULT: You have hours more time to enjoy all those great programs which appear in ***COMPUTE!'s Gazette***—programs like Disk Editor, Face-Off, Turbo Copy, Arcade Baseball, Sound Designer, SpeedScript, and hundreds of others.

So don't waste another moment. Subscribe today to ***COMPUTE!'s Gazette Disk*** at this money-saving price. Take a full year's subscription for just $69.95. You save 55 percent off the single issue price. Subscribe for two years and save even more! Return the enclosed card or call toll-free 1-800-247-5470 now (in Iowa call 1-800-532-1272).

Individual issues of the *Disk* are available for $12.95 (plus $2.00 shipping and handling) by calling 1-800-346-6767 (in NY 1-212-887-8525).

If you own a C-64, you

The one you purchased. And the

GEOS realizes the technical potential that has been in the C-64 all along. Speed. Power. Ease of use. Sophistication. Elegant, practical applications you might expect of a high-end personal computer, all made possible with GEOS. It's so simple—but then, so was fire. Once it caught on.

To begin at the beginning. **GEOS** stands for **GRAPHIC ENVIRONMENT OPERATING SYSTEM.** Why?

GRAPHIC: Because menus and icons replace long, typed command lines. Point and click, that's it.

ENVIRONMENT: Because GEOS provides a consistent, powerful way to use your computer. Learning new applications is a snap (or should we say click).

OPERATING SYSTEM: Because GEOS orchestrates every function so that they all work together systematically, even symphonically.

Some basics. Icons are graphic images which represent files or utilities. Each is different, and all are easy to recognize and easy to use.

A menu is just that: a list of functions available for selection. When a menu appears, move the pointer to any item you wish. Click. Click. You're on your way.

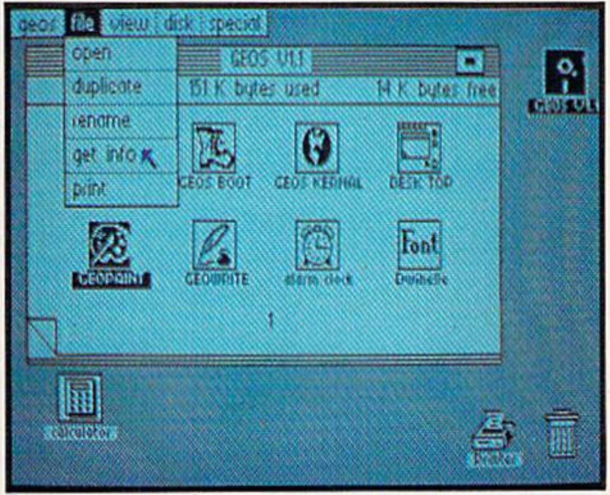
A pointer is used to select and activate items. To move the pointer, roll the mouse or trackball or rotate the joystick. Once on target, click once to select; click a second time to activate.

Fonts are a new way of looking at text. Choose from 5 different fonts (with more on the way). Try Dwinelle, or Roma, **bold,** or *italics,* even underline and outline. Need to fit more words on a line? Pick a smaller point size, like University 6 point, and get over one hundred characters per line.

All this and fast too. Because the integrated diskTurbo software improves 1541 disk drive performance 5 to 7 times. That's right. On both reads and writes.

GEOS can be divided into 4 areas: two functional aspects (deskTop and Desk Accessories), and two major applications (geoPaint and geoWrite).

deskTop. deskTop is a graphic interface, making file organization and management easy. As always, you call the shots. Load a disk. Files appear as icons on the disk notepad; to flip through, point at the folded corner and click. Prefer a file appear on a different sheet? Move it. It's easy.

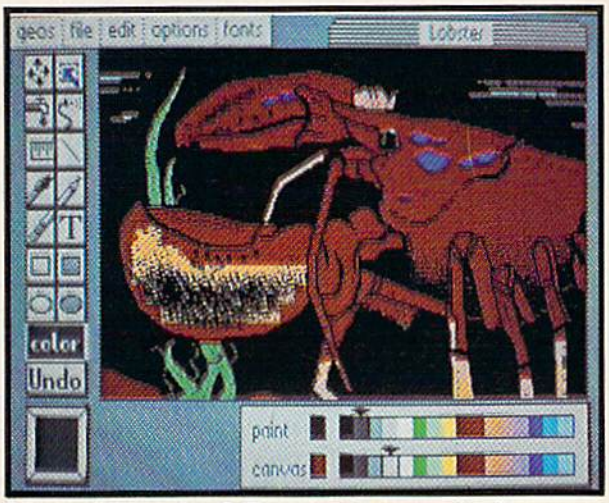


Create a new document or re-name an existing one. Want to copy a file onto the same or a different disk? Fine. Forgotten what a file contains? Select "get info" from the file menu. A description of that file's contents appears. Finished with a file? Print it. Save it. Or drop it in the trash and have done with it. Your call.

geoPaint. A full-featured, color graphics workshop at your fingertips. The pointer operates any one of the fourteen graphic tools and shapes in the drawing menu. Create masterpieces on the Drawing Window. By turns, use a pencil, an airbrush or a paint brush, each with a character all its own. Draw straight lines, squares, rectangles or circles. Fill in with any of the 32 patterns. Switch to pixel-mode, where each dot in a selected section is magnified many times its size for easy manipulation.

own two Machines.

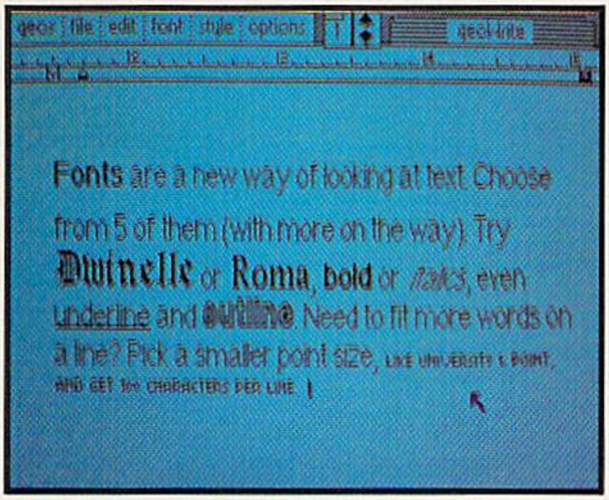
personal computer GEOS™ unlocks.



Second thoughts? Erase what you don't want. Or "UNDO" your last act. (If only life could imitate art!)

Add text if you like, in different fonts, styles or point sizes. Even change its position or layout at will.

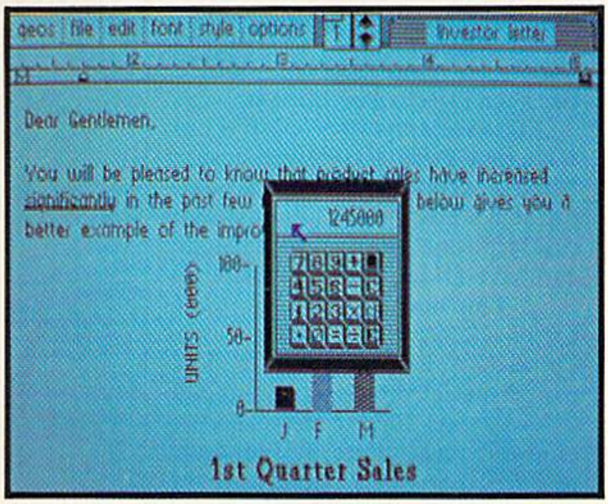
Move or copy any part of your creation. Once done, you can include your artwork in another document—a letter home perhaps. (Won't Mother be pleased?) GEOS makes it easy.



Fonts are a new way of looking at text. Choose from 5 of them (with more on the way). Try Dwinelle or Roma, bold or italics, even underline and outline. Need to fit more words on a line? Pick a smaller point size, LIKE UNIVERSITY 6 POINT, AND GET 100 CHARACTERS PER LINE.

geoWrite. An easy to use, "what you see is what you get" word processor. Create documents. Insert, copy, move or delete text as you wish. Choose from 5 different fonts in many different styles and point sizes. Preview your page exactly as it will appear off the printer. Typists will appreciate tabs, word-wrap and page breaks.

Documents may contain up to 64 pages. What's more, you can move to any page instantly. If you like, you can cut selected text from one section and move or copy it to another. Add graphics from geoPaint. It's a cinch.



Dear Gentlemen,

You will be pleased to know that product sales have increased significantly in the past few ... below gives you a better example of the impro...

1st Quarter Sales

Desk Accessories. Handy programs you can use while in any GEOS application. These include an alarm clock, a notepad for reminders, a four-function calculator, and photo and text albums which store pictures and phrases you may then paste into applications. The Preference Manager even lets you establish parameters for everything from mouse speed to the date and time—even background color. Civilized options, every one.

BERKELEY
Softworks

GEOS, A Whole New World for the C-64.

$59.95 TO ORDER: 800-443-0100 x234

GEOS Diskette includes deskTop, diskTurbo, geoPaint, geoWrite and desk Accessories $59.95.
CA residents add 6.5% tax. ($4.50 US/7.50 Foreign shipping and handling.)
US Funds only. B.S.W Order Processing, PO Box 57135, Hayward, CA 94545

Commodore 64 and C-64 are trademarks of Commodore Electronics, Ltd. GEOS, deskTop, geoPaint, geoWrite, diskTurbo and Berkeley Softworks are trademarks of Berkeley Softworks.

titles from MECC's software line, home users will increasingly be exposed to the company's products. This movement from the school market to the home user is only one example of the ways in which educational software companies are attempting to bridge the gap between the two markets—home and school.

A reverse trend is occurring with some other educational software companies such as CBS Interactive Learning. Previously known as CBS Software, the new Interactive Learning group has recently been concentrating more on sales to the education market than to home users. And the result, say representatives for both CBS and MECC, has been effective in different ways.

Both attempts rest on the premise that there *is* common ground between school and home for educational software, and that the right products and the right packaging can have an impact in both arenas.

"The home market complements the education market," says Dr. Richard Pollock, MECC's director of special projects. Pollock also believes that having more knowledgeable purchasers in the home and school markets is today providing a more discriminating consumer base for good educational software.

And, as he says, "The software itself is becoming more sophisticated and closer to the way teachers teach. Before now, teachers had to adapt to the way a program worked."

For MECC, the approach to the home computer market has been a slow, but steady progression. For the past three years, MECC has had Commodore products available—some 30 in all now. But only recently has the company worked directly with Commodore to provide a series of educational programs for the home market. The first of these, *Odell Lake*, is free with the 64C. Eight more titles, all in the young-learning area, are now available through Commodore. MECC also provides a free catalog of its entire product line to anyone who would like more information.

> ***"The software itself is becoming more sophisticated and closer to the way teachers teach."***

While CBS has had a very large presence in the home computer market over the past several years, one of the most successful areas of their business has turned out to be the school market, says Marylyn Rosenblum, director of sales and marketing for CBS Interactive Learning.

"Like lots of other people, we realized at some point that the consumer market wasn't going to live up to the expectations or the investment we had made. But at the same time, we had, almost without knowing it, a lovely small, profitable business selling some of our software to schools.

"Because CBS is very heavily involved in the educational market, it looked to people here like it would make sense to take what we had—a very valuable body of products—and focus on the area where we could realize a significant return. That's what really started us turning toward the education market."

The marketing considerations confronted by both of these companies are being repeated across numerous other software producers. And the net result has been that almost all of the successful software companies producing educational programs have been carefully, and usually quite cautiously, exploring both sides of the street—home and school.

Literally hundreds of educational software programs are available for the Commodore 64, developed during a period of over four years. While a compilation of all this software would be impossible to fit into one article, there is a variety of new programs that can help teachers, parents, and serious students get a flavor of the mixture of approaches being taken. The products and companies included below provide an overview of some of the newer educational software packages you'll find available for your Commodore computer.

American Educational Computer: AEC continues to offer an extensive line of curriculum-based software for many computers. Some of the newer Commodore programs available offer education in the areas of spelling, phonics, reading, vocabulary, grammar, Spanish, French, U.S. and world geography and history, and U.S. government. (American Educational Computer, 801 N.W. 63rd St., Oklahoma City, OK 73116)

Arrakis Technologies: A new marketing arrangement for Arrakis software is making several of the company's curriculum-based packages available in Commodore versions. Already well known in the schools, Arrakis has more than a dozen curriculum-oriented education programs for Commodore computers. The products, all aimed at grades 7–12 in a variety of subject areas, have recently begun to be distributed in the United States by the Webster Division of McGraw-Hill, and in Canada by Grolier Electronic Publishing. The school-oriented packages include algebra, trigonometry, statistics, physics, chemistry, biology, and other curriculum-based topics. (Computer Marketing, Webster Division, 28th

Floor, McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020, or Grolier Electronic Publishing, 95 Madison Ave., New York, NY 10016)

BCI Software: During the past year, low-cost educational software from BCI has become available, including a variety of Commodore education programs in the areas of mathematics, English–Spanish vocabulary, science, history, and geography, covering grades one through ten. (BCI Software, P.O. Box 730, Ringwood, NJ 07456)

Brøderbund: Brøderbund's bestselling *Print Shop* has recently spawned both the *Print Shop Companion* and *Print Shop Graphic Libraries* for the Commodore 64. These programs are popular in the school and home as hands-on introductions to creating cards, posters, banners, and similar printed material. Brøderbund also recently announced a Commodore 64 version of the popular *Where in the World Is Carmen Sandiego?* that was first developed for the Apple computer. (Brøderbund, 17 Paul Dr., San Rafael, CA 94903-2101)

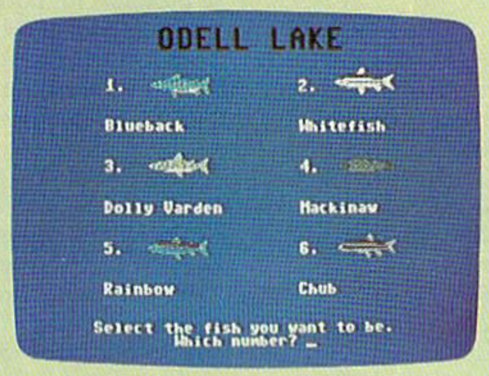
CBS Interactive Learning: Within the past year, CBS has changed its focus quite a bit, moving more into the schools and less aggressively into the home market. Among the newest Commodore entries is a series of reading motivation programs based on famous books that are frequently taught in the schools. Called The Novel Approach Series, these programs offer supplementary support before, during, and after reading the volumes. By this fall, CBS will have Commodore versions of *Animal Farm, Lord of the Flies, A Tale of Two Cities, The Call of the Wild,* and *Romeo and Juliet.* The CBS Success with Math and Success with Algebra series continue to be popular in both the home and the classroom. (CBS Interactive Learning, CBS, 1 Fawcett Pl., Greenwich, CT 06836)

Davidson & Associates: Bestsellers such as *Math Blaster!* and *Word Attack!,* first available on the Apple computers, have been released in Commodore format by Davidson, which has a strong presence in the Apple community. There are a variety of other Commodore programs, as well. (Davidson & Associates, 3135 Kashiwa St., Torrance, CA 90505)

DesignWare and **EduWare:** Among the newest Commodore educational packages from DesignWare are *European Nations & Locations,* a European facts and geography program; *The Grammar Examiner,* covering basic grammar skills; *Remember!,* a memory learning tool for high school students and adults; and *The Notable Phantom,* a music learning program for youngsters five to ten years old. EduWare also offers several Commodore educational programs. (DesignWare and EduWare Services, 185 Berry St., Bldg. 3, Suite 158, San Francisco, CA 94107)

DLM Teaching Resources: Another well-known educational software company with a strong presence in the schools is DLM Teaching Resources. The company's newest Commodore software includes the new *Create with Garfield,* in which students create their own posters, labels, and cartoons as they practice writing, creative thinking, design, and organization skills. The company also sells approximately two dozen other Commodore educational programs in math, language arts, and early childhood learning. (DLM, One DLM Park, P.O. Box 4000, Allen, TX 75002)

Gamco Industries: Gamco publishes educational software for Apple, Radio Shack, and Commodore computers. Its programs offer practice in learning many of the major subject areas taught in schools. Two of its latest Commodore products are *Money Squares,* a tic-tac-toe game that drills players in money skills; and *Time Explorers,* a two-player adventure game that teaches how to tell time. (Gamco Industries, P.O. Box 1911, Big Spring, TX 79720)

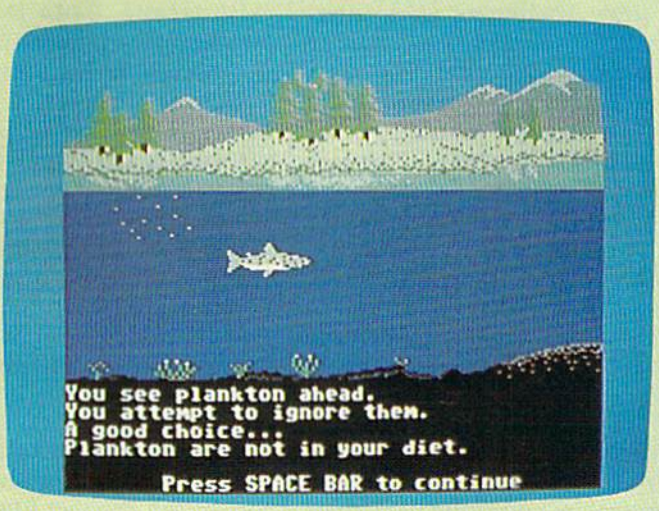


```
ODELL LAKE

1.              2.
Blueback        Whitefish
3.              4.
Dolly Varden    Mackinaw
5.              6.
Rainbow         Chub

Select the fish you want to be.
       Which number? _
```

In Odell Lake, *you assume the role of one of six kinds of fish (above), and then learn how to survive in the lake environment.*



```
You see plankton ahead.
You attempt to ignore them.
A good choice...
Plankton are not in your diet.
            Press SPACE BAR to continue
```

The Learning Company: The Learning Company built a reputation in the educational market with its acclaimed *Rocky's Boots*. Titles that have more recently become available in Commodore versions include *Reader Rabbit, Robot Odyssey I,* and *Magic Spells*. There are also a number of older titles in Commodore versions. (The Learning Company, 545 Middlefield Rd., Suite 170, Menlo Park, CA 94025)

Learning Technologies: Learning Technologies' educational strategy is to build what they call a Lifetime Learning Library. The company's programs are presented in three series: Early Learning, Thinking Strategies, and Math Concepts. Among recent titles are *Animal Hotel, Lion's Workshop, Number Please, Gremlin Hunt,* and *Math in a Nutshell*. Each package contains a Learning Kit, which includes a full-color poster illustrating the program's graphics, a custom lesson plan, three worksheets, user management charts, and award certificates. (Learning Technologies, 4255 LBJ, Suite 265, Dallas, TX 75244)

MECC: *Odell Lake* is a new environmental simulation that teaches children from five to eight years of age how a lake habitat functions. Other MECC titles, newly available from Commodore, include *Adventures with Fractions; Expeditions* (American history); *The Friendly Computer* (elementary introduction to the computer); *The Glass Computer* (how computers work internally); *The Market Place* (elementary economics); *Path Tactics* (racing robots to learn math skills); *Pre-Reading* (reading readiness); and *Spelling Bee* (elementary spelling). These MECC titles are available directly from Commodore. (Commodore Business Machines, 1200 Wilson Dr., West Chester, PA 19380)

Numerous other Commodore titles are available directly from MECC. (MECC, 3490 Lexington Ave. North, St. Paul, MN 55126)

Micro-Ed: Micro-Ed has been a major supporter of Commodore computers in educational settings for quite a while. Currently, about 1000 Micro-Ed programs are available for the Commodore 64, a line of curriculum-based software that is primarily sold into the schools, but is now also available to the retail market. Some of these titles include *Capitalization Series* and *Basic Grammar* (language arts series); *U.S. Time Zones, The Calendar,* and *The Clock* (social studies series); *Story Problems in Addition and Subtraction* and *Math Spin* (math series); and *The Atom* and *Save the Whales* (science series). (Micro-Ed, P.O. Box 444005, Eden Prairie, MN 55344)

PROBLEM 1

PRESS ± TO QUIT, H FOR HELP

The Commodore version of CBS's popular Success With Math series has been recently revised.

Mindscape: Some of Mindscape's earliest products were educational programs, though the company has since branched out to offer entertainment packages as well. The most recent Commodore educational titles include *The Halley Project, Crossword Magic, The Perfect Score, Bank Street Music Writer, Bank Street Story Book, Tink's Subtraction Fair,* and *Castle Clobber*. (Mindscape, 3444 Dundee Rd., Northbrook, IL 60062)

Simon & Schuster: Simon & Schuster's newest Commodore educational title is *Chem Lab*, a software-based chemistry laboratory for children interested in science. The company also has Commodore versions of many other titles, including such recent programs as *Kermit's Electronic Storymaker* and *The Great Gonzo in Wordrider*. (Simon & Schuster Software, Gulf and Western Bldg., One Gulf and Western Plaza, New York, NY 10023)

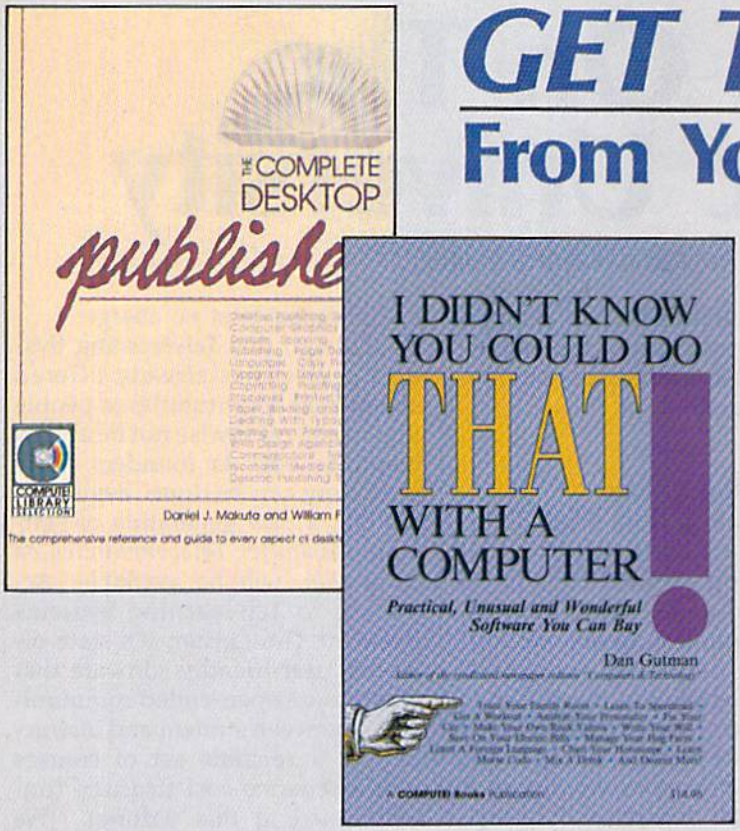
Spinnaker Software: Spinnaker, which earlier this year absorbed the product line of Hayden Software, recently released Commodore versions of two Homework Helpers Series programs, *Writing* and *Word Problems*, both for students in grades 7–12. The company also offers its Early Learning Series, Learning Discovery Series, Learning Adventure Series, and Fisher-Price Series. (Spinnaker Software, One Kendall Square, Cambridge, MA 02139)

Springboard Software: On the heels of Springboard's popular *The Newsroom* newspaper/newsletter printing program, the company has recently released Commodore versions of *Clip Art Collection 1* and 2 for use with *The Newsroom*. The program is used in both schools and homes, and is being further enhanced with Springboard's Commodore version of the *Graphics Expander, 1*, which adds more tools to *The Newsroom*. Two of the company's most popular products have been *Early Games for Young Children* and *Easy As ABC*. (Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435)

Sunburst: Sunburst has more than 30 Commodore educational programs, ranging from preschool to adult-level packages. Most recently, the popular Muppet Learning Keys tablet, first available on Apple computers, has become available in a Commodore version with supporting software. Two of the new Learning Keys programs are *Getting Ready to Read and Add* and *Tiger's Tales*, both young-learning packages. There are many other Commodore products, and the company publishes six catalogs a year featuring products for both the school and home markets. (Microcomputer Courseware Division, Sunburst Communications, Inc., 39 Washington Ave., Pleasantville, NY 10570)

Weekly Reader Family Software: Weekly Reader continues to introduce new titles in its successful *Stickybear* series of software products for younger children. Already available for Commodore are *Stickybear Town Builder, Stickybear Spellgrabber, Stickybear Typing, Stickybear Opposites, Stickybear ABC, Stickybear Shapes, Stickybear Reading, Stickybear Math I,* and *Stickybear Numbers*. (Weekly Reader Family Software, 4343 Equity Dr., P.O. Box 16754, Columbus, OH 43216)

GET THE MOST
From Your Computer!


THE COMPLETE DESKTOP publisher
COMPUTE! LIBRARY SELECTION
Daniel J. Makuta and William F
The comprehensive reference and guide to every aspect of deskt
I DIDN'T KNOW YOU COULD DO THAT! WITH A COMPUTER
Practical, Unusual and Wonderful Software You Can Buy
Dan Gutman
A COMPUTE! Books Publication
$14.95

Two new books from COMPUTE! bring you innovative ideas to exploit the capabilities of your computer to the fullest and to make use of it in ways you didn't know were possible. There's an in-depth look at desktop publishing—how to get started and how to set up your publication. A compendium of special software tells you where to purchase inexpensive software that will open up a new world of applications. The information is non–machine-specific and is designed for beginning to advanced computer users.

The Complete Desktop Publisher

Daniel J. Makuta and William F. Lawrence

A comprehensive reference and guide to what is fast becoming one of the most popular uses of computers, this book guides the novice through the intricacies of desktop publishing. Clearly written and understandable, this guide covers all the basics of typography, layout, and design. Sample formats and layout grids make it easy to get started. Details on using graphics to enhance publications, discussions of the desktop publishing software currently available, and comparisons of laser printers, typesetters, and other printing devices give you everything you need to make intelligent decisions. Chapters outline the elements of fitting copy, proofreading, binding and folding, and media conversion. You'll also find a wealth of information on telecommunications—how to transmit your publication for typesetting or printing—as well as tips on dealing with outside design agencies, typographers, and printers. The definitive guide.

$19.95 ISBN 0-87455-065-3

I Didn't Know You Could Do *That* with a Computer!

Dan Gutman

There's more to computers than processing words, juggling numbers in a spreadsheet, or filing records in a database. A world of unusual, practical, and amazing computer programs is available which can help you do anything from planting a garden to writing a will. These programs range from the simply fascinating to the outright esoteric. More than 100 little-known, yet intriguing commercial software packages are evaluated and reviewed: programs which can plot your astrological future, show you the night skies, plan your next road trip, help you raise your child, and give you the edge at the racetrack. Entertaining—certainly unique—this book puts an end once and for all to the question "Now that I have a computer, what do I do with it?"

$14.95 ISBN 0-87455-066-1

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

Or, to order directly from COMPUTE!, call toll-free 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 $2.00 shipping per book for U.S. and surface mail; $5.00 airmail. North Carolina residents add 4.5 percent sales tax. New York residents add 8.25 percent sales tax.

Please allow 4–6 weeks for delivery.

COMPUTE! Publications, Inc. abc

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

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

Update On The Electronic University

Kathy Yakal, Assistant Features Editor

The process of completing a college education can be complicated by many factors. First, finances. Tuition costs have skyrocketed over the last few years. Add to that basic cost the price of books and housing and all of the other necessities that seem to crop up in the course of four years, and you're talking about a fairly expensive proposition. Second, location. If you don't happen to live near a college or university, and are unwilling or unable to relocate, your options are limited. And third, timing. Some people simply cannot fit a college education into their schedules. They start families or take jobs after high school, and can never fit in a college education at the right time.

Correspondence courses are one solution, as are night classes at community colleges. Another option, made possible by the relatively simple, inexpensive process of telecommunications, is TeleLearning System's Electronic University, a unique personal computer delivery system for education. Since its inception in 1983, people of all ages and backgrounds who would otherwise have been unable to attend traditional classes have successfully completed courses—even earned degrees—using their computers to communicate with instructors across the country. The program has received acclaim from business, education, and government leaders, all the way up to the Vice President of the United States.

The Electronic University does not grant degrees. It serves as a delivery system, a liaison between students wanting to take classes and the institutions willing to administer them.

Software is available for Commodore 64, Apple, and IBM computers (Telelearning Systems sells these computers and modems at a discount, if you don't already own one). The registration fee for the network itself is $195. This, though, is just a starter kit, and does not include the cost of course registration or additional fees.

Several Alternatives

You may choose from three different paths in the Electronic University, according to your individual educational goals. First, you can take individual courses without pursuing a degree or certificate. Should you decide in the future to work toward a degree, these earned credits can be applied. Second, you can earn a degree or certificate. A variety of full degrees and specialized certificate programs is available through the Electronic University Network, including an Associate of Arts, Associate of Science in Management, Bachelor of Arts, Bachelor of Science in Business Administration, and Master of Business Administration (MBA). Six specialized certificates in business and computer-related fields are also available. Third, you can accumulate transferable credit. Many students already working toward a degree at a college in their local area take supplementary courses on the network. These course credits are then transferred to the local degree program.

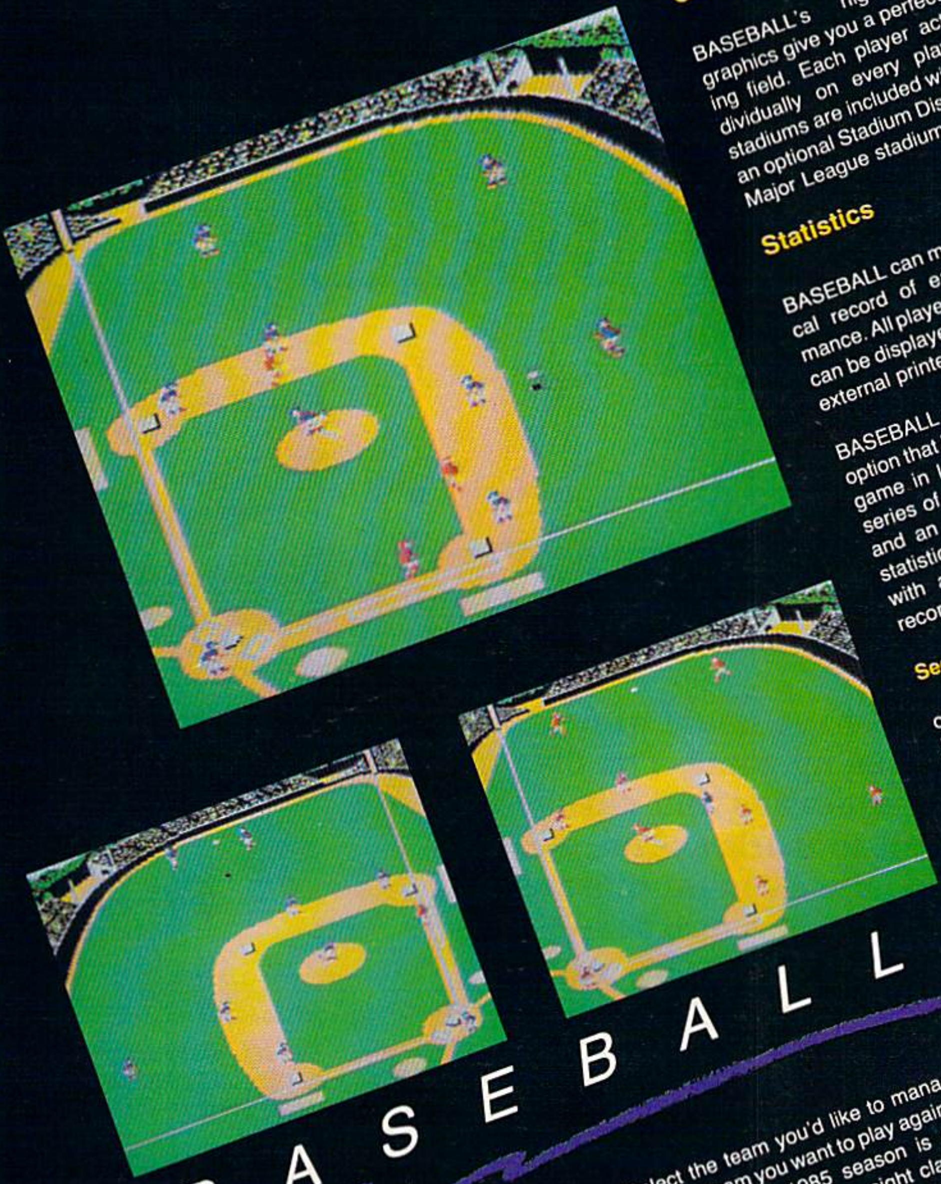
Once you've registered for a course and received the class materials in the mail, you and your instructor communicate via electronic mail on the network. You send completed assignments and questions to your instructor, and receive support and feedback. In addition, there are online forums—classroom situations where you simultaneously discuss topics with other students online.

An Electronic University Network membership gives you access to more than 60 databases and other services through the Electronic Library. Individualized counseling is also available at no charge.

Though this Telelearning Systems project has already offered educational opportunities to people who would otherwise not be able to attend school, its founders think even more can be done. Beginning this fall, a new generation of software, *Computer Telecommunicated Instruction*, will be available. According to Telelearning Systems president Tim Leister, it's state-of-the-art, user-friendly software that allows more open-ended communication between student and instructor. It's a separate set of courses with a separate cost structure (undetermined at this writing). "We ourselves have come up over the learning curve over the last three years," says Leister. "It's been a three-year process of really understanding what the customer base needed and wanted. Now we feel we know, and that will be reflected in this new generation of software."

To help motivate students, the new course structure calls for more negotiation between student and instructor to determine the structure and time frame of the course. Students can also continue to communicate with an instructor after a course is completed. Other improvements to the software include word processing capabilities, improved user documentation, and the transmission of all lesson parts to the instructor (in the past, every third lesson was not required to be sent in via modem). Though only a limited number of courses will be available initially with the new software, eventually all courses offered will be included.

For more information on the Electronic University Network, write to: Office of Admissions, Electronic University Network, 505 Beach St., San Francisco, CA 94133; or call 800-22LEARN (California residents call 800-44LEARN).



BASEBALL

From the author of FOOTBALL comes an incredibly sophisticated, realistic and complete statistical baseball simulation for one or two players. Game play, manager's functions, graphics, and a statistical library establish BASEBALL as the ultimate sports simulation program.

Game Play

BASEBALL's underlying statistical framework simulates the realities of baseball like never before. Within this framework, the program considers each player's batting statistics against both left-handed and right-handed pitchers, and pitchers statistics vs. left- and right-handed batters. Every player's fielding and base-running abilities are also considered (an important factor when attempting to steal a base, etc.).

Select the team you'd like to manage, then pick the team you want to play against. Every team from the 1985 season is accurately represented, along with eight classic teams from the past. Determine your starting lineup, designate a starting pitcher, and make player substitutions when necessary. You call the plays, offense and defense. In a one-player game, your computer opponent displays uncanny intelligence in reacting to your managing decisions.

Manager's Functions

Built-in manager's functions provide extra versatility. You can trade team players, draft new players, create your own team, even form your own league. The program's statistics tracker keeps track of your team's season statistics and individual player year-to-date stats.

Graphics

BASEBALL's highly-detailed animated graphics give you a perfect view of the playing field. Each player acts and moves individually on every play. Three different stadiums are included with the program, and an optional Stadium Disk lets you play in any Major League stadium in the United States.

Statistics

BASEBALL can maintain a complete statistical record of each team player's performance. All player stats and game Box Scores can be displayed on the screen or sent to an external printer for a hardcopy printout.

BASEBALL also includes a unique Auto-Play option that lets the computer play a complete game in less than three minutes. A whole series of games can be played unattended, and an entire season of player and team statistics can be compiled over several days with all game stats printed out for your records.

See Your Dealer. . .

or write or call for more information. BASEBALL is available on disk for the Commodore 64 and Commodore 128 computers. For direct orders please enclose $49.95 plus $2.00 for shipping and specify UPS or first class mail delivery. Visa, MasterCard, American Express, and Diners Club cards accepted.

Commodore 64 and Commodore 128 are trademarks of Commodore ELectronics, Ltd.

MAJOR LEAGUE BASEBALL PLAYERS
MLB

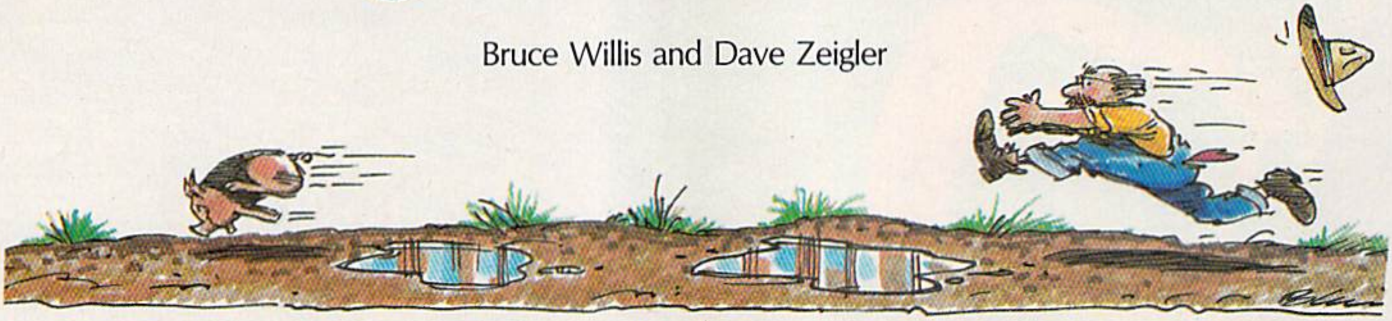
© 1986 MLBPA

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

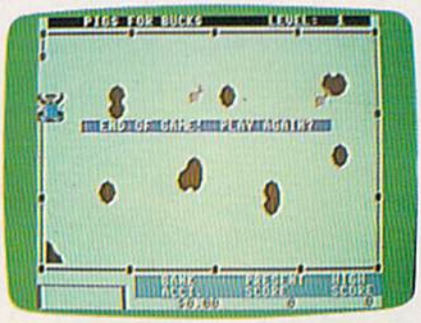
Order Line: (800) 637-4983
(except in Illinois, Alaska, and Hawaii)

Pig$ For Buck$

Bruce Willis and Dave Zeigler



Both children and adults will have hours of fun with this innovative and amusing game. The object is to catch each of your squealing and elusive pigs so you can take them all to market. And, as you'll see, keeping your overalls clean—a must as far as meticulous Farmer Brown is concerned—means you'll have to stay clear of the many mud puddles. For the Commodore 128. A joystick is required.



PIGS FOR BUCKS LEVEL: 1

END OF GAME - PLAY AGAIN?

BANK ACCT. $0.00 PRESENT SCORE 0 HIGH SCORE 0

Farmer Brown (left side of the screen) has just landed in a mud puddle, soiling his last clean pair of overalls—and ending the game.

As soon as we set eyes on the sprite and sound commands of Commodore's BASIC 7.0, we wanted to see if it was possible to write an arcade-style game almost entirely in BASIC. The result was "Pig$ For Buck$," a nonviolent game that's fun for children and adults. Just one word of warning: The game is extremely habit forming. And it's somewhat difficult: No one has yet made it past level 11.

Farmer Brown raises pigs, and now he must catch them to take them to market. But the price of pigs is dropping, so it must be done quickly to insure a profit.

Farmer Brown is somewhat vain—he must have clean overalls to wear to the market. The pigsty, however, is naturally slippery and full of mud puddles. Stepping in the mud or touching the electric fence will certainly cause him to slip and fall. His clean overalls will get muddy. But if he does fall, all is not lost. His clothesline contains four clean pairs of overalls, and he'll be able to buy up to four more pairs at the market.

Occasionally a pig will manage to squeeze out of a hole in the fence. Farmer Brown's assistant, positioned outside the fence, will eventually catch the loose pig and return it to the sty. Farmer Brown must be very cautious while leading a pig near the fence because the helpful but mischievous farmhand loves to see his employer fall in the mud. He just might push a pig back through the fence right into Farmer Brown's path.

The object of the game is to catch all the pigs in the pen as quickly as possible. The bank account total is added to your score, and the sooner you sell your pigs, the more profit you make. When all of your overalls have become muddy, the game ends.

Typing It In

The main part of the game is written in BASIC, but also included are one short machine language routine and three sprite definition files. (There are 24 sprites used in the game.)

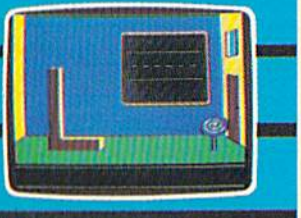
First type in Program 1, which is written entirely in BASIC. Be sure to save a copy when you're through typing. "MLX," a machine language entry program found elsewhere in this issue, is required to type in the four remaining files. After loading and running MLX, you'll be asked for a starting and ending address. Here are the correct values:

ILS INTERACTIVE LEARNING SERIES

FIREBIRD

TALKING TEACHER

SOFTWARE THAT TALKS AGES 2–8



Talking Teacher combines revolutionary software speech, superb graphics and engrossing play action to provide you with the next step in home computer software. No special hardware is required to make your computer talk! Talking Teacher contains three seperate learning games for children ages 2–8. The lessons teach:

- Capital and lower case letter identification and keyboard location
- A basic vocabulary and initial letter sounds of over 100 words.
- Spatial relationships (left, right, higher and lower).

Lesson One pronounces and prints each letter as it is typed. Children can type their names, short messages or just have fun with the keyboard! Ages 2–5.

Lesson Two Leads your child through the alphabet, pronouncing each letter and using it in a word. Typing the right letter causes a countryside filled with fantastic creatures and objects to unfold. Ages 3–6.

Lesson Three asks your child to identify the beginning letter of the words presented in Lesson Two. Mistakes are reviewed and correct answers are musically rewarded. Ages 4–8.

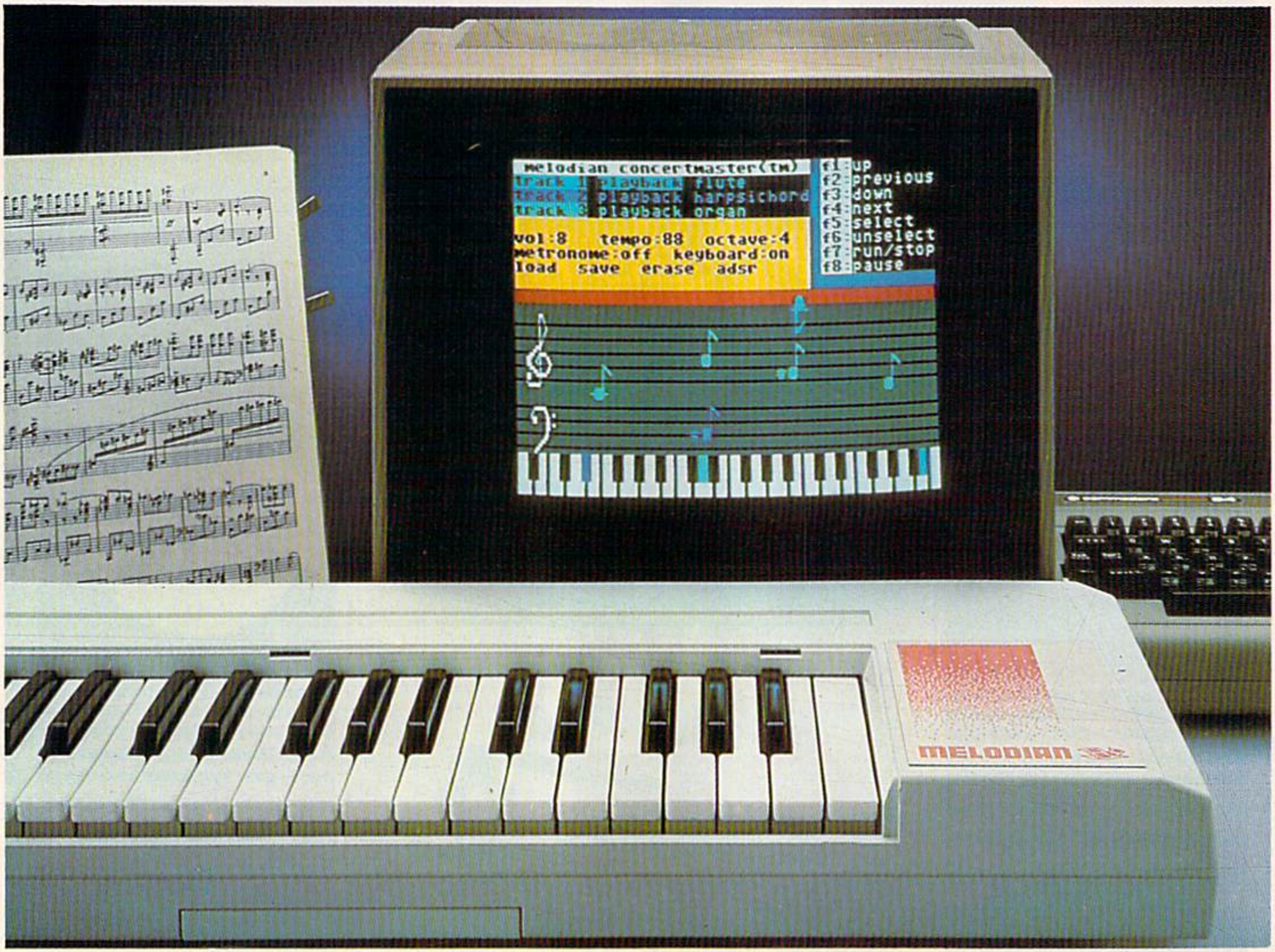
NOTE TO PARENTS: After loading, all three lessons may be enjoyed by your children without your direct supervision. However, the educational value can be enhanced by your participation and encouragement.

COMMODORE 64/128K

FACTSHEET/ORDER HOTLINE: 201.934.7373 FIREBIRD LICENSEES INC. P.O. BOX 49, RAMSEY, N.J. 07446

COMMODORE 64/128 ARE TRADE MARKS OF COMMODORE BUSINESS MACHINES.

Melodian will teach you to play, compose,



```
melodian concertmaster(tm)       f1:up
track 1 playback flute           f2:previous
track 2 playback harpsichord     f3:down
track 3 playback organ           f4:next
                                 f5:select
vol:8    tempo:88   octave:4     f6:unselect
metronome:off   keyboard:on      f7:run/stop
load  save  erase  adsr          f8:pause
```

A True Breakthrough In Music Education

At last, a program that makes it not only easy but fun to learn music. The Melodian keyboard and software were designed by Harry Mendell who designs custom synthesizer electronics and software for professional musicians such as **Stevie Wonder** and Eric Himy, an award winning concert pianist. The Melodian boasts many of the professional features found only on more expensive equipment. These features include multitrack recording, the ability to create custom instrument sounds and most importantly, ease of use.

Start your lesson with **RhythmMaster** Software. With its built-in metronome, RhythmMaster will display the treble and bass musical staffs and a picture of a piano keyboard. RhythmMaster will then play a measure of music and you must try to play the same measure back on the Melodian keyboard. You're not familiar with the keyboard or can't read music? No problem. RhythmMaster displays the notes you are to play on the musical staff and on the keyboard pictured on the monitor. If you strike the wrong key the note on the musical staff turns red and shows you which key you played wrong, making it ever so easy to correct what you played.

If you should hold a key too long a turtle runs across the screen. Inversely if you should release a key too quickly a rabbit scurries by. If you don't play it correctly RhythmMaster knows it and repeats the measure for you to play.

ConcertMaster teaches you how to play 35 pre-recorded songs from Bach to Rock. With ConcertMaster you can analyze music note by note, instrument by instrument and learn how a music composition is put together. Then you can compose your own music and record it right onto your floppy disks.

There are nineteen different instrument sounds to choose from in over a seven octave range giving you a wide choice of instruments to suit your musical taste and expression. You can also create your own instrument sounds.

ScoreMaster enables you to print out your music in standard music notation for other musicians to play, or for yourself.

New York Times Says . . .

Erik Sandberg-Diment of the New York Times states "really useful and instructive item ... Tanya, our 10 year old beginner quickly caught the spirit of matching the dance of her fingers to the measured metronome." "One piece of educational software that, unlike most of its kinfolk, actually delivers. These software-hardware combinations offer a lot of entertainment to the Commodore owner."

RUN Magazine Says . . .

Tom Benford of RUN notes "Whenever a selection of products of the same genre is available, one among the bunch rises head and shoulders above the rest. Such is the case with Melodian ConcertMaster keyboard and software. The combined features of RhythmMaster and ConcertMaster give you a complete music tutorial."

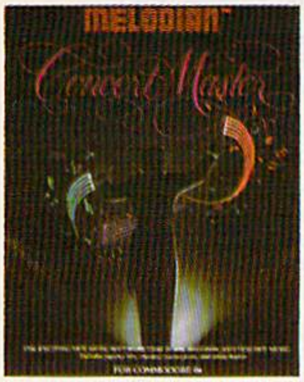
AHOY! Magazine Says . . .

Peggy Herrington of AHOY! said "The system is so easy to use that I didn't need the documentation". "It's fun, challenging, and educational, and for playability and ease of use it is nothing short of spectacular."

Satisfaction Guaranteed When You Buy Direct

By selling directly to you, we are able to give you the Melodian Keyboard and Software at far lower prices than ever offered before. You take no risk. **If the Melodian keyboard or any of the programs don't please you, for any reason whatsoever, send it back within 60 days for a full refund!**

and record music in just one evening!!



RhythmMaster Software rm-01 $59.95

RhythmMaster teaches a beginner how to read music and play it correctly and in rhythm on the musical keyboard.

RhythmMaster will have you reading and playing musical notes in minutes with fun and excitement.

RhythmMaster Features:
Trumpet, organ, violin, and synthesizer instrument sounds. Built in metronome. Pause/Play control. Set-up menu for customizing RhythmMaster.

RhythmMaster Teaches:
How to read notes on the treble and bass musical staffs,the names of the notes, where the notes are on the keyboard how to play whole notes, half notes, quarter notes, eighth notes and sixteenth notes in combinations, in both 3/4 and 4/4 time. How to play in different tempos.

RhythmMaster Requires:
A Commodore 64 or Commodore 128 with disk drive. Melodian Musical Keyboard kb-01 is required to study the reading and playing of musical notes.

Melodian Musical Keyboard kb-01 $99.95

40 Keys (A-C) in professional gauge spring loaded to give the feel and response of a real keyboard instrument. Polyphonic.

Registers (with ConcertMaster)
Organ, Trumpet, Flute, Clarinet, Piano, Harpsicord, Violin, Cello, Bass, Banjo, Mandolin, Calliope, Concertino, Bagpipe, Synthesizer 1, Synthesizer 2, Clavier 1, Clavier 2, which can be played over a 7 octave range. Programmable sounds as well.

Recording (with ConcertMaster)
Three track sequencer (recorder) with overdubbing and multitimbral (different instrument sounds at the same time) effects.

Interface
Built in interface for Commodore 64, Commodore 128, plugs right in to joystick port no. 2 and user port.

Power Supply
Powered direct by the computer, no batteries and cords required.

Finish
Table Model in white high-impact material, with carrying handle, protective key cover, and built in music stand. Size 29 -1/8 X 9-9/16 X 3-11/16, weighs 9 pounds.

ConcertMaster Software cm-01 $59.95

ConcertMaster teaches how a composition is put together, note by note, instrument by instrument. You learn to play 35 pre-recorded songs from Bach to Rock. Then you can compose your own songs and record them right onto your floppy disk.

ConcertMaster Teaches:
Scales, Bass lines, Familiar Beginner Songs such as "Jingle Bells", Easy classical songs such as "Bach Minuet" and Ravel's "Bolero", Advanced classics like "A Midsummer's Night Dream" by Mendelssohn, Popular hits such as "Thriller"

Instruments Sounds
Organ, Trumpet, Flute, Clarinet, Piano, Harpsicord, Violin, Cello, Bass, Banjo, Mandolin, Calliope, Concertina, Bagpipe, Synthesizer 1, Synthesizer 2, Clavier 1, Clavier 22, which can be played over a 7octave range. Programmable sounds as well.

Recording Functions:
Three track sequencer (recorder) with overdubbing and multitimbral (different instrument sounds at the same time) effects.
Each track can be set to one of seven different functions:

- **Monitor:** Lets you use a track to play music live, without recording it.
- **Record:** Records a track as you play.
- **Playback:** Lets you hear whatever has been recorded or loaded into the track. You may playback one track while recording another to build layers of instruments.
- **Mute:** Turns a track off. This is useful when you want to listen to or record one or two tracks at a time.
- **Save:** Stores a track to the disk.
- **Load:** Loads a track from the disk.
- **Protect:** Write protects a track.

Create New Instrument Sounds
Choose from pulse, sawtooth, triangle and noise sound sources. Control the sound envelope with attack, decay, sustain, and release times. Ring Modulation and Syncronization effects. Set Low pass, band pass, and high pass filter frequencies.

ConcertMaster Requires:
A Commodore 64 or Commodore 128 with disk drive. Melodian Musical Keyboard kb-01 is required to study the reading and playing of musical notes.

Melodian ScoreMaster sm-01 $59.95

With the ScoreMaster program your music can be printed out in music notation, which other musicians can read and play. Any music recorded with the ConcertMaster program can be printed by ScoreMaster.

ScoreMaster Requires:
A Commodore 64 or Commodore 128 with disk drive and printer compatible with the Commodore graphics mode such as the Commodore MPS 803, 1515, and 1525.
Melodian ConcertMaster program.

ACCESSORIES

Demonstration Disk	$ 9.95

RECORDINGS

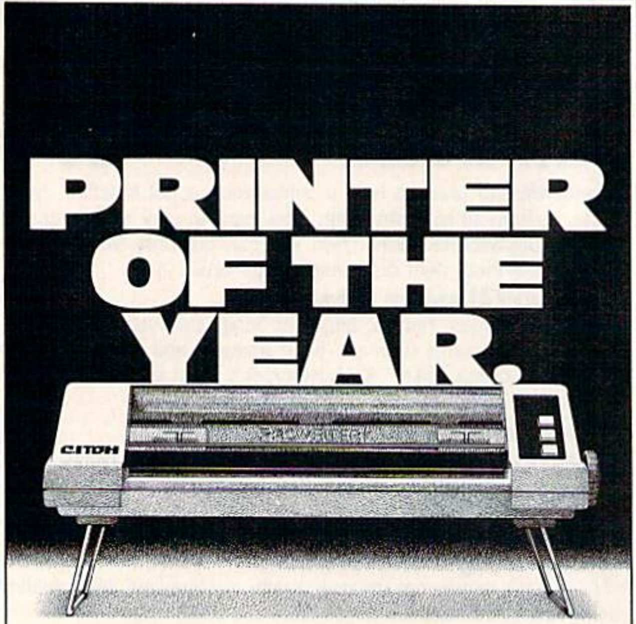
Christmas Carols	$12.95
Tchaikowsky Nutcracker	$12.95
Bach's Hits	$12.95
Classical Favorites	$12.95

FOR CREDIT CARD ORDERS, CALL TOLL-FREE
1-800-327-4566
IN FLORIDA, CALL 1-800-351-8777

For our International customers: Please send credit card number or international money order in U.S. dollars, or call **305-979-3777.** For Canada and Mexico, add $15.00 for air mail. Overseas add $16.00 for surface mail, and $70.00 for air mail.

MELODIAN™

The new, fast way to learn, play and compose music.



Program 2
Starting Address: 1E06
Ending Address: 1F05

Programs 3, 4, and 5
Starting Address: 0E00
Ending Address: 0FFF

Be sure to save each of these programs with its respective filename:

Program 2: PIG.ML
Program 3: PIG.SPR1
Program 4: PIG.SPR2
Program 5: PIG.SPR3

Also, be sure to save each of the five programs to the same disk.

Note that several lines in Program 1 are packed with commands. This was done to maintain speed during the main program loop. The decimal points used in many commands are not errors. They can be used in place of a zero and are slightly faster than variables or constants. Programs 1 and 2 make use of the FAST command, which blanks the screen during DATA POKEing and screen setup.

To start play, load Program 1 and type RUN. (The machine language program and sprite files will be loaded automatically.) Most of the screen is taken up by the pigsty. At the bottom of the screen is a clothesline, bank account balance, your present score, and the high score. The current level is displayed at the top of the screen. The selling price of a pig appears at the top right when one is sold.

The Pork Business Isn't Easy

With a joystick—plugged into port 2—chase the pigs and catch them by pressing the fire button when contact is made. But beware: If you collide with a pig while leading another pig out of the pen, you'll trip and fall in the mud. If you see that a collision is unavoidable, press the fire button to release the pig you are leading.

Lead each pig out of the gate in the upper-left corner and go catch the next one. The selling price of a pig is set as soon as you lead it out of the gate. The price starts at $100 at the beginning of a round and drops $6 every 10 seconds to a low of $60. The selling price is added to your present score, along with bonus points if the pig was caught before the selling price hit $60.

When all the pigs have been captured, you'll automatically purchase as many piglets as you can afford. Piglets—including the food needed to raise them—cost $60 dollars each. You can buy a maximum of seven per round. You'll then be given the option to purchase overalls to replace the ones taken from the clothesline during play. The store has four pairs in stock at the beginning of a game, and sells them for $25 a pair. When the last pair of pants from the clothesline gets muddy, the game ends, even if there are more pants in stock at the store.

After all pigs have been caught and sold, you advance to the next level. As the levels increase, so do the size and number of mud puddles on the screen. At higher levels, the speed of the farmer will vary.

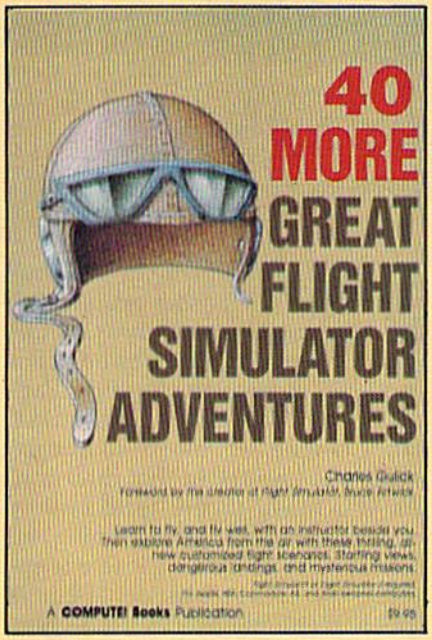
At the end of a game, you have the option to play again. The high score for the current session of play is displayed at the bottom right.

See program listings on page 121.

Fly Faster and Farther

with COMPUTE!'s

40 More Great Flight Simulator Adventures


40
MORE
GREAT
FLIGHT
SIMULATOR
ADVENTURES
Charles Gulick
Learn to fly, and fly well, with an instructor beside you. Then explore America from the air with these thrilling, all-new customized flight scenarios. Starting views, dangerous landings, and mysterious missions.
A COMPUTE! Books Publication
$9.95

This follow-up to the popular *40 Great Flight Simulator Adventures* brings you 40 more exciting, suspenseful flight simulator scenarios to help you get the most enjoyment from your *Flight Simulator* or *Flight Simulator II* software.

40 More Great Flight Simulator Adventures

Charles Gulick

Microsoft's *Flight Simulator* (IBM PC) and Sublogic's *Flight Simulator II* (Apple II, Commodore 64, Atari) are two of the most popular software packages today. They put you in the pilot's seat, letting you experience the thrill of flight. Now, *40 More Great Flight Simulator Adventures,* the second such guide, brings you more dazzling flights drawn from both fact and fancy.

Now you can fly where no one has flown before, with the author as your flight instructor. He helps you explore the unknown regions of flight simulation—taking you through cities, over mountains, under bridges—as you explore America from the air. Complete parameters and instructions at each step make flight easier and more exciting than ever before.

$9.95 ISBN 0-87455-043-2

Note: The *Flight Simulator* program itself is not included and must be purchased separately.

To order any COMPUTE! book, call toll free 800-346-6767 (in NY 212-887-8525). Or mail in the attached coupon with your payment. Include $2.00 shipping and handling per book.

Please send me ________ copies of *40 More Great Flight Simulator Adventures* at $9.95 each. (ISBN 0-87455-043-2)

All orders must be prepaid in U.S. funds.

Subtotal ________

NC residents add 4.5% sales tax ________
NY residents add 8.25% sales tax

$2.00 shipping and handling
per book
($5.00 per book airmail) ________

Total amount enclosed ________

COMPUTE! Books
P.O. Box 5038
F.D.R. Station
New York, NY 10150

□ Payment enclosed (check or money order)

□ Charge □ Visa □ MasterCard □ American Express

Name ________________________________

Address ________________________________

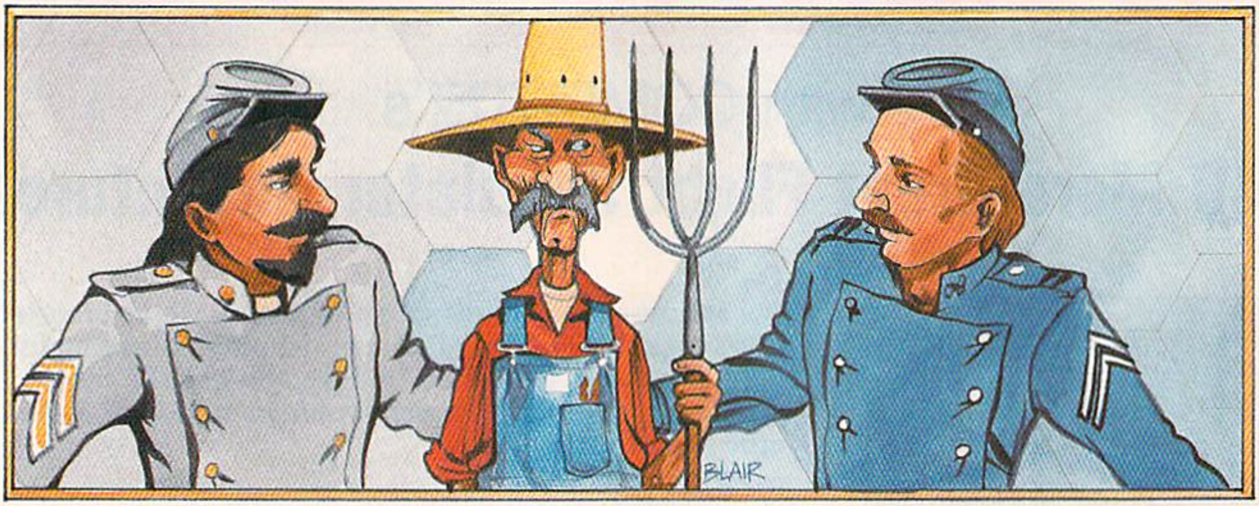
City ________________ State ________________ Zip ________________

Please allow 4–6 weeks for delivery.

COMPUTE! Publications,Inc. abc

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

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.



Persuasion

Kevin Mykytyn and Mark Tuttle

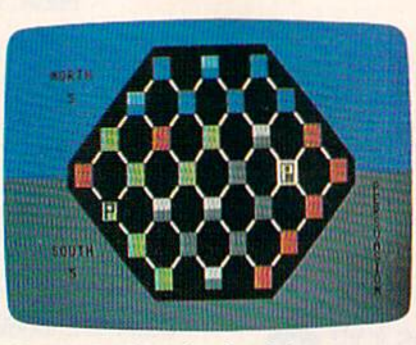
Match wits with a friend in this unique strategy game. There are no random game elements, so the best player always wins. For the Commodore 64. A joystick is required.

"Persuasion" is a different kind of strategy game. Players take turns "persuading" neutral zones to join their domain. There are a few obvious thematic similarities to the American Civil War—North vs. South, blue vs. gray—but the comparison ends there. Persuasion is a nonviolent game with no random factors. Like checkers or chess, the outcome is decided entirely by player movements.

Blue Vs. Gray

After typing in the program, save a copy to tape or disk. To play, type RUN. Be sure to plug a joystick into port 2 (both players share the same joystick in this game). Player 1, who starts at the top of the playing field, is the North player. Player 2, the South player, begins at the bottom. Player 1 always moves first.

As you can see in the game photo, the playing field consists of 30 locales, each colored red, laid out in a hex pattern. The field is positioned on top of a screen divided by the players' colors, blue and gray. The number of locales owned is shown in each player's color area on the left. On the right is the game title—Persuasion—which is alternately displayed in the blue or gray area to indicate whose turn it is. To win, you must capture 16 locales. However, a tie is possible if each player captures 15 locales.



The North player has landed on a neutral locale and changed it from red to green. The South player has just claimed a locale and tied the score.

To begin the game, move your piece (a yellow border around the active player's piece always indicates whose turn it is) to an adjacent locale by moving the joystick diagonally to the desired destination. For example, to move southeast, move the joystick in that direction. As you move to the locales, they change color. At the beginning, all are red. The first time a locale is entered by either player, it turns green. If either player enters a green locale, it turns two colors—light blue and dark blue for the North player, or light gray and dark gray for the South player. With one more reentry by the same player, the locale turns a solid color (blue or gray), and that locale is in the player's domain. The score is then updated.

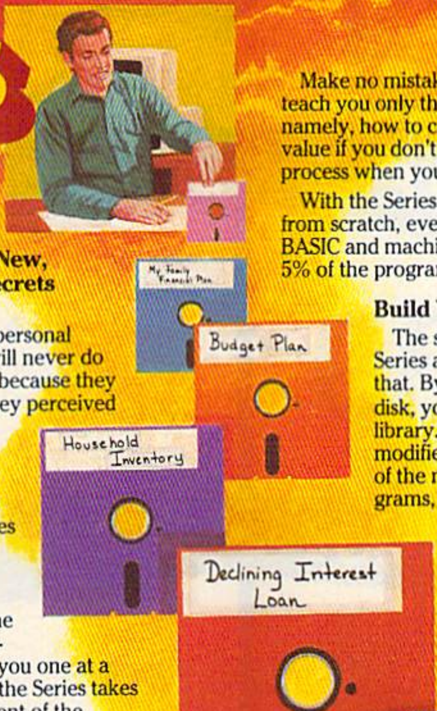
If the opposing player enters a two-color locale, one that's about to fall to his or her rival, it's turned back to green. Once a locale has been won over, a movement into that area by the owning player will turn it back to red, the neutral color. This is an important feature because it may determine your strategy. As you'll see, it's not difficult to box yourself in where your only move is to reenter a controlled locale, thus undoing all your hard work. Once a locale is totally owned (solid gray or solid blue), the opposing player cannot change it.

Persuasion might appear a bit complicated at first, but after a few minutes with your first game, you'll see that it's easy to play. A good strategy is the key to winning.

See program listing on page 107.

Make Any Computer Do Exactly What You Want With McGraw-Hill's

Contemporary Programming & Software Design Series



From Writing Your Own Programs to Modifying Existing Software, Here's the New, Easy, and Low Cost Way to Unlock the Secrets of Your Computer

Whether you use computers for business, for personal applications, or for fun, off-the-shelf programs will never do everything you want them to do for you. That's because they were written by programmers to satisfy what they perceived as the needs of the greatest number of potential users—often missing some or many of your specific needs.

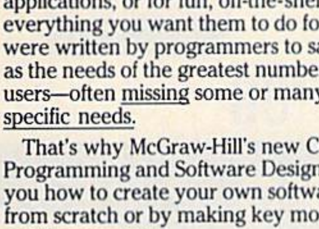
That's why McGraw-Hill's new Contemporary Programming and Software Design Series teaches you how to create your own software. . . either from scratch or by making key modifications to existing programs.

There is nothing magical about it. You learn the process of building a computer program step-by-step with McGraw-Hill *Concept Modules* sent to you one at a time, once a month. Each of the ten modules in the Series takes you through an important step in the development of the structure and detailed logic of a program, including testing, debugging, and documentation.

Unique Interactive Hands-On Instruction

Each module includes an easy-to-understand guide PLUS a 5¼" floppy disk containing typical programs and interactive instruction that you can run on Commodore 64 and 128 computers, IBM PCs and PC compatibles for hands-on experience.

In the first Module, for example, when your sample program (Declining Interest Loans) appears on your screen, you'll find errors on certain program lines. You'll also see that the program is only three-quarters completed.



Now comes the fun part. You'll discover how this program is built, and in the process you'll learn how to identify and correct errors. And by the end of Module 1, you'll actually have completed this program yourself.

But there's more. Special graphics on your screen work in conjunction with the accompanying guide to amplify, illustrate, and deepen your understanding of software design principles.

The Crucial 95%—Learn the Foundation of Computer Programming

While the Series includes interactive disks that run on specific computers, everything you learn you can apply to any language or machine. Why is this possible? Because McGraw-Hill knows programming is far more than coding a program into the computer using a specific language. In the real world of computers, 95% of the programming process is carried out using design techniques that are independent of specific language or machine. It is this crucial 95% that you thoroughly understand and master in the Series.

Make no mistake. Almost all books and courses on "programming" teach you only the final 5% of the total programming process—namely, how to code in a specific language. . . information of little value if you don't know how to reach the point in the programming process when you are ready to code.

With the Series, however, you'll learn to create your own programs from scratch, even modify off-the-shelf programs. You'll learn enough BASIC and machine language to get you started on the remaining 5% of the programming process.

Build Your Own Personal Software Library

The sample programs you work with throughout the Series are excellent learning tools. But they're more than that. By combining the sample programs onto one master disk, you'll have the start of your own personal software library. In addition to the programs you've written and modified throughout the Series, you'll also receive dozens of the most popular public domain and user-supported programs, such as data base manager, word processor, calendar generator, appointments reminder and much, much more.

15-Day No-Risk Trial

To order your first module without risk, send the postage-paid card today. Examine the first module for 15 days and see how the Series will help you make your computer do exactly what you want it to do!

If someone has beaten you to the card, write to us for ordering information about the Contemporary Programming and Software Design Series.

McGraw-Hill
Continuing Education Center
3939 Wisconsin Avenue
Washington, DC 20016

PRO-TECH TRONICS

MONITORS

RETAIL
$120

$59

GREEN SCREEN

RETAIL $299

$99

FULL COLOR

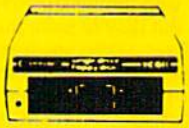
RETAIL $400

$238

RGB

DISK DRIVE

1541
COMMODORE



$139

MODEMS

300
BAUD

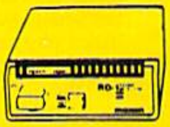
$28

1200 BAUD
HAYES COMPATIBLE

$88

RADAR DETECTORS

HIGH
PERFORMANCE



SUPERHET

RETAIL
$249

NOW
$78.00

FREE HOME TRIAL

VIDEO

VHS-HQ

CAMERA
& VCR
ALL IN ONE!

- LOW LIGHT
- LIGHT WEIGHT

$1299

$1600
RETAIL

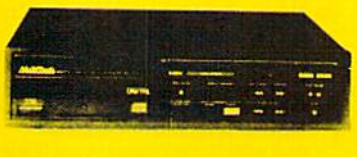
CORDLESS PHONE

RETAIL
$189

$99

FREE TRIAL

COMPACT DISC PLAYER



"TOP OF THE LINE"

RETAIL
$289

$169

FREE HOME TRIAL

CALL
TODAY
1-800-345-5080

COMMODORE

C-64 COMPUTER

NEW UNITS

$99

*WITH PURCHASE OF OUR SPECIALLY PRICED SOFTWARE

COMPUTER

128k

INCLUDES DISK DRIVE

$395

Laser 128

FREE HOME TRIAL

PRINTERS

DOT MATRIX

RETAIL $200

$69

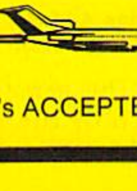
COMMODORE

FREE HOME TRIAL

NEAR LETTER QUALITY

$149

POWER SUPPLY



$39

C-64

CAR STEREO

FM/AM CASSETTE

$39

FREE CAR TRIAL

QUICK-DELIVERY

MasterCard AMERICAN EXPRESS VISA C.O.D.

SCHOOL P.O.'s ACCEPTED

PRO-TECH-TRONICS
6870 Shingle Crk. Pkwy. #103
Minneapolis, MN 55430
(612) 560-6603

Ringside Karate

Anthony Bertram

In this outstanding commercial-quality, arcade-style game, you go head-to-head with another player to see who deserves a black belt. For the Commodore 64. Two joysticks are required.

As in real karate, skill, speed, and timing are the necessary ingredients to play "Ringside Karate." Each player controls one fighter with a joystick. Moving the stick right and left moves the fighter forward and back. To punch, move the joystick toward your opponent and press the fire button. To kick, push the joystick away from you (diagonally, northeast for the left player and northwest for the right player) and press the fire button. To footsweep, move it southeast (left player) or southwest (right player) and press the fire button. Pressing the fire button in any other position causes a defensive block. To win, you must score ten knockdowns, each worth 100 points.

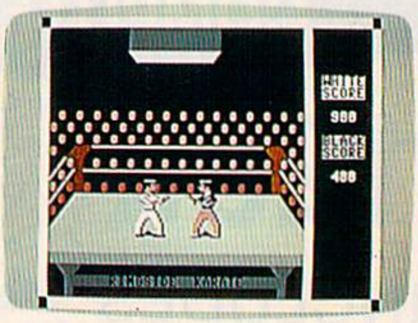
Timing is crucial because the fire button must be released for a moment before another attack can be made. If the button is pressed rapidly, the fighter won't move and will likely be knocked down. Blocking provides a limited defense and is best used just before a counterattack or when backing up. Those who have played an arcade karate game will find the controls easy.

Typing It In

Ringside Karate is written entirely in machine language. "MLX," the machine language entry program found elsewhere in this issue, is required to type it in. After loading and running MLX, enter the following addresses at the prompts:

Starting Address: 3000
Ending Address: 45BE

The game is fairly long, but much of the sprite and screen DATA are repeated numbers, which makes the typing a little easier.



The white player needs one more knockdown to win this match.

To load the game, disk users should type LOAD "*filename*",8,1 (where *filename* is the name you used to save the program). Tape users should type LOAD "*filename*",1,1. To start the game, type SYS 15490. The joystick plugged into port 1 controls the fighter dressed in white, and the joystick in port 2 controls the fighter in black. When the game is over, a winner is declared, and another game may be started by pressing any key.

See program listing on page 109.

Introducing a new printer that's based on an old favorite.

The new Star Gemini II. The printer that was made for your Commodore PC.

We took all the best features from our original Gemini,® our best-selling printer for Commodore® users.

Then we added enhanced Near Letter Quality printing... a built-in Commodore interface (with cable included)... a rear tractor feed... and an array of front panel controls that makes printing easier than ever.

The result? The new Star Gemini II. A printer designed to help your Commodore's performance. Without hurting your budget.

Now at Toys 'R' Us and other major chain stores.

In front with quality up front.

star®

The new Star Gemini II.

® Commodore is a registered trademark of Commodore Business Machines, Inc.

LYCO COMPUTER

MARKETING & CONSULTANTS

NEW HOURS

Mon-Thur—9 AM-8 PM
Fri—9 AM-6 PM
Sat—10 AM-6 PM

PRINCETON GRAPHICS

MAX-12 Amber	175
HX-12 RGB	458
SR-12 RGB	575

COMMODORE

1571	CALL
1541	CALL

INDUS

GT Commodore	179

COMTEL

Enhancer 2000 (C-64)	149

COMMODORE

128	CALL
C 1902-A	CALL
C 1350 Mouse	39
C 1700 128 K RAM	145
C 1750 512 K RAM	269
Sane	35
Perfect Writer	45
Perfect Calc	45
Perfect Filer	45

EPYX-64

Movie Monster	24.75
Vorpol Kit	22.75
Fast Load	24.75
Summer Game II	24.75
Super Cycle	24.75
World Games	24.75
Football	24.75

5¼" DISKETTES

MAXELL

SSDD	9.99
DSDD	12.99

VERBATIM

SSDD	9.50
DSDD	12.99

BONUS

SSDD	6.99
DSDD	7.50

SKC

SSDD	8.50
DSDD	9.50

3.5" DISKETTES

3M

SSDD	16.99
DSDD	23.99

MAXELL

SSDD	16.99
DSDD	23.99

VERBATIM

SSDD	16.99
DSDD	24.99

SKC

SSDD	14.99
DSDD	19.99

XETEC

Font Master II 64	32.95

TOLL FREE 1-800-233-8760

ACTIVISION
(C-64/128)

Alter Ego	28.75
Hacker	18.75
Little People	20.75
Game Maker	24.75
Borrowed Time	18.75
Space Shuttle	18.75
Music Studio	22.75
Mind Shadow	18.75
Road Race	18.75
Fast Tracks	22.75
Count Down	18.75
Basketball	18.75

SUBLOGIC

Nightmission Pinball	18.95
Flight Simulator	29.95
Set Simulator	25.95
Football	25.95
Scenery Disk ... Each	14.95
Set 1-6	69.96

BATTERIES INCLUDED

Paperclip	35.95
Consultant	35.95
Paperclip W/Spell Pack	48.95

BRODERBUND

The Print Shop	24.75
Graphics Library I, II, III	15.75
Karateka	17.75
Bank Street Writer	29.75
Lode Runner	19.75
Printshop Comp.	24.75
Bank Street Speller	29.75
Bank Street Filer	29.75
Bank Street Mailer	29.75
Championship Lode Runner	19.75

SSI

Colonial Conquest	22.75
Wings of War	22.75
Computer Ambush	32.75
Field of Fire	22.75
Fighter Command	34.75
Kampfgruppe	34.75
Mech Brigade	34.75
Market Garden	22.75
Sux Gun Shootout	22.75
Computer Baseball	22.75
Computer Quarterback	22.75
Imperium Galactum	22.75
Phantasie	22.75
Cartels & Cutthroats	22.75
50 Mission Crush	22.75
Questron	22.75
Gemstone Warrior	22.75
Battellion Commander	22.75
Panzer Grenider	22.75
Norway 85	22.75
U.S.A.A.F.	34.75
Battlegroup	34.75
NAM	22.75
Phantasie II	22.75
Wizards Crown	22.75
Rings of Zelphin	22.75

ACTIVISION
(Amiga)

Hacker	26.75
Mind Shadow	26.75
Music Studio	34.75
Borrowed Time	26.75

ACCESS
(C-64 — Amiga)

Leader Board	24.75
Leader Board Amiga	24.75
Tournament I	12.75
MACH 128	28.75
10th Frame	24.75

Buy Lyco and Enjoy

★ THE LOWEST PRICES ★ TOLL FREE ORDER LINE ★

★ Free shipping on prepaid cash orders in U.S. ★ All Merchandise Factory Fresh ★ Fast Service from one of the oldest and most established Computer Supplier ★ 24 hrs. shipping on in-stock product ★ Access to our Multi Million $ inventory ★ No deposit on UPS C.O.D. orders ★ Orders outside PA Save state sales tax ★ Air freight service available ★ Full Manufacturer's Warranty apply! ★ Full accessory line in stock ★ Purchase orders Accepted from educational institutions! ★ We check for stolen credit cards! ★ We ship to our servicemen overseas! ★ You'll love our Courteous Sales Staff! ★ We satisfy more customers monthly than most of our competitors combined ★

LYCO COMPUTER

MARKETING & CONSULTANTS

C.ITOH

1550 SP+	CALL
D1040	CALL
Prowriter Junior	CALL
Prowriter 8510SP+	CALL

PANASONIC

1080	195
1091	225
3131	249
1092	309
1592	419
1595	595

OKIDATA

Okimate 10XE	179
292	CALL
293	CALL
Okimate 20ST	199
120 NLQ	205
182	214
192	348
93	CALL

JUKI

Juki 6100	CALL
5510 Juki	CALL
Juki 6300	CALL
RS-232 Serial Board	55
Color Kit	105
6100 Tractor	119
6100 Sheet Feeder	209

EPSON

LX80	CALL
FX85	CALL
DX10	CALL
HI80	CALL
HS80	CALL
FX286	CALL
LQ800	CALL
LQ1000	CALL

STAR MICRONICS

LV 12-10 (New)	CALL
NL-10	CALL
NX-10	CALL
NB-15	CALL
SB-15	CALL
SG-15	367
SD-10	319
SD-15	438
SR-10	469
SR-15	578
SB-10	589

1080 .... $195 **SAVE** ON THESE IN STOCK **PRINTERS** **NX-10 ... CALL**

SEIKOSHA

SP-1000 VC (C-64)	175
SP-1000 A Centronics	195
SP-1000 IBM	195
SP-1000 As RS-232	195
SP-1000 AP Apple IIC	195
BP-1300	469
BP-5200	649
BP-5420	999
Color Kit	119
BP-5420 Ribbon	12.50
SP-1000 Ribbon	8.50

CITIZEN

120-D	179
MSP-10	285
MSP-15	385
MSP-20	325
MSP-25	485
Premier 35	469

LEGEND

808	148
1080	CALL
1380	229
1385	289

BROTHER

HR-15XL-P	359
HR-15XL-S	359

SILVER REED

EXP 420P	CALL
EXP 600P	489
EXP 800P	649
EXP 770	740

DIABLO

D25	549
P-32 CQI	699
635	1029
P-38	1749
D-80 IF	2395

TOSHIBA

321 P/S	495
351 Sheet Feeder	529
P 341P	969
P 341S	999
P 351+	1149

MONITORS

TEKNIKA

MJ-10	149
MJ-22	249
MS-305 RGB	309

HITACHI

MM-1218 12" Green	99
CM-1406 13" Color W/Cable	179

PANASONIC

TR-122 MYP 12" Amber TTL	139
TR-122 MYP 12" Green TTL	139
TX-12H3p 12" RGB	369
DT-H103 10" RGB	349

THOMPSON

365 12" RGB	CALL

NEC

Multisync	CALL

ZENITH

ZVM 1220	89
ZVM 1230	89
ZVM 1240	149

INTERFACING

CARDO

G-Wiz (C-64)	47
Super G (C-64)	54
C/? PS (C-64)	37

XETEC

Super Graphix 64	64
Super Graphix SR 64	45

TYMAC

Connection (C-64)	49

ORANGE MICRO

Grappler CD (C-64)	79

MODEMS

DSI

Messenger 64/128	39.95

SUPRA

1064 Modem (C-64)	49.95

COMMODORE

1670 Modem	155

Compuserve....18.95

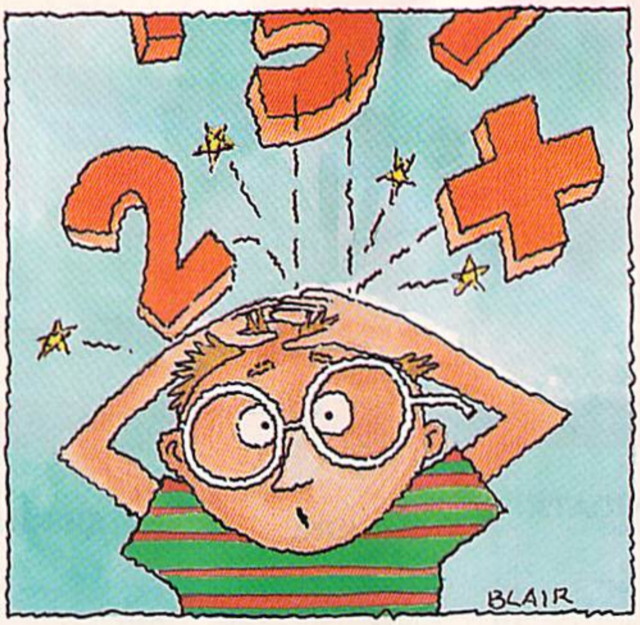
TOLL FREE 1-800-233-8760

In PA 717-494-1030
Customer Service 717-494-1670

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

RISK FREE POLICY

In stock items shipped within 24 hours of order. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. Volume discounts available. PA residents add sales tax. APO, FPO, and international orders add $5.00 plus 3% for priority mail service. Advertised prices show 4% discount for cash, add 4% for MasterCard and Visa. Personal checks require 4 weeks clearance before shipping. We do not guarantee compatibility. We only ship factory fresh merchandise. Ask about UPS Blue and Red label shipping. All merchandise carried under manufacturer's warranty. Return restriction applicable. Return authorization required. All items subject to change without notice.



Math Attack

Dan Dickerman

School children of any age can sharpen their math skills with this clever educational program for the Commodore 64. It offers a choice of addition, subtraction, and multiplication challenges at four levels of difficulty.

Many of today's educational programs don't hold a child's attention. Others do—but they're better as games than as educational tools. My goal in writing "Math Attack" was to present a riveting game—with lots of color and sound—in which math problems played the central role. That way, players would receive painless drills in the basic math skills. I designed the math problems as sprites, so they could move smoothly about the screen to play an active role in the game.

Math Attack is easy to learn and easy to play. The premise is fairly simple: Sets of math problems have begun to descend upon an almost helpless city, and the only weapons available to defend it are the correct answers. In testing this game, I found that many of the young testers quickly overcame their initial fear of math problems once they saw how easily a descending swarm of them could be eliminated.

The Player's Choices

After typing in the game and saving a copy, load it and type RUN. The screen is cleared and the city assembled. The player is then asked which type of problem he or she wishes to tackle: addition, subtraction, or multiplication. To respond, press either the +, −, or x key. I used the letter x to represent multiplication because it's similar to the symbol children use in school. Also, I avoided division to lend a more common appearance to the problems and to avoid other complications such as fractions and long decimals.

The player next chooses a starting level of difficulty (selected by pressing either 1, 2, or 3) in which upper limits are set on the sizes of the numbers in the incoming problems. This also provides the player with slower starting speeds at higher levels—a kind of reward for choosing a more difficult problem to begin with.

Now the onslaught begins. Three math problems appear in the sky, descending menacingly toward the city. The player enters answers from the keyboard, and launches them via the RETURN key. The − key should be used before the answer if a negative value is required, and typos may be erased using the INST/DEL key. A correct answer destroys an attacking problem and lifts the sprite (math problem) beyond the top of the screen, where it soon reappears as a new problem. An incorrect answer sends the whole set of problems closer to the city or, in some cases, increases the rate of their descent. If a problem reaches the city,

Calling All COMMODORE *Users!*

COMPUTE!'s
Machine Language Games for the COMMODORE 64
Seven outstanding machine language games for the Commodore 64 and 128 (in 64 mode). Dodge missiles, slide on ice, run for president, and much more. Complete source code for all games included.
A COMPUTE! Books Publication $16.95
MAPPING THE COMMODORE 128
Ottis R. Cowper
A comprehensive memory guide for BASIC and machine language programmers.
A COMPUTE! Books Publication $16.95
The COMPLETE 64
Seven powerful, sophisticated applications on disk including SpeedScript and SpeedCalc, COMPUTE!'s excellent word processing and spreadsheet programs. For the Commodore 64 and 128 personal computers.
A COMPUTE! Books Publication $29.95
PROGRAM DISK Enclosed

Three new books from COMPUTE! bring you fully tested and completely documented machine language programming tutorials, exciting games, valuable utilities, and easy-to-use applications for home, business, and school.

Exceptional new books from COMPUTE! for your 64 and 128.

Mapping the Commodore 128

Ottis R. Cowper

This clearly written, comprehensive memory map and programmer's guide, written by the technical editor of COMPUTE! Publications, provides a detailed explanation of the inner workings of the Commodore 128 computer. Topics include memory management, BASIC 7.0, I/O chip registers, the operating system, system RAM, and much more. This is an important reference for both BASIC and machine language programmers.

$16.95 ISBN 0-87455-060-2

COMPUTE!'s Machine Language Games for the Commodore 64

Edited

This collection of the best machine language games from *COMPUTE!* magazine and *COMPUTE!'s Gazette* is a must for 64 game players and machine language programmers. The book includes both the source code (studying these listings is an excellent way to improve your machine language programming) and object code for the popular games "Cut-off!," "Campaign Manager," "Nessie," "Miami Ice," "Heat Seeker," "Whirlybird," and "Laser Beam." *A disk is available for $12.95 which includes programs in the book, 610BDSK.*

$16.95 ISBN 0-87455-061-0

The Complete 64

Edited

A book/disk combination with seven powerful, sophisticated applications on disk. The book has complete, easy-to-follow documentation for each of the six applications. It features *SpeedScript* and *SpeedCalc,* COMPUTE!'s commercial-quality word processing and spreadsheet programs; "Plus/Term," an easy-to-use, yet versatile, telecommunications program; and "Sidplayer," a powerful music editor and player. "Hi-Resolution Sketchpad," "Ultra-Font +," and "Mini-filer" are also included. All the applications are on a disk ready to load and use on the Commodore 64 and 128 running in 64 mode.

$29.95 for book/disk package

ISBN 0-87455-062-9

Visit your local book or computer store and ask for one of these new summer releases from COMPUTE! Books.

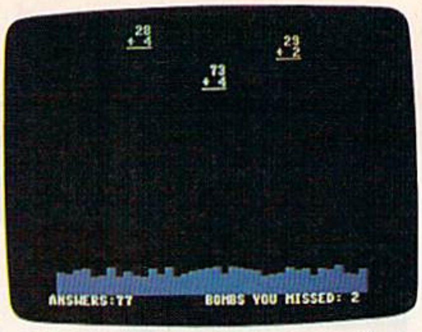
You can order directly from COMPUTE! by calling toll-free 800-346-6767 (ih NY 212-887-8525) or writing COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please include $2.00 per book or disk for shipping and handling in U.S. and surface mail or $5.00 per book airmail. NC residents add 4.5 percent sales tax. NY residents add 8.25 percent sales tax.

Please allow 4–6 weeks for delivery.

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

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.



Three menacing multiplication problems descend toward the city—and the only defense is the correct answers.

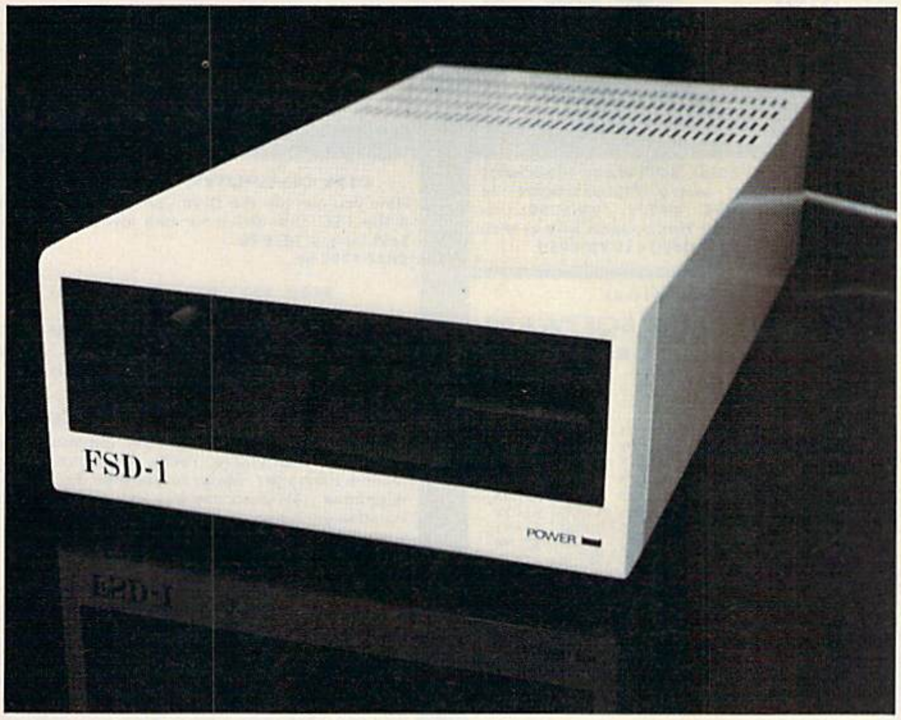
it explodes into a cloud which then retreats to the top of the screen. It then regenerates as another problem to fall from the skies.

After 20 problems have been correctly answered, the player is informed that the level has been completed, and that the new limits on the problems' sizes have been increased by a factor of ten. In all, there are four basic levels, with combinations of one-, two-, and three-digit numbers, the highest level having problems comprised of two- and three-digit numbers. This final level cannot be accessed directly at the start of the game. It can only be reached by progressing from the lower levels.

The game ends either when the fourth level has been successfully completed (that is, 20 problems on this level have been correctly answered) or after three problems have reached the city and exploded on its surface. At the end of the game, the player is presented with the number of correct and incorrect answers given during the course of the game, and the number of math problems reaching the city (which will always be three, unless the player completes the fourth level). It's up to the parent or teacher to decide whether the student may use a calculator.

When played by children just learning to solve simple math problems, Math Attack provides an exciting and effective drill of the basics, which increases in difficulty as their skill grows. For us older kids, it will stump the best of us on its higher levels. In fact, I challenge anyone to keep the city intact through the fourth level in multiplication without using a calculator.

See program listing on page 112.



The FSD-1 Disk Drive

Just when everyone thought the Commodore 64 was on its last legs, the arrival of new, improved, and innovative hardware and software products not only revived the 64, but stretched its capabilities to new limits.

In the hardware area, new peripherals are faster and more sophisticated than their predecessors. Collectively, third-party manufacturers have developed products which have expanded the capabilities of the 64 and improved the overall quality and reliability of Commodore computer peripherals. One such entry into the market is the FSD-1, a 1541-compatible disk drive from Emerald Components International (ECI). Produced in Japan by the original manufacturer of Commodore's 1541 drive, the FSD-1 is a significant improvement over the 1541: It's less expensive, sturdier in construction, and more reliable.

In specifications, the FSD-1 is identical to the 1541 in capacity (174K storage, 144 directory entries, 256 bytes per sector, 35 tracks), and while the FSD-1 has somewhat faster data loading and writing speed, it is not a speed demon. Capacity and speed are two sacrifices necessary to obtain true compatibility with the 1541. A mistake some other 1541 compatibles make is to try to improve speed and capacity, but this only yields significant impairment of compatibility with the 1541 and software designed to run on the 1541. The FSD-1, however, does have a larger memory buffer than the 1541. Also, its reliability is superior to that of the 1541, with an average MTBF (mean time between failures) of 8000 hours.

The FSD-1's shape is sleek and its construction is sturdy. It's considerably smaller than the 1541, with dimensions of 2-3/4 × 6-3/8 × 13 inches. It weighs 6-1/4 pounds. The FSD-1 has a metal case which, combined with the built-in solid state power supply, allows the FSD-1 to run cool and avoid the overheating problems of the 1541. It also eliminates any of the radio frequency interference sometimes associated with the 1541 and its plastic case. In operation, the FSD-1 runs quietly, with only an occasional faint whirring noise to give you a clue that the drive is in operation. The only head knocks I heard using the FSD-1 were caused by software copy protection.

As you would expect, the FSD-1 has dual serial ports to daisychain disk drives and printers. An unexpected (and welcome) feature, however, is a DIP switch on the bottom of the drive to permit the drive's device number to be changed. You can easily change the number of the drive to 9, 10, or 11 without the need to open up the case and cut and solder wires (as you must do with the 1541 to permanently use it as a second drive).

The user's manual accompanying the FSD-1 is brief, but has a complete explanation of setup instructions, disk commands, and troubleshooting tips. Since the manual is a bit short on technical data, advanced users may wish to write ECI for more information. The FSD-1 is warranted by the manufacturer for six months, which is three months longer than most computer peripheral warranties cover.

Where the FSD-1 really shines is in its compatibility with software designed to run on the 1541. The FSD-1 appears to be truly 100-percent compatible with the 1541. Every piece of 64 software I tried (over 300 programs) loaded on the FSD-1. The programs I loaded using the FSD-1 included some of the latest available and have some of the most complicated copy protection schemes: *The Newsroom, GEOS, Paperback Writer* (now *Pocket Writer*), *Vizastar*, and *Elite*. *GEOS* was a particularly good test of the FSD-1, since it not only has sophisticated copy protection, but also utilizes its own operating system, has a built-in disk drive speedup feature, and implements a virtual memory routine that requires extensive disk access. The FSD-1 performed flawlessly with *GEOS*.

Loading has been a problem with some compatible disk drives, since copy protection schemes often depend heavily on the intricacies of the 1541 ROM. This apparently is not the case with the FSD-1. In fact, the FSD-1 is more compatible with the 1541 than the Commodore 1571. The FSD-1 can load several programs that the 1571 has problems with (or at least had problems with until the software manufacturers corrected the incompatibility difficulties).

COMMODORE 64 COMPUTER

(Order Now)

$149⁹⁵

- C128 Disks 44¢ ea.*
- Comstar Writer 64 $19.95
- 13" Color Monitor $139.95

COMMODORE 64 SYSTEM SALE

Commodore 64

Com. 1541 Disk Drive

13" Color Monitor

Plus $30.00 S&H

$457

(Ltd. Qty)

C128 Commodore Computer & 1571 Disk Drive

$499⁰⁰

- Voice Synthesizer $39.95
- 12" Monitor $79.95

CALL BEFORE YOU ORDER

SPECIAL SOFTWARE COUPON

PRICES MAY BE LOWER

COMMODORE 64 COMPUTER $149.95

You pay only $149.95 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your computer that allows you to SAVE OVER $250 off software sale prices!! With only $100 of savings applied, your net computer cost is $49.95!!

* C128 DOUBLE SIDED DISKS 44¢ EA.

Get these 5¼" Double Sided Floppy Disks specially designed for the Commodore 128 Computer (1571 Disk Drive). 100% Certified, ***Lifetime Warranty***, Automatic Lint Cleaning Liner included. 1 Box of 10 - $4.40 , 5 Boxes of 10 -$22v00., 10 Boxes of 10 -$44.00 . Boxes of 10 limit 1,000. (Add .50 per box for shipping & handling)

13" COLOR MONITOR $139.95

You pay only $139.95 when you order this 13" COLOR MONITOR. LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your monitor that allows you to save over $250 off software sale prices!! With only $100 of savings applied, your net color monitor cost is only $39.95. (16 Colors).

Premium Quality 150-170 CPS Comstar Aero 160 Printer $199.00

The COMSTAR Aero 160 gives you a 10" carriage, 150-170 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin setting, true lower decenders with super and subscripts, prints standard, block graphics and special characters. It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) List $499.00 **Sale $199.00**.

9" SAMSUNG GREEN SCREEN MONITOR

Super High Resolution composite green screen monitor. Perfect for 80 column use with The C128 computer (Req. $19.95 Cable)
List $129.95 **Sale $59.95**.

80 COLUMNS IN COLOR COMSTAR WRITER 64 WORD PROCESSOR $19.95

This COMSTAR WRITER 64 WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE FOR PROFESSIONAL Word Processing, DISPLAYS 40 or 80 COLUMNS IN COLOR or black and white! Simple to operate, powerful text editing, complete cursor and insert/delete key controls line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers! List $99.00. **SALE $19.95. *Ltd. Qty   Closeout Item***

We pack a SPECIAL SOFTWARE DISCOUNT COUPON with every COMMODORE 64 COMPUTER, DISK DRIVE, PRINTER, or MONITOR we sell! This coupon allows you to SAVE OVER $250 OFF SALE PRICES!!

(Examples)

PROFESSIONAL SOFTWARE COMMODORE 64

Name	List	Sale	Coupon
PaperClip	$59.95	$34.95	$29.95
Consultant	$59.95	$49.95	$39.95
Leader Board	$39.95	$24.95	$22.95
The Print Shop	$44.95	$27.95	$26.95
Halley's Project	$39.95	$22.95	$19.95
Practicalc (spread sheet)	$59.95	$19.95	$14.95
Voice Command Module	$79.95	$39.95	$34.95
Nine Princes in Amber	$32.95	$24.95	$21.95
Super Bowl Sunday	$35.00	$22.95	$19.95
Flip and File Disk Filer	$24.95	$14.95	$12.95
Pro Joy Stick	$19.95	$12.95	$10.00
PartyWare	$19.95	$14.95	$11.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60
Financial Planner Sylvia Porter	$59.95	$38.95	$35.95
Hardball	$29.95	$18.95	$16.95
C64 Troubleshoot & Repair Guide	$24.95	$15.95	$12.95

(See over 100 coupon items in our catalog)

Write or call for Sample SPECIAL SOFTWARE COUPON!

ATTENTION Computer Clubs

We Offer Big Volume Discounts

CALL TODAY!

PROTECTO WARRANTY

All Protecto's products carry a minimum 90 day warranty. If anything fails within 90 days from the date of purchase, simply send your product to us via United Parcel Service prepaid. We will IMMEDIATELY send you a replacement at no charge via United Parcel Service prepaid. This warranty proves once again that ***We Love Our Customers.***

C128 COMPUTER & 1571 $499.00

Now you can get the C128 Commodore computer & the 1571 Disk Drive for one low price of only $499.00. List $698.00
SALE $499.00.

340K 1571 COMMODORE DISK DRIVE $259.00

Double Sided, Single Disk Drive for C-128 allows you to use C-128 mode plus CPM mode. 17 times faster than 1541, plus runs all 1541 formats. List $349.00. **Sale $259.00.**

SUPER AUTO DIAL MODEM $29.95

Easy to use. Just plug into your Commodore 64 computer and you're ready to transmit and receive messages. Easier to use than dialing your telephone, just push one key on your computer! Includes exclusive easy to use program for up and down loading to printer and disk drives. ***Best in U.S.A.*** List $99.00. **SALE $29.95.** *Coupon $24.95.*

VOICE SYNTHESIZER $39.95

For Commodore-64 computers. Just plug it in and you can program words and sentences, adjust volume and pitch, make talking adventure games, sound action games and customized talkies!! PLUS ($19.95 value) TEXT TO SPEECH program included FREE, just type a word and hear your computer talk — ADD SOUND TO "ZORK", SCOTT ADAMS AND OTHER ADVENTURE GAMES!! (Disk or tape.) List $89.00. **SALE $39.95**

12" SAMSUNG AMBER SCREEN MONITOR $79.95

Super High Resolution amber screen monitor. 80 columns x 24 lines, easy to read, a must for wordprocessing . Fantastic value.
List $129.00. **Sale $79.95.**
(C128 cable $19.95. C64, Atari cable $9.95)

PRINTER/TYPEWRITER COMBINATION $199.95

Superb letter quality, daisy wheel printer/typewriter combination. Two machines in one — just a flick of the switch. Extra large carriage, typewriter keyboard, automatic margin control compact,lightweight, drop in cassette ribbon! (90 day warranty) centronics parallel interface Built -in. List $349.00. **SALE $199.95.** ***(Ltd. Qty.)***

14" RGB & COMPOSITE COLOR MONITOR $239.95

Must be used to get 80 columns in color with 80 column computers (C128 - IBM - Apple). (RGB Cable $19.95) Add $14.50 shipping.
List $399.00. **SALE $239.95.**

- LOWEST PRICES • 15 DAY FREE TRIAL
- BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL

PHONE ORDERS
8 a.m. - 8 p.m. C.S.T. Weekdays
9 a.m. - 12 noon C.S.T. Saturdays

- 90 DAY FREE REPLACEMENT WARRANTY
- OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling, and insurance. Illinois residents please add 6½% sales tax. Add 20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. All orders must be in U.S. Dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.

VISA — MASTER CARD — C.O.D.   C.O.D. on phone orders only.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Famous National Brand

Commodore® • Atari® • Apple® • IBM®

We Like this Printer so much
We Bought Out the Factory.

Sale $39.95 List $199.00

Sale $39.95 List $199.00

80 Column Printer Sale

• Word Processing • Program Listings • Graphics • Quiet Operation • Upper and Lower case • All points addressable Graphics • Underline • Enhanced • Much much More

Super Print Quality

This printer was made by **Canon**® for one of the largest computer manufacturers in the world. The Big Blue printer comes ready to hook up to the serial port of the **IBM**®PC jr. Plus with low cost adapter cables you can connect the Big Blue printer to the **IBM**®PC, **IBM**®XT, **IBM**®AT, **IBM Compatibles**, **Apple**®II,IIe,IIc, **Apple Compatibles, Commodore**® 64,128, **Atari**®, plus many more.
(Also VIC 20)

• 90 Day Immediate Replacement Warranty

• 15 Day Free Trial-Satisfaction or Money Back

Now you can have a full fledged 8½'' letter size 80 column printer for less than the cost of a large box of paper. This printer uses advanced thermal technology to print upper and lower case (with true lower descenders), underline, enhanced, all points addressable graphics (works with Printshop) plus More. Print out pictures, program listings, wordprocessing pages, graphics and more. Perfect for the homeowner or student and definitely affordable. Fantastic Printer at an unbeatable price. List $199.00 **Sale $39.95**

Paper (2 Rolls) List $19.95 **Sale $9.95**

Intelligent Commodore Interface— Allows you to connect the Big Blue printer to the printer port of the Commodore 64 and 128 computer. Print Commodore graphics, use Printshop, Word processors and more... List $49.95 **Sale $24.95**

Intelligent Atari Interface— Allows you to connect the Big Blue printer to Atari computers (except 1200). Print Atari graphics, Printshop ,word processors, and more... List $49.95 **Sale $24.95**

IBM®PC,AT,XT,PC Jr. **Apple**® **RS-232 Adapter**— Adapts the Big Blue printer to be used with any RS-232 port.
List $19.95 **Sale $9.95**

AppleII,II plus,IIc,IIe,Atari 520 ST,**Laser128**, **Amiga** interface— List $24.95 **Sale $12.95**

IBM, Apple, Canon, Commodore, Atari are trademarks of International Business Machines, Apple Computer, Canon Inc, Commodore Business Machines, Atari Inc. Respectively.

Add $10.00 for shipping, handling, and insurance. Illinois residents please add 6¼% sales tax. Add 20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. All orders must be in U.S. Dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.

VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only.

PROTECTO
We Love Our Customers
22292 N. Pepper Rd., Barrington, Illinois 60010
312/382-5244 to order

Commodore Software Sale

For more information send for a free catalog today!

Phone Orders

(T) Tape, (C) Cartridge, (D) Disk

Call 8 to 8 C.S.T.-M-F

312-382-5244

Games

Access

0118	Leader Board (D)	$22.95
0120	Courses for Leader Board (D)	$12.95
6010	10th Frame (D)	$22.95
3033	Beachhead 2 (D)	$19.95
0451	Beachhead (D or T)	$17.95
0752	Raid Over Moscow (D)	$19.95

Accolade

6022	Ace of Aces (D)	$17.95
6024	Deceptor (D)	$17.95
5958	Dam Busters (D)	$17.95
5954	Fight Night (D)	$17.95
5950	Hardball (D)	$17.95
5952	Law of the West (D)	$17.95
5956	PSI 15 Trading Co. (D)	$17.95

Avalon Hill

5140	Black Thunder (D)	$14.95
0396	Super Bowl Sunday (D)	$19.95
5234	Season Disk (D)	$12.95
3576	The Masters (D)	$14.95
3574	Pebble Beach (D)	$14.95
0860	Tournament Golf (D)	$18.95
2375	Computer Title Bout (D)	$19.95

Broderbund

5986	Where is Carmen Sandiego (D)	$21.95
2905	Karateka (D)	$19.95
2903	Lode Runner (D)	$19.95
5170	Lode Runner's Rescue (D)	$19.95
3038	Champ Lode Runner (D)	$21.95

Datasoft

3025	Bruce Lee (D)	$14.95
3027	Conan (D)	$9.95
5218	The Goonies (D)	$14.95
6026	Never Ending Story (D)	$17.95
6028	Mind Pursuit (D)	$17.95
5220	Zorro (D)	$14.95

Electronic Arts

5176	Carriers at War (D)	$29.95
3830	Dr. J & Larry Bird (D)	$12.95
5180	Heart of Africa (D)	$12.95
6038	Lords of Conquest (D)	$22.95
3601	Super Boulderdash (D)	$12.95
5192	Pinball Construction (D)	$12.95
5178	Reach For Stars 2 (D)	$26.95
3840	7 Cities of Gold (D)	$12.95

Epyx

0360	Jet Combat (D)	$18.95
1556	Movie Monster (D)	$20.95
6065	Super Cycle (D)	$20.95
0382	Summer Games 2 (D)	$20.95
5210	World Karate Champ (D)	$18.95
0365	W.G. Baseball (D)	$18.95

Mindscape

3702	Bop'N Wrestle (D)	$19.95
5118	The Halley Project (D)	$19.95
5120	Indiana Jones (D)	$29.95
3690	Infiltrator (D)	$19.95
5916	Lords of Midnight (D)	$9.95
5912	Luscher Profile (D)	$19.95

Strategic Simulation Inc.

1561	Battle of Antietam (D)	$27.95
5156	Colonial Conquest (D)	$22.95
3021	Computer Baseball (D)	$22.95
6002	Gettysburg (D)	$32.95
1565	Mech Brigade (D)	$32.95
6036	Nam (D)	$22.95

Educational

American Educational Computer

2521	AEC Spelling (D)	$19.95
2499	Biology Facts (D)	$9.95
2496	French Vocabulary (D)	$9.95
2495	Spanish Vocabulary (D)	$9.95
2493	Grammar Word Use (D)	$9.95
3747	Learn to Read (D)	$19.95
3745	Phonics (D)	$19.95
2519	U.S. Geography (D)	$9.95
2520	U.S. Goverment (D)	$9.95
2498	U.S. History (D)	$9.95

Designware

5100	Algebra 1 (D)	$12.95
2518	Body Transparent (D)	$22.95
2517	European Nations (D)	$17.95
0824	Grammar Examiner (D)	$22.95
0980	Mission Algebra (D)	$22.95
5102	Remember (D)	$34.95
5105	Spelling/Reading Primer (D)	$12.95
0832	States & Traits (D)	$22.95

DLM

3806	Alien Addition (D)	$14.95
3808	Alligator Mix (D)	$14.95
3800	Alphabet Circus (D)	$14.95

Spinnaker

5300	Alphabet Zoo (C)	$4.95
5090	Delta Drawing (C)	$4.95
5301	Facemaker (C)	$4.95
5303	Kindercomp (C)	$4.95
3077	Number Tumblers (C)	$4.95
3078	Sea Spellers (C)	$4.95
3076	Up & Add 'Em (C)	$4.95

Fisher Price

3095	Alpha Build (C)	$4.95
3070	Dance Fantasy (C)	$4.95
3069	Linking Logic (C)	$4.95
3072	Memory Manor (C)	$4.95
3073	Logic Levels (C)	$4.95

Weekly Reader

2515	Stickybear ABC (D)	$14.95
5130	Stickybear Math (D)	$14.95
2512	Stickybear Numbers (D)	$14.95
5132	Stickybear Reading (D)	$14.95
2516	Stickybear Shapes (D)	$14.95
5129	Stickybear Typing (D)	$14.95

Miscellaneous

Batteries Included

6060	Home Inventory (D)	$8.95
6062	Address Book (D)	$8.95
6064	Mail List (D)	$8.95
6066	Checkbook (D)	$8.95
6068	Recipes (D)	$8.95
6018	Keys to Typing (D)	$9.95
6020	Cal-Kit (D)	$9.95
1253	PaperClip (D)	$34.95
1254	Consultant (D)	$34.95

Software Country

3633	Chessmaster 2000 (D)	$26.95
3632	Golden Oldies (D)	$17.95

Business

C128 Software From Timeworks

5022	Word Writer (D)	$59.95
5024	Data Manager 2 (D)	$39.95
3048	Sylvia Porter (D)	$39.95
5030	Partner (C)	$39.95
5026	SwiftCalc/Sideways (D)	$39.95

Add $10.00 for shipping, handling, and insurance. Illinois residents please add 6½% sales tax. Add 20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. All orders must be in U.S. Dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.

VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Famous Comstar National Brand

10" Printer Sale

Includes Commodore Interface

Near Letter Quality *Near Letter Quality*

Best Value in the USA

• 100 CPS draft/20CPS near-letter quality • Dot Addressable Graphics • Adjustable Tractor and Friction Feed • Automatic Paper Loading • Right and Left Margin settings • Pica, Elite, Condensed, Italics • Superscript • Subscript • Underline, Bold print, Double Strike • Superb NEAR LETTER QUALITY

$179^95

$179^95

Easy to Use

Fantastic Graphics

2 Year Warranty

The Comstar 1000 is one of the best values in the United States today. Print your letters, documents, programs, pictures, and more at a blazing 100 Characters Per Second or 20 cps in the Near Letter quality mode. (Looks just like it came from a typewriter.) Plus, choose your printing mode (NLQ, Draft) from your wordprocessor or by simply pushing the controls on the front panel. Fantastic Quality at a Fantastic Price.List $349.00 **SALE$179.95.**

Print Method
Serial impact dot matrix (9 pin)
Print Speed
Draft- 100 CPS NLQ- 20 CPS
Character Sets
96 ASCII Characters, Marker, Symbols
(includes italic font)

Ribbon (Life exp.)
Black: cassette (2.5 million characters)
Dimensions
15.4 (W) x 10.9 (D) x 4.7 (H) inch
Weight
Approx. 10 lbs
Character Spacing
Fixed

Line Spacing
1/6, 1/8, 7/72, and 1/216 inch
Paper Feed
Adjustable tractor and friction feed
Paper feeding Direction
Bi-directional
Copies
2 plus original

Supply is Limited so Buy Today

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

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO
We Love Our Customers
22292 N. Pepper Rd., Barrington, Illinois 60010
312/382-5244 to order

In putting the FSD-1 through its paces, I daisychained the drive with a 1571 and my Commodore 128. The drive operated perfectly. Since most of the 128's software is still on single-sided disks and will probably continue to be for the foreseeable future, 128 owners should seriously consider the FSD-1 as an inexpensive alternative to the 1571, especially as a second disk drive.

Frankly, I have yet to find fault with the drive. It works well with Epyx's Fast Load cartridge and Access's Mach 128 cartridge. The drive runs smoothly, efficiently, and without heat buildup. It runs everything that the 1541 will, and easily daisychains with other drives. The FSD-1 is one of the best of the new hardware offerings to come out for use with the 64 and the 128. Considering all of its features and the price, I highly recommend the FSD-1 to 64 and 128 users.

—Scott Thomas

Emerald Components International
Dept. DD414
540 Willamette St.
Eugene, OR 97401
$139

GATO

The message from SUBCOM crackles over the radio: "Pilot down! Must rescue. Coordinates 5100, 2100! Repeat: Must rescue!"

You issue the command: "All ahead flank speed," and turn to a bearing to take you to the given coordinates.

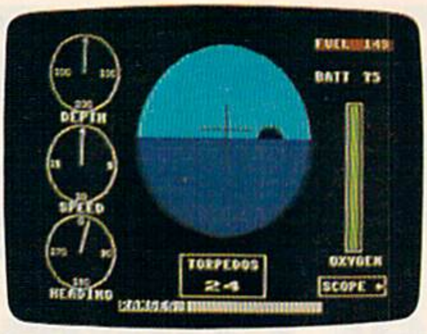
As commander of a GATO-class World War II submarine, you have the responsibility of rescuing the downed pilot. You have 150,000 pounds of fuel, a chart of the area, radar, and 24 torpedoes. But your most effective weapon is surprise.

As a precaution, you elect to submerge as you approach the area. You raise the periscope for a quick scan of the area.

It's a trap! The enemy, not SUBCOM, has summoned you here and you're surrounded by warships. Even as you fire your first fish, their first salvo destroys your radio room....

Spectrum Holobyte's *GATO* for the Commodore 64 is more of a game than a realistic simulation. The emphasis is on *fun*. It's the arcade come home, and time spent letting loose a few torpedoes can let loose some steam as well.

In *GATO*, you are the commander. Missions are assigned by a direct radio transmission from SUBCOM (SUBmarine COMmander). You must carry out the missions while avoiding enemy attack. This involves keeping track of depth, speed, heading, diesel and battery power, oxygen consumption, and enemy movements. The opposition is both smart and aggressive. If—or *when*—your sub is detected, enemy forces will attempt to destroy it.



You control the sub via one- or two-stroke keyboard commands—no joystick is required. Six screens are provided to help you command your vessel.

The Control screen contains the control panel and periscope view. From this you control speed, depth, heading, periscope, power use (diesel/battery), and torpedoes. You also monitor oxygen level and enemy range.

Navigational aids include a chart of the sub's patrol area and a simulated radar display. The chart diagrams an aerial view of the waters. The radar screen is much like an actual one, tracking enemy ships and noting island locations relative to your sub.

In any encounter with the enemy, you're likely to sustain some damage. A Damage Report screen indicates what areas of the sub have been affected. This is done either by list or by filling in sections on a cut-away view of the sub.

The Ship's Log displays the sub's name, the number of missions completed, the number of enemy ships sunk, and the total tonnage sunk. This log is automatic, so that information from each play session is added to that of the previous session. The log can be erased and a new one started if desired.

The Parameters screen allows you to switch levels of difficulty (1–5) and to change the submarine's coordinates.

The manual is detailed and extremely well written, providing play instructions as well as a history of submarines. In addition, there is a removable easy-to-use reference card which saves flipping through the manual during play. Because of the manual, as well as the simplified nature of the simulation, *GATO* is easy to play. Not easy to win, just easy to play.

GATO's lesser sophistication means it has some limitations. There are only four directions in which you can look, either with the raised periscope or from the bridge. There is no indicator of sea-bottom contours or the depth of the water you're in. Although the more dramatic sounds of submarine operation are built into the program, there is no background engine noise. No sound is provided for running aground either. The graphics for the ships and islands are somewhat crude. The ships look like something made out of blocks rather than finely-defined images. The islands, also artistically unspectacular, all look alike.

On the plus side, however, are several features. The radio messages are excellent. There's a realistic "fuzzy" sound built into the transmission that makes you halfway expect to hear "Tokyo Rose." The "aauuugha" of the diving horn and the "ping" of the enemy sonar are reminiscent of late night submarine movies. The explosions are done so well you can almost feel them. The sub responses are most appropriate—quick at high speeds, sluggish at low. Uncharted sandbars and dud torpedoes are thrown in just to make things more difficult—and realistic.

The final plus is how easy it is to learn. One evening at the controls and you're well into the challenge of matching your wits against those of the enemy's navy. Still, you must remember, World War II wasn't won—or lost—in a day. It took many battles—and the challenge of many battles is what awaits you in *GATO*.

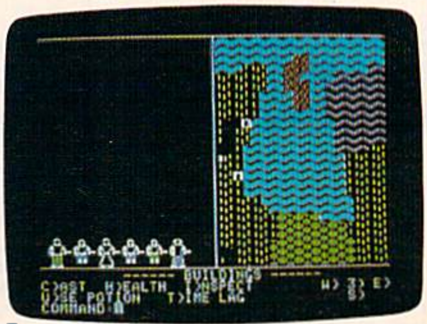
—David and Robin Minnick

Spectrum Holobyte
1050 Walnut, Suite 325
Boulder, CO 80302
$29.95

Phantasie II

A few years ago, there were few exceptionally good computer games. While this was unpleasant for the hobbyist who wanted more high-quality entertainment packages, it insured that the well-done games would be played and replayed until they were mastered.

Things are different now. It seems safe to say that there are more computer games than most people can possibly find time to play, and many of them are good. What separates the games that



are replayed from those that are not is addictiveness.

Addictiveness, in games at least, is a personal thing. What addicts one person does not addict another. But we all know what it means—sitting up until all hours of the night, trying one more time to beat your high score, or to finally defeat the computer, or to complete the long, arduous quest, or to master the intricacies of a coordinated attack. It means waking up in the morning with a sudden inspiration about what to do next, and sitting at the computer for hours at a time working at the game instead of doing the word processing you ought to be doing. It happens, and not always when you would expect it.

When I first examined SSI's *Phantasie*, the forerunner of the Commodore 64 game reviewed here, I was largely unimpressed. The graphics were functional but hardly dazzling, disk access (on my 64) was frequent and annoying, and game play was predictable. I persisted for a while, but my old standards drew me away. When the sequel, *Phantasie II*, came along, I expected more of the same, but, I thought, perhaps there were enough changes to make it more appealing.

I don't know if there are any significant changes from the first game, but there has been a big one in my reaction. I am addicted to *Phantasie II*. I'm not quite sure why, since graphically it's about the same as the original, and game play is virtually identical. But I am, nonetheless. I don't even particularly like fantasy adventure games that demand killing monsters and getting treasure (which this game certainly does). Nor do I like waiting for my Commodore disk drive to reveal another part of *Phantasie II*'s world to me. But I do like this game. Too much, maybe, for my own good and that of my family.

For a game to be addictive, it must possess several qualities. First, it must present a purpose. High score is one purpose, and most pure arcade games provide this. Winning the game is another, and this purpose is provided by all sports games, games with a time limit, and strategy games like chess and war games. Adventure games, however, depend on fulfilling the quest, the third type of purpose.

Equally important, though, is the ability to achieve the game's purpose. A game that's nearly impossible to crack will fail to addict all but the most dedicated gamer. An addictive game will allow the player to progress. Maybe the computer opponent is too tough at first, but if you can come close after a handful of games, you'll continue to try.

In *Phantasie II*, the quest is unknown (except through hints provided in the documentation) until the player

makes some progress. What keeps the player going, though, is the promise of progress. A party of adventurers begins in a town, then leaves town for wilderness. There they encounter monsters of various shapes, sizes, and fighting abilities. In most cases they will fight to the death. Some beginning adventurers will die, but some will survive to return to the town and receive training, new spells, new weapons, and other forms of aid. Many games fail in one essential way: Progressing, and staying alive, is nearly impossible. In *Phantasie II*, though, as long as the player is content with adventuring for a short while, then returning to town, the game is relatively easy. Even beginning adventurers can survive long enough to improve.

And improvement is fairly rapid. Before long, if you do things competently, your party will be at the eighth level or above, and you can start doing some strong adventuring. Three levels of the Netherworld, and several beasts that serve Pluto and have special powers, await an advanced party. Other intriguing elements include the granting of divine spells—their uses unknown—to party members. And an amulet that teaches the party about beast runes, but not how to use them. And lots of other things.

The quest itself, as in most of these games, is almost secondary. Getting there is definitely more than half the fun, and the game requires that you keep your wits about you. But it's primarily a fighting game, and a good one at that. The monsters are many (annoyingly so at times), and some are deadly enough to strike fear into the player. The world is large enough to take hours to get through, and the magic spells are potent enough to make nonfighters consider investing in them. I wish there was more interaction and less emphasis on killing monsters and getting treasure, but the game pretends to be nothing else. If you like hack-and-slash fantasy adventuring, *Phantasie II* should serve you well.

One hint, and only one. Listen to the crazy gnome that runs around the first dungeon. He's a bit machinelike. And the princess is a bit unpredictable. And...sorry—my addiction, you know.

—*Neil Randall*

Strategic Simulations
1046 N. Rengstorff Ave.
Mountain View, CA 94043
$39.95

The Keeper

I've always dreamed of turning on my Commodore 64 and having the DOS Wedge ready at my command, or maybe a machine language monitor, or even a word processor, utility, or game. I've wanted to show off my own programs by having them placed on a cartridge so they can start up immediately. Now I can. Thanks to the Keeper from Marvco Electronics, I can store just about anything I want in the cartridge.

The Keeper cartridge is a very exciting and practical device with three important capabilities: It adds 8K of nonvolatile RAM memory to your computer (a 16K version is also being develòped); it can be programmed, and it offers autostart capability; and it has switch-selectable options that include RAM/OUT/ROM modes.

What will the Keeper do, and how can these capabilities be put to practical use? In my experience, I've found that it does more than its enclosed literature and advertisements suggest. The cartridge takes up the space at the top of BASIC memory—from 32768–40959 ($8000–$9FFF)—as do most cartridges, and will not interfere with BASIC programs. A slide switch on the Keeper allows it to be totally switched out from the computer's view, and helps save wear and tear on the expansion port. The slide switch can also select the RAM or ROM modes. In RAM mode position, the Keeper's memory can easily be changed, and is treated like typical random access computer memory. In the ROM position, the Keeper's memory is read-only, and can't be changed even if you turn the computer off. The Keeper's secret is that it contains a battery that preserves its memory after you turn off the computer.

The Keeper comes with an easy-to-use menu-driven program that's written in BASIC. This program guides you through the operation of the Keeper and controls the storing of program information to cartridge. There is also a special KEEP command which may be used to store programs in Keeper's memory in lieu of using the menu program. This command allows the Keeper cartridge to perform like an instant disk drive.

One of the most interesting capabilities of this cartridge is its autostart ability. With it, any program stored in the Keeper will run when the computer is powered up. I've used the Keeper to store many programs, including the DOS Wedge, a machine language monitor, *SpeedScript*, various utilities, and my own programs.

In a search of similar hardware on the market, I noticed a couple of EPROM (Erasable Programmable Read

Only Memory) systems—but these lack the flexibility and convenience of the Keeper. For an equivalent EPROM system you'd need an EPROM chip, an EPROM programmer, an EPROM eraser, and a housing cartridge to plug the chip into the computer. The combined cost of these items would be much greater than the cost of the self-contained Keeper. If you like the idea of an autostart cartridge that you can program yourself and that has the features described here, you'll like the Keeper.

—Marvin Green

Marvco Electronics
15702 S. Neibur Rd.
Oregon City, OR 97045
$39.95 (plus $3 shipping and handling)

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.

Gemini II Printer

Star Micronics made a name for itself a few years ago with Commodore 64 owners when it introduced the Gemini 10X printer. The 10X, based on price and performance, went on to become one of the most popular printer choices for Commodore owners.

At the recent Summer Consumer Electronics Show, Star Micronics premiered the Gemini II, a Commodore-specific dot-matrix printer that combines the most popular features of the earlier model with many ease-of-use features found on more expensive business printers. With this new addition to its product line, Star has mirrored the industry's move toward producing both hardware and software that can be operated easily by new computer owners. Of particular note are the Gemini II's paper-handling, switch-selectable modes, and quality and variety of type styles.

As with most other printers that come with cables designed for a specific computer, setup is simple and fast. You just plug one end of the specially designed Commodore serial cable into the printer and the other end into the back of the disk drive. The printer cover and paper holder snap easily into place, and are just as easily removed when you want to change from continuous paper feed to single sheet feed.

You can't really appreciate a printer with a slick, well-designed paper feed system unless you've used a poorly designed one. Even if you plan to use your printer only for printing out program listings or rough copies of correspondence, a reliable tractor feed that won't lose its grip on the paper every third or fourth sheet or get stuck and refuse to feed the paper to the next line is essential. The real challenge comes for the user who needs to switch frequently between continuous form-feed paper and single sheets or stationery. The Gemini II handles this operation admirably. You set up the snap-on paper holder that separates the incoming and outgoing streams of form-feed paper at about a 75-degree angle, rest your piece of paper against it, move the paper-control lever, and your paper is automatically fed around the platen and into position. Switching back to form-feed operation requires you to move the tractors back into the right position and refeed the paper into the tractors. The process may seem a bit clunky and time-consuming at first—having to keep moving the tractors and paper separator—but it's *reliable*.

A second area in which the Gemini II excels is mode selection. Not only can you easily switch back and forth between draft mode and NLQ (Near Letter Quality) mode by pressing buttons on the front panel, but you can also choose among three different pitches (characters per line) within draft mode in the same way: 80 characters per line (pica), 96 cpl (elite), and 136 cpl (condensed). Turning boldface on and off is

```
This is a sample of text Printed by the Gemini II at 80 cPl.

This is a samPle of text Printed by the Gemini II at 96 cPl.

This is a samPle of text Printed by the Gemini II at 136 cPl.

This is a sample of text printed by the Gemini II in NLQ mode.
```

accomplished the same way. You simply press a button on the control panel. It's unusual to see a printer for the home market in this price range that offers pushbutton control of functions that are often controlled only through software or by adjusting the DIP switches.

The Gemini II is Commodore 1525-compatible with some extra features so that things like underlining, italics, superscript, and subscript are controllable through software. For the novice who has never worked with printer formatting commands, the accompanying documentation offers clear, thorough instructions on how to work with different type variations.

As far as the noise level and speed of the Gemini II go, they're both about average for a printer in this price range. Though the noise is muffled somewhat by the printer cover, long printing jobs may be a bit disturbing to anyone within earshot. But that's a problem with any impact printer; when a metal object is striking something else with the regularity of a printer, there's bound to be some noise. The Gemini II runs at 120 cps (characters per second) in draft mode and 30 in NLQ mode. That can seem pretty slow sometimes when you have several printing jobs in a row, but it's the norm for printers in this general price range.

A manufacturer's reputation for quality products is extremely important to consider when you're shopping for a printer. Star Micronics has a good name. Should you have technical difficulty in setting up or operating your printer (like disk drive lockup when the printer is on), remember that the Commodore 64 has gone through several ROM changes in its four-year life. Operational problems may be the fault of your computer or disk drive, not the printer itself. Should something come up that you think is a problem with the printer, Star operates a toll-free technical assistance line—(800) 367-3119.

—Kathy Yakal

Star Micronics
200 Park Ave., Suite 3510
New York, NY 10166
$329 (includes Commodore interface)

ESCaping With The Commodore 128

Jim Vaughan

The ESC key makes the Commodore 128 an even more powerful machine. With it, 27 new screen-editing features can be accessed. This article discusses each of the ESC key sequences and how to use them in your own programs. Two helpful demonstration programs are also included.

The Commodore 128 offers a variety of editing and screen control features which are accessed using the ESC (escape) key, located on the far left of the upper row of gray keys. These features—new to Commodore machines—make Commodore's popular full-screen editing even better.

Each of the 27 features is accessed by first pressing the ESC key, releasing it, and then pressing another key—either a letter from A through Z or the @ (at) symbol. Unlike control characters such as CONTROL-9 ({RVS}) where you hold down one key (CONTROL, Commodore, or SHIFT) while pressing another, you *should not* hold down the ESC key. You press the ESC key once and then press the other key. The two keystrokes together form an *escape sequence*. The escape sequence functions can also be used within programs by printing the character code for ESC (CHR$(27)), followed by the appropriate letter for the sequence you wish to use. (It's purely coincidental, but there are 27 new editing sequences for ESC, which is CHR$(27)). The sequences function identically with shifted and unshifted letters—ESC SHIFT-A does the same thing as ESC A—so the same techniques work if SHIFT LOCK or CAPS LOCK is pressed. (The exception is ESC @, since ESC SHIFT-@ does nothing.)

The new set of commands can be broken down into three categories: editing enhancements, screen control enhancements, and miscellaneous.

Editing Enhancements

The new editing commands allow quicker movement around the screen and easier entry or deletion of part or all of a program line. These sequences are summarized in Table 1.

The first sequence, ESC A, puts you into auto-insert mode. This allows you to enter text or program lines at the current position of the cursor without writing over the rest of the line. Auto-insert mode is most useful when you need to go to the middle of a program line and add statements. ESC C cancels the insert mode. (The Commodore 128 *System Guide* that comes with the computer erroneously states that this sequence cancels quote mode.) Once you're finished inserting text, it's advisable to cancel the mode with ESC C, since printing is noticeably slower when the computer's in auto-insert mode. Note that auto-insert mode is different from the other kind of insert mode which appears when you hold down SHIFT and press the INST/DEL key. With ESC A, you don't have to open up the space first; also, auto-insert mode doesn't act like quote mode. Control characters such as RVS ON won't print in reverse video as they do in quote mode or insert mode.

ESC D allows you to delete the program line the cursor is currently on. (To delete just part of a line, see ESC P and ESC Q below.) Be warned, however, that if the program line extends over more than one physical screen line, this will delete the entire logical line—not just the single screen row the cursor is on.

ESC I allows you to insert a blank line at the current cursor position. This sequence will move down one row all text on and below the line containing the cursor. Anything on the bottom row will be scrolled off the screen, and the cursor will remain on the new blank line. If you insert a line in the middle of a multirow program line, the text on the rows following the new line will still be considered part of the program line. Remember to put the cursor on the screen row where you want the new line to be added before you issue this sequence.

Table 1: Editing Enhancements

ESC A	Enable auto-insert mode
ESC C	Cancel auto-insert mode
ESC D	Delete the current logical line
ESC I	Insert a blank line
ESC J	Move to the beginning of the current line
ESC K	Move to the end of the current line
ESC O	Cancel quote mode (ESC ESC also works)
ESC P	Erase from beginning of line to cursor
ESC Q	Erase from cursor to end of line
ESC @	Erase from cursor to end of screen

Table 2: Screen Control Enhancements

ESC B	Set bottom right corner of output window
ESC L	Allow scrolling
ESC M	Disable scrolling
ESC T	Set top left corner of output window
ESC V	Scroll screen up one line
ESC W	Scroll screen down one line
ESC X	Switch between 40- and 80-column modes

Table 3: Miscellaneous Sequences

ESC E	Set cursor to nonblinking mode
ESC F	Set cursor to flashing mode
ESC G	Turn on (enable) bell tone
ESC H	Turn off (disable) bell tone
ESC N	Return to normal 80-column display
ESC R	Reverse 80-column character and background colors
ESC S	Change 80-column cursor to block
ESC U	Change 80-column cursor to underline
ESC Y	Restore default tab stops
ESC Z	Clear all tab stops

ESC J and ESC K allow quick movement on the program line the cursor is on. ESC J moves the cursor to the beginning of the program line, and ESC K moves the cursor to the position just beyond the last nonspace character in the program line. (Again, these commands work with logical program lines, which may consist of more than one physical screen row.) I find these sequences the most useful when editing, since I'm always adding statements onto the end of program lines. Using ESC K saves me from having to hold down the cursor-right key and wait until the cursor reaches the end of the line.

ESC O cancels quote mode. Probably every Commodore programmer has tried editing a portion of a program line between quotation marks and ended up with a collection of reversed characters. The operating system treats editing characters like {DOWN} and {RVS} differently when they are typed within quotation marks. Instead of acting immediately, they appear as reverse characters. This is fine if you're trying to create a string to print cursor-right characters, but annoying if you want to use the cursor-right key to move to the end of the string. By using ESC O, you can cancel quote mode to use the normal editing keys without getting the reverse characters. Insert mode (which is almost identical to quote mode) is in effect when you insert spaces with the INST/DEL key. For example, if you insert five spaces, those five character positions behave as if they were within quotes. ESC O cancels this effect as well. It's not documented in the manual, but pressing ESC twice in a row (ESC ESC) is the same as ESC O—a handy shortcut. (ESC O is erroneously defined in the *System Guide* as cancelling auto-insert mode. The manual has ESC C and ESC O functions reversed.)

ESC P will erase everything from the beginning of a program line to the position of the cursor. That means if you put the cursor in the middle of a program line and then press ESC P, the first half of your line, including the line number, will be erased. ESC Q is the complement of ESC P, meaning that it will erase everything from the cursor to the end of the line. Erased positions are filled with spaces; the cursor and remaining text do not move. As before, it's important to remember that these sequences affect logical lines, not just screen rows. If you press ESC P when the cursor is in the middle of the third row of a program line that spans four screen rows, 2-1/2 rows will be erased—not just the row on which the cursor resides.

ESC @ is an enhanced version of ESC Q. Instead of erasing to the end of the current line, it erases everything from the cursor position to the bottom right corner of the current output window. This can be useful when you're loading a program from the directory. Press f3 to list the disk directory, move the cursor to the program you want to load, and type DLOAD (or RUN) to the left of the filename. Now press TAB two or three times, until the cursor is past the name of the program. Finally, type ESC @ to clear the bottom portion of the screen and press RETURN to load the program.

Screen Enhancements

The following group of sequences manipulate the entire screen area as opposed to just editing lines. These sequences are summarized in Table 2. It's not really proper to refer to the screen when talking about the 128's display, since all screen operations actually depend on the height and width of the currently defined output window. It's easy to forget this; the output window is most often set to cover the entire screen, but that doesn't have to be the case. The 128's windowing capabilities aren't as powerful as those of a Macintosh or Amiga—you have only one window to work with—but they do allow you great flexibility in designing screen displays.

ESC B and ESC T can be used to change the boundaries of the output window. ESC T defines the current cursor position as the top left corner of the window—the home position of the output display, where the cursor will go when the CLR/HOME key is pressed. ESC B defines the current cursor position as the bottom right corner of the window. Just move the cursor to the desired position and use the sequence to set the new boundary. You probably wouldn't use these escape sequences in a program (although you could) because the WINDOW command is more convenient to use.

There are a lot of creative ways to employ windows when you're programming. For example, you may decide you want to look at a disk directory. Just create a window and press f3. Whatever is outside the window will remain unchanged. You could also compare two sections of a program by listing the first part, setting up a window, and listing the second part. (See "Windows on the 128," COMPUTE!'s GAZETTE, April 1986, for more about windows.)

The window isn't cleared when it is resized, but all lines are unlinked (the text in each row is "disconnected" from that in any previous or following rows). The sequences are easy to remember: ESC T for Top and ESC B for Bottom. Pressing CLR/HOME twice in a row (HOME HOME) will reset the window to full screen size.

ESC M and ESC L control how text is handled when there's no more room at the bottom of the window. Normally, all lines of text in the window scroll upwards to add the text to a new line at the bottom of the window (the previous top line is scrolled off the screen). ESC M turns off the scrolling. When scrolling is disabled and the screen is full, the next PRINT statement will cause the text to be displayed at the top of the screen, *unless the PRINT statement is followed by a semicolon (;)*. When you LIST a program or directory which is longer than the current height of the window, printing automatically jumps back to the top of the window after a line is displayed at the bottom. However, things get a bit strange when the PRINT statement ends with a semicolon. Press ESC M; then try the following line:

```
FOR I=1 TO 5000: PRINT
  CHR$(INT(RND(1)*64+32));: NEXT
```

When the bottom line of the window fills up, the printing will continue in the position at the lower right corner of the screen, with each successive PRINT overwriting the previous character in that position. This mode is useful when you're trying to display some graphics effects that you don't want to scroll off the screen, but you must plan your PRINT statements carefully when scrolling is disabled. Use ESC L to turn the normal scrolling feature back on.

ESC V and ESC W move all lines of text in the window up (ESC V) or down (ESC W) one row, adding a new blank line at the bottom or top of the screen. Any text that is scrolled off the top or bottom of the window will be lost—it's not recovered when the window is scrolled back in the opposite direction. Scrolling down is an interesting effect which could be exploited in an auto-racing game. You'd put the cursor on the top line, print the two edges of the road, scroll it down, and repeat. It would be relatively easy to add a sprite for the automobile, a joystick movement routine, and a routine that subtracts points when the car hits the side of the road (using the COLLISION command).

The 128 provides both 40- and 80-column displays, but only one can be active at any given time. That is, only one display can have a "live" cursor, and all printing will be sent to that display. The ESC X sequence allows you to switch between the two display modes. This sequence is a toggle: Whenever you use ESC X, the display that is currently active becomes inactive and the one which was previously inactive becomes the active display. This sequence doesn't actually turn either video chip on or off—whatever is currently displayed on the screen which becomes inactive will remain intact until the screen becomes active again (or until the reset button or RUN/STOP–RESTORE is pressed). If you are using the same monitor for both displays, you still must switch the monitor to the desired display. The 40/80 DISPLAY key has no effect on ESC X (other than to determine which display will be active after reset or RUN/STOP–RESTORE).

Miscellaneous Sequences

This last set of sequences is called the miscellaneous group simply because they don't fall into either of the above two groups. These sequences are summarized in Table 3.

The 128 allows you to customize the cursor to your own tastes. Some people find the blinking cursor irritating, so ESC E stops cursor flashing. The cursor remains visible—it's now a nonblinking block. ESC F will make it flash again. Those who use the 80-column screen have an additional choice. ESC U changes the 80-column display's cursor from a character-size block into a character-width underline (the 40-column cursor is unaffected). ESC E and ESC F can still be used to make the underline cursor nonblinking or flashing, respectively. ESC S changes the cursor back to a block.

The 128 has a feature that is common on most other computers and terminals, but which had been missing on the VIC-20 and Commodore 64: a BELL character. Printing CHR$(7)—or holding down the CONTROL key and pressing G—causes a bell-like tone if your monitor is equipped with a speaker. This is a handy way to generate an attention-getting sound, but some programs may make excessive use of the feature. Maybe you just find the noise annoying. Whatever your reason for using it, ESC H disables the tone. Use ESC G to turn it back on.

The background color of the display is normally determined by the value in a video chip register, and the color of each character is normally determined independently by the value in a color memory location corresponding to the character's position on the display. On the 80-column display, this can be reversed. After you issue the ESC R sequence, all characters take what was previously the background color, and the background for each character is determined by the color value in the corresponding color memory position. (The 40-column display is unaffected.) Use ESC N to return the display to normal. ESC R is useful when you have the 80-column output displayed on a monochrome monitor and want dark characters on a light background.

The two final sequences deal with tab stop settings. Whenever you press the TAB key (located next to the ESC key on the upper row of keys) or print the tab character (CHR$(9)), the cursor moves to the next column for which a tab stop has been set or—if there are no more tab stops—to the right boundary of the output window. This is useful for aligning columns of text and numbers. When the 128 is turned on or reset, it establishes a tab stop every eight spaces. ESC Z erases all tab stops so the cursor always goes to the right margin of the window when you press TAB. You can change the tab stop setting of the column where the cursor currently resides by pressing SHIFT-TAB (or CONTROL–TAB). This is a toggle: If no tab stop has been previously set at the column, one will be set there when you press SHIFT-TAB or CONTROL–TAB. However, if a tab stop was already set for the column, it will be cleared. You can get the same effect within a program by printing CHR$(24). ESC Y resets the default tab stops (one every eight spaces). Changing tab settings—including using ESC Y and ESC Z—affects only the active display. Each display has its own tab stop table, so it's perfectly possible to have default tab settings for the 80-column display while having tab stops every four spaces, or no tab stops at all, for the 40-column display.

Using The Sequences From Within A Program

As noted earlier, even though these new ESC sequences are useful for program editing and for onscreen fun, they can also be used within a BASIC or machine language program to achieve the same results. Some of the sequences may not have an obvious effect while the program is running. For example, since the cursor is turned off when a program is running, changing the cursor mode won't have any immediately visible effect except in INPUT commands. Program 1 is a short demonstration of how a few of these ESC codes can be used in a BASIC program. Type in the program carefully, since it contains a lot of cursor control codes inside quotation marks. To use these sequences in a machine language program, print the ESC code ($1B) using the BSOUT routine ($FFD2); then print the ASCII code for the desired function. For example, the following code would scroll the display up one line:

```
LDA  #$1B
JSR  $FFD2
LDA  #$56
JSR  $FFD2
```

A handy way to use these sequences is to include them in programmable function key definitions. You can use the line

```
KEY 1, CHR$(27)+"I"
```

and a blank line of text will be inserted whenever you press f1. If you want to have the editing definitions loaded automatically whenever the computer is turned on or reset, you can write a key definition and make it an autobooting disk file. Program 2 was written for just that purpose. It assigns an editing function to each of the eight function keys and prints a menu at the bottom of the display for easy reference. The output window is then reduced by two lines so that the menu won't be overwritten. To make the progam autobooting, use the "Autoboot Maker" program on the 1571 *Test/Demo* disk that came with your drive, or use a program like "128 Autoboot" (COMPUTE's GAZETTE, March 1986).

The key definitions included in Program 2 may not be the ones that you wish to use most often, but it's easy to change any of the definitions to ones that suit you. The definitions currently included in Program 2 are as follows:

f1	Enable auto-insert mode
f2	Cancel auto-insert mode
f3	Insert a blank line
f4	Delete a program line
f5	Move cursor to start of line
f6	Move cursor to end of line
f7	Delete to start of line
f8	Delete to end of line

Once you start to learn the ESC sequences, you'll find yourself using them more and more frequently.

See program listings on page 117.

Menu System

Don J. Reynolds

With this useful program, you can easily create your own disk directory menus with descriptive filenames. Once you've created the custom menu, all it takes is one keypress to load and run a BASIC or machine language program. For the Commodore 64 and Plus/4.

"Menu System" asks a series of questions and then goes to work for you. And it's very easy to use.

You begin by entering information about files on your disk. Menu System then writes a custom directory menu. This menu remembers and executes the proper load extension (,8 or ,8,1), and it remembers the SYS number or the RUN command if needed. If programs are added or removed from the disk at a later date, the customized menu can be loaded back into Menu System to be edited.

Making A Menu

Menu System is written entirely in BASIC. It works as listed for the Commodore 64. If you have a Plus/4, add lines 1 and 5 and substitute lines 190, 620, and 630 as shown below:

```
KE 1 COLOR Ø,7,4:COLOR 4,7,4
RC 5 KEY1,CHR$(133):KEY2,CHR$(
     137):KEY3,CHR$(134):KEY4,
     CHR$(138):KEY5,CHR$(135)
XB 19Ø S8$="7Ø FORZ=ØTOX:B=131
       9+Z:POKEB,13:POKE239,Z:
       NEXT:PRINT"+CHR$(34)+"
       {HOME}"
HB 62Ø PRINT"SAVE"+CHR$(34)+"@
       Ø:MENU"+CHR$(34)+",8:SY
       S65526{HOME}"
GP 63Ø FORZ=ØTO1Ø:POKE1319+Z,1
       3:POKE239,Z+1:NEXT
```

After typing it in, be sure to save a copy. To use the program, load it and type RUN. The main menu will appear, and you'll see a list of five available functions. To create a new menu, press f1. The editing screen is displayed, with 20 blank program slots. Follow the prompts on the screen:

1. Type E to edit.
2. Enter the slot number you want to edit.
3. Enter the program name as it should appear on the custom menu.
4. Enter the boot name, the actual name of the file in the directory.
5. Enter the load extension:
   a) 0 to load ,8 (BASIC programs)
   b) 1 to load ,8,1 (machine language programs)
6. Enter the SYS number or press RETURN.
7. Enter Y if a RUN command is needed, N if it is not.

Continue entering data until you've covered all programs on the disk, up to a maximum of 20 programs. Then press RETURN to display the main menu again. Press f4 to save the menu file to disk. You'll be prompted to insert the disk. A sequential file named ".M" that contains all the information for your custom menu will then be saved to your disk.

To verify that the file was saved correctly, press f1. This erases the file in memory and displays the editing screen again. Press RETURN to get to the main menu. Press f2 to load the menu file from disk. At this point you can edit the menu and resave it, or erase it to create more menus.

The last step is to write the menu driver program MENU to disk by pressing f5. The program driver will be written to your disk and a cold start will take you back to BASIC. The dynamic keyboard technique (lines 600–630) is used to generate the MENU file, and Menu System is erased from memory in the process. If you delete the SYS 64738 (64 version) or SYS 65526 (Plus/4 version) in line 620, the program will end with the menu program in memory. You can then run it or save it to other disks.

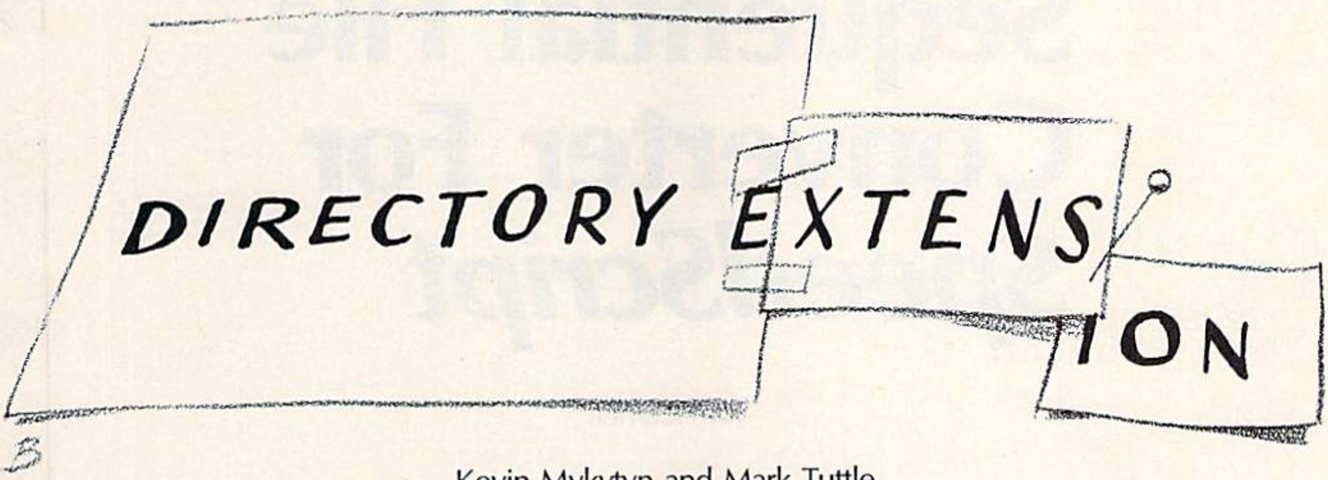
The MENU file is identical on all disks. It reads in the file .M, which is unique to each disk. You must always edit and save .M with the Menu System program. Once you've generated a copy of MENU, it can be copied to the other disks by any conventional means.

Using Your New Menu

Type LOAD "MENU",8 and then type RUN. The program displays your custom menu on the screen. Enter the number corresponding to the program you want to load. The program loads and automatically executes any other commands you have requested, such as SYS 49152 or RUN.

Editor's Note: While you are using this program, we suggest having a printout of your disk directory on hand for entering information. Or you can view your directory—with Menu System in memory—using "MetaBASIC" or with the "Wedge" program on the disk that came with your drive.

See program listing on page 111.



Kevin Mykytyn and Mark Tuttle

Have you ever looked at a cryptic filename like "ZYM/414/LUB" and wondered what it was and where it came from? Have you ever forgotten the SYS for a favorite program? This program will help; it adds up to 55 characters per directory entry—plenty of room for descriptions and information. For the Commodore 64.

Commodore disk drives limit filenames to 16 characters, which sometimes just isn't enough. "Directory Extension" allows you to add a 55-character extension to any or all files on a disk. After creating the descriptions, simply save them on the disk they describe. Then, whenever you want to see the descriptions, load and run Directory Extension. Also, you can update or modify this file as often as you like.

Easy Operation

Directory Extension is written entirely in BASIC. Type it in and save a copy. To use it, type RUN. The program first looks for the description file for the disk in the drive. If one exists, it's loaded into memory. Next, the directory of the disk is read and you're presented with a menu.

The menu offers five options. Option 1 allows you to view the directory and enter or edit the descriptions. To the left is the file number. In inverse black is the filename, and the inverse white area is the place where you enter the descriptions. At the bottom of each screen are five additional options, each chosen when you press the appropriate function key. If the disk contains more than six files, use the f5 and f7 keys to page forward and backward, respectively. The f3 key allows you to edit the descriptions.

After pressing f3 to select a file, enter the file number. The cursor moves to the beginning of the white field next to the file you have chosen. (If you mistakenly choose the wrong file, press RETURN and all will remain intact.) Now you may type up to 55 characters. Any character is allowed except quotation marks. The cursor automatically wraps around at the end of a line. Use the INST/DEL key to erase. When you have finished the description or entered the maximum number of characters, press RETURN. You can enter and change descriptions as many times as you like. When you've finished the descriptions, return to the main menu by pressing f1. Choose option 3 and the descriptions will be saved on the disk with the filename DIR FILE. Be sure you have enough room on the disk for this file.

Option 2 allows you to perform various disk functions: scratch, rename, initialize, and validate. Option 3 saves the descriptions to the disk. If DIR FILE already exists on the disk, the program will erase the file and replace it with what is in memory. Option 4 allows you to work with other disks without having to rerun Directory Extension. Option 5 ends the program.

See program listing on page 120.

Sequential File Converter For *SpeedScript*

Ron Carnell

SpeedScript becomes an even more valuable word processor with this program that converts sequential files into _SpeedScript_ format. It's written in machine language for fast results. For the Commodore 64.

I've been using the Commodore 64 version of *SpeedScript*, in one version or another, since it was originally published in the January 1984 issue of COMPUTE!'s GAZETTE. The only drawback I've ever found is its incompatibility with sequential files. (*SpeedScript* uses program files.)

My database uses sequential files, and many downloaded programs are in sequential file format. And I have a few friends that use other word processors that create sequential files. In short, there are many instances when I'd like to load a sequential file into *SpeedScript*.

Over the years, COMPUTE! and COMPUTE!'s GAZETTE have published several BASIC programs designed to convert sequential files into *SpeedScript* files. And I've used them, mostly with great success. But BASIC is inherently slow; when my database started creating sequential files over 100 blocks long, it became obvious that I needed something faster.

"Sequential File Converter" (SFC), the program accompanying this article, is the solution. SFC accomplishes the same thing as the earlier BASIC conversion programs, but at the speed of machine language.

To type in the program, you'll need "MLX," the machine language entry program found elsewhere in this issue. After loading and running MLX, answer the starting and ending address prompts with the following:

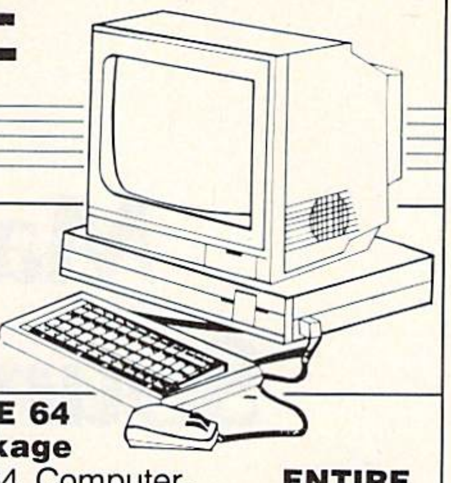
Starting address: 0801
Ending address: 0AD0

To use SFC, you need only load it and type RUN. You'll be prompted for the name of the source file (the sequential file) and for the name you'd like for the new *SpeedScript* file. (Be sure to have the correct disk in the drive when you enter the destination name.)

If you load SFC and type LIST, you'll find several lines of BASIC. With one exception, you should *never* change any of this code. The machine language is attached to the end of the BASIC program, and changing any of the BASIC code will almost certainly cause SFC to crash. The exception is line 140. In this line, SFC is adding ",s,r" to the end of your source filename. The *s* tells the disk drive to access a sequential file, and the *r* indicates that it should be read. You can, in this instance, change the *s* to another letter without changing the length of SFC and therefore without causing the machine language to crash. You might wish to do this if you want to change a file format other than sequential into a *SpeedScript* file. For example, a few word processors store their text as USR files. By changing that *s* to a *u*, you could make SFC a USR converter able to access USR files.

See program listing on page 113.

YOUR COMMODORE CONNECTION



AMIGA SYSTEM PACKAGE

Includes: AMIGA 1000 CPU, 512K, Mouse, Two 3½'' Double Sided Drives, Amiga RGB Monitor, Amiga DOS, Basic, Tutorial, Kaleidoscope, and Voice Synthesis Library.

ENTIRE SYSTEM $\$1599^{00}$

COMMODORE 128 SYSTEM

INCLUDES: CB128 Computer, CB1571 Disk Drive, Magnavox 8562 RGB/Composite Monitor. **$\$779^{00}$**

COMMODORE 64 System Package

INCLUDES: CB64 Computer, CB1541 Disk Drive, Taxan 220 Color Monitor

ENTIRE SYSTEM $\$499^{00}$

DRIVES

CBM1541	$\$199^{00}$
CBM1571	$\$249^{00}$
MSD-1	$\$219^{00}$
MSD-2	$\$459^{00}$
INDUS-GT (64/128)	$\$189^{00}$

MODEMS

Anchor Volksmodem	$\$59^{99}$
Anchor Amiga 80	$\$139^{00}$
Anchor 6480 (64/128) 300/1200 Baud	$\$119^{00}$
CBM 1660 (C-64)	$\$59^{99}$
CBM 1670 (C-128)	$\$179^{00}$

MONITORS RGB/COLOR

Taxan 220 composite	$\$179^{00}$
Magnavox 515	$\$289^{00}$
NEC 1216	$\$349^{00}$
Amiga 1080 Hi-Res	$\$299^{00}$

PRINTERS

EPSON Homewriter 10	$\$159^{00}$
OKIDATA Okimate 10	$\$199^{00}$
STAR SL10C	$\$219^{00}$
PANASONIC 1080	$\$199^{00}$
C.ITOH 7500 AP	$\$169^{00}$

DISKETTES

MAXELL SS/DD 5¼	$\$11^{99}$
NASHUA SS/DD 5¼	$\$9^{99}$
Sony SS/SD 5¼	$\$9^{99}$
Sony DS/DD 3½	29^{99}
GENERIC DS/DD w/Flip'n File 10 Disk Holder	$\$11^{99}$

ACCESSORIES

CompuGuard Surge and Spike Protector	$\$14^{99}$
Curtis SP2 Surge and Spike Protector	$\$39^{99}$
Curtis Safety Strip with 6 Outlets	$\$19^{99}$
Curtis Universal Printer Stand	$\$14^{99}$
Computer Specialties Power Plus	$\$54^{99}$
Amaray Disk Tubs 3½ or 5¼''	$\$9^{99}$

INTERFACES

CARDCO Super G	$\$49^{99}$
Digital Devices U-Print C	$\$44^{99}$
Orange Micro Grappler	$\$89^{99}$
PPI Printer Interface	$\$34^{99}$
Xetec 8K	$\$69^{99}$
Micro R & D MJ350	$\$59^{99}$

SOFTWARE FOR 64/128

ATARISOFT/Commodore

PacMan	$\$4^{99}$
Centipede	$\$4^{99}$
Defender	$\$4^{99}$
Dig Dug	$\$4^{99}$
Donkey Kong	$\$4^{99}$
Robotron	$\$4^{99}$

BRODERBUND

The Print Shop	$\$29^{99}$
Graphics Library 1,2,3 ea.	$\$17^{99}$
Bank Street Writer	$\$34^{99}$
Graphics Companion	$\$24^{99}$

CARDCO

Freeze Frame	$\$32^{99}$
S'more Basic	$\$44^{99}$

COMMODORE

Jane-integrated (128)	$\$39^{99}$
Assembler 64	$\$9^{99}$
Pilot 64	$\$9^{99}$
Geos for 64, 64C, 128	$\$9^{99}$

DATA SOFT

Genesis	$\$5^{99}$
Dallas Quest	$\$5^{99}$

HES

Benji Space Rescue	$\$3^{99}$
HES Games/Olympics	$\$3^{99}$

Professional Software

Fleet Systems II C128/64	$\$49^{99}$
Fleet Systems III C128	$\$59^{99}$
Trivia Fever	$\$19^{99}$

PFS

File	$\$32^{99}$

SPRINGBOARD

The News Room	$\$32^{99}$
Clip Arts	$\$19^{99}$
Clip Arts II	$\$27^{99}$

SUBLOGIC

Flight Simulator II	$\$37^{99}$
Jet Simulator C64	$\$37^{99}$

SPINNAKER

Rhymes & Riddles	$\$9^{99}$
Alf Cocor Caves	$\$7^{99}$
Bubble Burst	$\$9^{99}$
Trains	$\$9^{99}$

SOFTWARE FOR AMIGA

ACTIVISION

Hacker	$\$34^{99}$
Mind Shadow	$\$34^{99}$
Borrowed Time	$\$32^{99}$

COMMODORE

Textcraft	$\$59^{99}$
Graphicraft	$\$29^{99}$
Musicraft	$\$79^{99}$
TLC Logo	$\$79^{99}$
Amiga Pascal	$\$79^{99}$
Lattice "C"	$\$119^{00}$
Amiga Assembler	$\$79^{99}$
Lisp	$\$156^{00}$

ELECTRONIC ARTS

Deluxe Paint	$\$64^{99}$
Archon	$\$31^{99}$
One on One	$\$31^{99}$
Skyfox	$\$31^{99}$
Financial Cookbook	$\$39^{99}$
Arctic Fox	$\$31^{99}$
7 Cities of Gold	$\$32^{99}$
Deluxe Print	$\$74^{99}$

DISCOVERY

Marauder Back-up	$\$32^{99}$

INFOCOM

Wishbringer	$\$32^{99}$
Hitchhiker	$\$32^{99}$
Spellbreaker	$\$32^{99}$
Planetfall	$\$32^{99}$
Witness	$\$32^{99}$

MEGASOFT

A-Copier	$\$34^{99}$
A-Report	$\$44^{99}$

MICRO SYSTEMS

Analyze	$\$59^{99}$
Scribble	$\$59^{99}$
On-Line	$\$49^{99}$
Organize Data Base	$\$59^{99}$

MINDSCAPE

Halley Project	$\$34^{99}$
Deja Vu	$\$34^{99}$
Keyboard Cadet	$\$29^{99}$

COMPUTER MAIL ORDER

CALL TOLL-FREE 1-800-233-8950

477 East Third Street, **Dept. A410,** Williamsport, PA 17701

EDUCATIONAL INSTITUTIONS CALL TOLL-FREE 1-800-221-4283

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

1-800-268-3974
Ontario/Quebec

1-800-268-4559
Other Provinces

CANADIAN ORDERS
All prices shown are for U.S.A. orders
Call the Canadian Office for Can. prices

1-416-828-0866
In Toronto

Telex:06-218960
2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

Mastering 128 Sound And Music

Part 1

D. C. Holmes

As you'll see in this first installment of a series, you don't have to be a musician to create professional-sounding music on the Commodore 128. Several short, useful programs are included—this month and throughout the series—to demonstrate how just a few BASIC commands can make your 128 a powerful and versatile musical instrument.

The Commodore 64 offers extensive music-generating capabilities to the patient and interested programmer. The built-in Sound Interface Device (SID chip) has the potential for producing sophisticated electronic music, with the computer manipulating the various parameters of three independent voices simultaneously. Programming the SID chip is not that easy, however; a tedious process of POKEing values into memory locations is necessary to play even a simple musical composition. I suspect that many a masterpiece has been abandoned at the computer keyboard, owing to the frustration of dealing with PEEKs and POKEs.

Enter the 128. With the new sound commands available in BASIC 7.0, programming the SID chip is much less confusing and tedious. Technical understanding of memory maps and programming details is no longer a prerequisite for composers and arrangers. In fact, once the user becomes familiar with the PLAY statement, the painstaking chore of translating the manuscript into computer language is virtually eliminated. You can type the score directly into the 128—straight from the sheet music.

This series will explore the musical capabilities of the Commodore 128. Each month, we'll discuss different aspects of BASIC 7.0's sound and music statements. Each article will also include programs to illustrate what we've covered. We'll start by examining the default parameters for the ENVELOPE statement, and we'll take a detailed look at the TEMPO statement.

Default Envelopes

Let's first look at the ten default (built-in) envelopes defined in BASIC 7.0. These envelopes are named after the musical instruments whose sound they resemble:

Envelope #	Name
0	piano
1	accordion
2	calliope
3	drum
4	flute
5	guitar
6	harpsichord
7	organ
8	trumpet
9	xylophone

This month's first program is a short excerpt from Mozart's *The Magic Flute*. I programmed the computer to play it over and over again, each time using a different one of the default envelopes. When you run this program, you'll be able to compare the different sounds of the envelopes. It doesn't appear that Commodore necessarily intended for these preset envelopes to imitate the instruments for which they're named. More likely, they wanted to provide a sampling of the various possibilities, and named them to facilitate identification. To suggest that envelope 0 really sounds like a piano or that envelope 8 sounds just like a trumpet stretches the imagination a bit, but some of the envelopes do come reasonably close to the sound of the instrument they are named after.

In particular, I thought that envelope 2 sounded a lot like a steam calliope, so I selected that envelope for the second program this month, "American Patrol" by F. W. Meacham.

Take a look at line 80 of the program:

```
80 PLAY "V1T2V2T2V3T2"
```

In this line, we've specified that all three of the SID chip's voices (V1,V2, and V3) will play in envelope 2 (T2). Since we've not used the ENVELOPE statement to redefine envelope 2, the 128 uses default envelope #2 (calliope).

After you've typed this program in and saved it, try specifying a different envelope in line 80. To use the organ envelope (default envelope 7), you would type:

```
80 PLAY "V1T7V2T7V3T7"
```

Or you could even get fancy and play it with a mixed combo:

```
80 PLAY "V1T8V2T4V3T5"
```

This statement specifies envelope 8 for voice 1, envelope 4 for voice 2, and envelope 5 for voice 3. If you don't use line 80 at all, the 128 plays all three voices in the default envelope 0.

The TEMPO Statement

Composers usually specify a tempo for their works. It may be a general marking such as "Largo," "Andante," or "Allegro," or it may be very specific such as "♩ = 126," which indicates the exact tempo (126 beats per minute in this case). The TEMPO statement allows you to control the tempo of your musical program. The format for this statement is

```
TEMPO n
```

where *n* is a variable between 1 and 255. The equivalent musical notation is ♩ = 12.49 * *n* because the statement defines the duration of a quarter note at 4.805/*n* seconds, or 12.49 * *n* quarter notes per minute. Some examples of TEMPO statements and the corresponding musical notations:

TEMPO #			
4	♩ = 50	9	♩ = 112
5	♩ = 62	10	♩ = 125
6	♩ = 75	15	♩ = 187
7	♩ = 87	20	♩ = 250
8	♩ = 100		

If you do not specify a tempo, the 128 automatically sets the value of *n* to 8.

In Program 1, "The Magic Flute," the TEMPO statement is used in line 10:

```
10 TEMPO 13
```

This defines a tempo of ♩ = 162. In Program 2, American Patrol, it's found in line 20:

```
50 TEMPO 20
```

This defines a tempo of ♩ = 250. This might seem to be a very rapid tempo, but I've arranged this piece in 2:2 or *cut time*, so there are actually only 125 beats per minute.

After you've typed these programs in and saved them, try changing the value of *n* in the TEMPO statements to some other number between 1 and 255. Then run the programs again and listen to the effect this change has on the tempo.

Next month we'll explore the programming and synchronization of individual voices playing simultaneously (polyphonic sound). And we'll learn how to code sheet music or your own manuscript arrangements to play on the 128.

See program listings on page 114.

power BASIC

Line Count

Kenneth J. Rogerson

This utility is a boon to programmers: It helps you increase the speed of BASIC programs by reporting which lines are executed most often so that your efforts may be concentrated where they will be most effective. Although it's written in machine language, you don't need to know any machine language to use it. Versions for the Commodore 64, 128, Plus/4, and 16.

"Line Count" is a program I had waited a long time for someone else to write. I finally learned enough machine language to write it myself, however, and to make it as useful and efficient as possible. Line Count is a utility that reports which lines are executed most often in a BASIC program. With this information, you can then target inefficient or redundant areas of code that slow the program down.

Typing It In

Type in the version appropriate for your computer: Program 1 for the 64, Program 2 for the 128, or Program 3 for the Plus/4 and 16. Each of the programs has a checksum feature in line 20, which will detect typing errors in the DATA section. It's also recommended that you use "The Automatic Proofreader," found elsewhere in this issue, for error-free typing.

After you've finished typing in the program, save a copy to disk or tape. To use it, load it and type RUN. (*Plus/4 and 16 users must type the following* before *loading Line Count:* **POKE 44,32: POKE 8192,0: NEW.**) If it's been entered correctly, it will POKE the machine language into memory (locations 49152–49663 on the 64, 3072–3517 on the 128, and 4096–4479 on the Plus/4 and 16). After Line Count has been loaded and run, load the BASIC program you want Line Count to check. Before running your program, enter **SYS 49155** (64 version), **SYS 3075** (128 version), or **SYS 4099** (Plus/4 and 16 version) in direct mode, and press RETURN. This activates Line Count. In a moment, the READY prompt and cursor will reappear. Now run your BASIC program. When it's finished running—or if it stops for any reason—you can then get a report from Line Count. To do this, enter in direct mode **SYS 49152** (64 version), **SYS 3072** (128 version), or **SYS 4096** (Plus/4 and 16 version), and press RETURN. Line Count will list the line numbers that your BASIC program executed, followed by a number indicating how many times each line was executed.

As each section of Line Count reinitializes itself when activated, you can check several programs in succession without having to rerun Line Count. Simply load and run each program to be checked, remembering to activate Line Count with the appropriate SYS and to generate the report with the other SYS listed above.

Features And Limitations

Because it adds its own routine to the normal functions of BASIC, Line Count causes your program to run more slowly than normal. Line Count cannot keep track of more than 896 lines. If your program is longer, Line Count will record only the *first* 896 lines. And Line Count can count up to 65535. If a line is executed more times than that, Line Count will stop your program with an error message. Also note that Line Count may not work with other cartridges or utilities.

Line Count will not work reliably if either of the SYS commands (used to activate or generate a report) is used within a program. Use Line Count's SYS commands only in direct mode.

It is not important how your program is numbered or how many commands are on each of your program lines.

If a program runs *without* Line Count resident, it will run *with* Line Count, unless your BASIC program uses any of the following locations:

64 Version	128 Version	Plus/4 and 16 Version
49152-53247	3072-3517	4096-8191
251-254	251-254	251-254
776-777	776-777	776-777

If your BASIC program contains DATA statements, how they are counted will depend on where they are placed. DATA statements at the end of your program, which are not referenced by a GOTO or GOSUB and are preceded by END or STOP, will not be counted, no matter how often the DATA is read. If your program contains DATA statements placed in the normal flow of the program and therefore scanned by the BASIC interpreter, they will be counted once each time they are scanned no matter how often the DATA is read.

You may notice strange results when a FOR-NEXT loop is split between two or more lines. The explanation is simple. Assume that the program below is being checked and that there are no other lines in the program. Because there are no commands on line 100 after the FOR-TO portion of the loop, this line will be counted once, while lines 110 and 120 will be counted 100 times. This will occur in your programs in similar circumstances. If there is a command or statement on the same line after the FOR-TO command, the line will be counted the number of times that the loop itself is executed—in this case, 100 times.

```
100 FOR X=1 TO 100
110 REM
120 NEXT X
```

See program listings on page 116.

BASIC for beginners

Learning The Language

Larry Cotton

We're happy to welcome Larry Cotton and his new column, "BASIC For Beginners," to COMPUTE!'s GAZETTE. *Larry is an engineer, keyboard musician, and avid Commodore computer enthusiast. Larry purchased a Commodore 64 in 1982—for $600—and has been a* GAZETTE *reader since the first issue in July 1983. He recently built his own MIDI interface for the 64. Several of Larry's programs have been published in the* GAZETTE, *including "Baker's Dozen," "Zounds!," "Soundpix," and "Super Synth."*

Thanks for the welcome. I'm delighted to be here. Let's for a moment go back to June 1982. The home computing revolution was rapidly gathering momentum, and I decided that it was not going to pass me by. But how does one catch a train that's running wild and never stops? And which train to catch anyway? The computer choices in my price range were the Apple II, the Atari 400 and 800, and the Commodore 64.

I finally decided on the 64—primarily because I liked the musical capabilities of the machine. Software for the 64 at that time consisted of a Monopoly clone and a typing tutor originally written for the VIC-20.

It must have been because of the dearth of software that I started learning BASIC programming. Today there's so much software for the 64 that it's more a temptation not to learn to program. But don't succumb to that way of thinking. Learn to program. It will make you think more clearly, make you more organized—and probably keep you up later at night.

This column is for those who may have recently bought a Commodore computer, or who just want to learn BASIC. I'll start from scratch and try not to leave you behind. And I'll assume only that you own a 40-column Commodore computer, a disk drive, and a TV or monitor—all properly connected and working.

What Is BASIC?

BASIC is a language which any computer with a built-in BASIC interpreter, or translator, understands.

There are several versions of BASIC, each with its own peculiarities. Commodore's version is not totally compatible with Atari's; Atari's is not totally compatible with Apple's. And the Commodore 128's version of BASIC is much larger than the 64's.

Simply typing a BASIC command on the screen and pressing the RETURN key causes something to happen. When you do this, the computer responds immediately—it is in the *immediate* mode of operation. If you precede the BASIC commands by numbers, called line numbers, the computer takes them to be part of a program, and puts them into its memory. When you type RUN, the computer executes them in line number order. This is called the *program* mode of operation.

BASIC programming means typing. If you touch type, you're ahead of the game. But if you're unfamiliar with the standard keyboard layout, it may be rather tough learning BASIC and how to type at the same time. There are a lot of good commercial typing tutor programs to help you become more familiar with the keyboard. But even if you decline formal instruction, programming itself can hone your typing skills.

Keywords: The Heart Of BASIC

The user's guide which came with your computer is jam-packed with information on BASIC programming. If you haven't referred to it (except the part on how to hook up your computer), you really should.

The meat of the guide is the section on BASIC keywords: English-like codes which the computer understands—the vocabulary of this language.

One keyword you've probably already used is RUN. Some commercial programs require that you type RUN to start them. When you write your own BASIC programs, that's what you'll type to get them running, too.

Another keyword (or command) which you may know is LIST. LIST usually shows you (on screen) a "list" of the BASIC programs in your computer. It won't show you cartridge programs, or any other programs which don't recognize the word LIST. BASIC recognizes the commands LIST and RUN, but if you type either one with the computer's memory "empty" or with a program in an unusual area, nothing happens.

Turn your computer on. If it's already on with a program in it, remove any cartridge you might be using, and turn the computer off, then on again. This resets its memory, making it ready to accept anything you want to put into it.

The incessantly flashing block you see on the screen is the *cursor* (you'll eventually discover how it got its name); you can make it careen all around the screen with the cursor keys (at the lower right corner of your keyboard). It's a good idea to type BASIC commands in a screen area which has no other typing. Of course, your typing always starts wherever the cursor is.

When you type RUN or LIST and press RETURN, you are utilizing the computer's immediate mode. Most other BASIC commands can be entered in the immediate mode, also.

Two more keywords which your computer will instantly recognize are PRINT and GOTO, which we'll delve into more deeply in future columns. For now, suffice it to say that PRINT and GOTO mean just what they say—whatever the computer "sees" after the word PRINT will be printed on your TV or monitor screen (not on your printer, if you have one). And GOTO means "go to" a designated program line.

PRINT and GOTO can each be used in immediate mode, but for now, let's see what their effects are in a simple BASIC program. You can use any line numbers you want to, as long as they're in ascending order. Most programmmers prefer to use line numbers in increments of 10, so that if they leave something out, it can be added between existing line numbers.

Clear the screen (press SHIFT and the CLR/HOME key in the upper right corner of the keyboard) and type this program:

```
10 PRINT "COME TO BASIC WEL"
   (press RETURN)
20 GOTO 10
   (press RETURN)
```

Note the quotation marks around the strange message and the lack of space between the letters in GOTO. When you press the RETURN key with the cursor on each line, that line is entered in the computer's memory.

Before you type RUN, try to predict what this program does. Now clear the screen again. The program disappears from the screen, but is very much alive and well in the computer's memory. Type LIST and press the RETURN key. Voilà.

Now type RUN and press RETURN. You should see a never-ending column of "COME TO BASIC WEL"s.

Line 10 tells the computer to print the words in quotation marks on the screen. Line 20 is then immediately executed, but it does nothing except put the program in a repeating loop by sending control back to line 10 again. Since the screen is not cleared from within the program, the words are automatically printed underneath each other.

How do you stop it? Just press the RUN/STOP key on the left side of the keyboard.

If you don't see a column of "COME TO BASIC WEL"s—or if you see the words ?SYNTAX ERROR in line 10 or 20—LIST the program and check your typing carefully. The computer is a perfectionist. When it encounters a command it doesn't recognize, it invariably brings the program to a screeching halt and prints ?SYNTAX ERROR (in whatever line the error occurred in).

Commodore computers have an excellent built-in screen editor which makes correcting programming errors much easier than on some other computers. The easiest way is to place the cursor over the mistake and type the correction. If you're not familiar with other editing features, consult your user's guide. Be sure to press RETURN with the cursor on that line so the computer knows you've changed it.

BASIC provides a way to arrange PRINTed messages in columns. Immediately after the second quotation mark in line 10, type a comma, and press RETURN. The line should look like this:

```
10 PRINT "COME TO BASIC WEL",
```

Run the program again. You should see two columns if you're working with a 40-column 128, 64, Plus/4, or 16. On the 80-column 128, you should see four columns.

To make sense out of this message, add a semicolon (;) immediately after the second set of quotation marks in line 10, and press RETURN. Run the program again. Creative possibilities for using the PRINT command should be coming to mind right now.

This is a very elementary exercise, to be sure. We'll investigate many more sophisticated uses of the PRINT command next month.

If you have any questions on Commodore BASIC, send them to me at the address below. I won't be able to answer personally, but I'll save the best for a future column.

Larry Cotton
COMPUTE!'s GAZETTE
P.O. Box 5406
Greensboro, NC 27403

horizons

Changing Your Computer's Personality

Todd Heimarck
Assistant Editor

Commodore has sold between five and six million 64s, which makes it the most popular home computer in this country, if not the world. Although production of the 64 was halted once or twice in 1985, continued and steady demand for the 64 forced Commodore to restart production of this old favorite. With the announcement a few months ago of the new 64C, Commodore has made a commitment to continue supporting the 64.

Some 64 owners have expressed concern about compatibility. Is the 64C somehow better than the 64? Is there a way to upgrade an older 64 to a 64C?

The two machines are functionally identical. BASIC is the same, PEEKs and POKEs are the same, input/output ports are the same, and so on. The 64C's keyboard looks something like the 128's case, but the new look has no bearing on software compatibility.

What else is new? The 64C package includes several disks and some improved documentation. First, there's the *Graphics Environment Operating System* (*GEOS*) disk, which includes two programs, *geoWrite*, a word processor, and *geoPaint*, a graphics program. A manual of 100+ pages explains how to use the programs. A second *GEOS* disk contains various utilities, including a notepad and calculator. (For more about *GEOS*, see the June and August issues of the GAZETTE.)

The flip side of the *GEOS* disk contains the software necessary to log onto the telecommunications service QuantumLink. In addition, you get an access code which allows you to get into QuantumLink and sign up as a member. Of course, you'll need a modem before you can begin to use this service. Finally, there's an educational children's game, "Odell Lake," which illustrates the interaction between different species within an aquatic ecosystem.

The new *System Guide* is much better than the *User's Guide* once included with the 64. The *System Guide* takes an organized approach to learning BASIC, with plenty of examples. There are also warranty cards and a variety of pamphlets.

The question is, will *GEOS* catch on among owners of the 64C? *GeoWrite* is a pretty good word processor; its primary advantage is that it can print multiple fonts: different typefaces, different styles (such as bold, italics, or underline), and different sizes. And *geoPaint* is a pretty good drawing utility, suitable for making posters or page layouts.

Software Development For *GEOS*

Will there be any other programs written for the *GEOS* environment? Berkeley Softworks is said to be working on several projects. Additional typefaces may soon be available, and they're adding printer drivers for printers not currently supported. And there are rumors of a *geoBASIC*, which may come later.

More importantly, they've put together a comprehensive *Programmer's Reference Guide*, to be published at the end of this year. The preliminary photocopied version appears to be thorough and comprehensive. It includes full details on drawing lines and rectangles, creating icons, clipping areas of the screen, adding character sets, reading the mouse and joystick, using sprites, accessing files, handling strings, displaying dialog boxes, and printing screen dumps, as well as information on turbo (fast disk) routines and math routines. There's also a memory map and a list of equates to use for machine language programming.

By publishing this information, Berkeley Softworks has made it possible for programmers and software publishers to begin developing new applications for use with *GEOS*.

Given the memory limitations and speed of the 64, *GEOS* is a good implementation of an icon-based operating system, similar to what's available on the Apple Macintosh, Commodore Amiga, and Atari ST. You don't have to buy a 64C to get it; if you have a 64, you can obtain *GEOS* for a suggested list price of $59.95.

IBM Compatibility For The 128

If you own a 128 and want a Macintosh-like operating system, *GEOS* should be available soon, probably as a disk, but there's an intriguing possibility that it will be available on a chip. The 128 has an empty internal ROM socket, which is addressed as bank 4. The 32K which fits there has plenty of room for *GEOS*. Adding a new operating system would give the 128 a fourth personality to complement 64 mode, 128 mode, and CP/M.

Although Commodores are popular in the home, IBM is still the most popular personal computer for business. If you have an IBM at the office and a 128 at home, you might think that the two machines are totally incompatible. In one sense, they're not.

The 1571 disk drive is valuable in CP/M mode because it can read Osborne, Kaypro, Epson, and IBM CP/M-86 disks. If the 1571 can read IBM CP/M-86 disks, why can't it read standard IBM MS-DOS disks? Well, it can.

The following program reads a specific track and sector from an IBM disk.

```
XG 10 BANK15:PRINTCHR$(14)
DF 20 T=29:S=2:N=1: REM TRACK,
       SECTOR, NUMBER OF SECTO
      RS
QG 30 SD=0: REM SD=1 TO READ S
      IDE ONE
BM 40 OPEN15,8,15
DR 50 PRINT#15,"U04";
PR 60 PRINT#15,"U0"+CHR$(64OR(
      SD*16))+CHR$(T)+CHR$(S)+
      CHR$(N);:SLEEP1
MS 70 FORJ=0TO511:P=PEEK(56576
      ):POKE56576,XOR(P,16):A=
      PEEK(56332):POKE3072+J,A
      :NEXT
HD 80 PRINT#15,"UJ":SLEEP2:CLO
      SE15
DR 90 A$="M C00 DFF"+CHR$(13)
RJ 100 FORJ=1TOLEN(A$):POKE841
       +J,ASC(MID$(A$,J,1)):NE
       XT:POKE208,LEN(A$)
SM 110 MONITOR
```

Adjust the variables in lines 20 and 30 to read other sectors. Line 50 sends the burst command that tells the 1571 to figure out which kind of disk it's got, and line 60 tells it to start reading. Lines 90–110 use the dynamic keyboard technique to read the memory which has been filled with the characters from the IBM disk. Since the IBM uses true ASCII instead of Commodore ASCII, the upper- and lowercase letters will be switched.

The program isn't especially useful, but it does demonstrate that a 128 and 1571 can read an IBM disk.

A company named S.O.G. W.A.P. Software has recently announced a program that takes this idea a step further. Their $29.95 program, *The Big Blue Reader*, reads IBM disk files, translates them into Commodore ASCII, and writes them to a Commodore disk. The program can also read a Commodore disk and write back to the IBM disk in true ASCII. This means you could take home an IBM word processing disk from work and make changes or additions to your files on a 128 and 1571.

This adds a potential fifth personality to your 128. With a 512K expander, an 80-column RGB monitor, and the IBM file translator, the 128 begins to look something like an IBM. It might even be possible to create an IBM emulator chip to drop into the empty socket. It's highly unlikely that we'll see complete IBM emulation, but it would certainly be possible to create a word processor, spreadsheet, or database that reads and writes files from Commodore, CP/M, and IBM disks. It would add even more power to the 128.

GEOS
Berkeley Softworks
2150 Shattuck Ave.
Berkeley, CA 94704
$59.95

The Big Blue Reader
S.O.G.W.A.P. Software
611 Boccaccio Ave.
Venice, CA 90291
$29.95

hints & tips

If you've discovered a clever time-saving technique or a brief but effective programming shortcut, send it to "Hints & Tips," c/o COMPUTE!'s GAZETTE. *If we use it, we'll pay you $35. We regret that, due to the volume of items submitted, we cannot reply individually to submissions.*

Commas And Colons Not Allowed

Steven Dodds

One of the main problems with the INPUT command is that it won't read past a comma (,) or colon (:). If the person using the program enters either of these characters during an input routine, the computer rudely responds with an ?EXTRA IGNORED message.

On the 64, one solution is to place **POKE 631, 34: POKE 198, 1** just before the INPUT line. This puts a quotation character (") into the keyboard buffer. (The INPUT routine accepts commas and colons within quotation marks.) On the Plus/4 and 16, substitute **POKE 1319, 34: POKE 1343, 1**, and on the 128, **POKE 842, 34: POKE 208,1.** This method isn't perfect, though. If the user types a quotation mark as part of the input, the effect of the POKEd quote is negated.

Another more interesting solution is to call a built-in ROM routine. The routine beginning at location 44025 on the 64 reads keypresses and puts ASCII values into the input buffer at 512. After the SYS 44025, just PEEK 512 and following locations. The end of the input is marked by a zero byte. Try running the following program on a 64:

```
MX 10 PRINT"ENTER NAME"
GK 20 SYS 44025
PX 30 FORX=512TO600
RD 40 A=PEEK(X):IFA=0THENX=900
      :GOTO60
FK 50 B$=B$+CHR$(A)
GJ 60 NEXT
MF 70 PRINT"YOU ENTERED ";B$
```

If you're using a 128, substitute **BANK 15: SYS 20371** in line 20 and change the FOR-NEXT loop in line 30 to count from 512 to 771. On the Plus/4 and 16, replace line 20 with **SYS 37190** and count from 512 to 609 in line 30.

The FOR-NEXT loop could be revised to watch for certain types of input. When entering a person's name, some people type the last name, a comma, and the first name. Others might type first name and then last name, separated by a space. While the FOR-NEXT loop is checking the input buffer, a comma would indicate that the last name was typed first. If a single space divides the names, the user probably typed the first name first. Checking for commas and spaces would make your program smart enough to accept both types of input.

Another way to use this routine is in an adventure game parser. Spaces separate words, commas separate items in a list, and periods end sentences. By looking for certain characters and punctuation marks, you'll be able to figure out a little more easily which commands have been typed.

Disabling Keys On The 64

Timothy R. Alexander

Several characters lead to trouble within an INPUT statement. Commas and colons trigger the ?EXTRA IGNORED error mentioned above. The cursor-up and -down keys can make the screen scroll. And CLR clears the screen, which erases any kind of menu of choices you might have printed on the screen.

There are times when it would be simpler to disable certain keys during input. The program below does just that. It's a short machine language routine, for the 64 only, which copies BASIC and the Kernal from read only memory (ROM) to the random access memory (RAM) underneath. It then modifies the keyboard matrix decode tables, which start at location 60289, to ignore the comma, colon, cursor-up, cursor-down, CLR, and HOME keys. The rest of the keyboard is unaffected.

```
BS 10 FORX=828TO875:READA:CK=C
      K+A:POKEX,A:NEXT
PH 20 IF CK<>8492 THEN PRINT"P
      LEASE CHECK THE DATA STA
      TEMENTS":END
CB 30 SYS828
MJ 828 DATA 120,169,000,168,13
       3,251
XQ 834 DATA 169,160,133,252,17
       7,251
GA 840 DATA 145,251,136,208,24
       9,230
PP 846 DATA 252,208,245,169,25
       5,141
HF 852 DATA 136,235,141,174,23
       5,141
SF 858 DATA 176,235,141,180,23
       5,141
DP 864 DATA 201,235,141,245,23
       5,169
EB 870 DATA 058,141,070,171,08
       8,096
```

You may use this as a standalone program or as a one-shot subroutine within a larger program. To turn the six keys off, place a **POKE 1, 53** in your program just before the INPUT statement. To turn the keys back on, **POKE 1, 55**:

```
CG 10 POKE1,53
QS 20 INPUTA$:PRINTA$
EK 30 POKE1,55
```

Since the first POKE disables the comma, you won't be able to type the second POKE in direct mode; it needs a comma between the 1 and 55. If you get stuck in direct mode without a comma, press RUN/STOP–RESTORE. The keyboard will go back to normal.

There's a much simpler way to use this technique on the Commodore 128. See "KeyDef" in the August issue.

computing for families

Drum Suits, Rhythm Sticks, And Digital Pianos

Fred D'Ignazio
Associate Editor

Recently I visited the National Association of Music Merchants (NAMM) Show in Chicago, a twice-a-year extravaganza showcasing all the latest musical instruments, from nine-foot concert grand pianos to electronic harmonicas and violins. I carried my little Sony 8mm camcorder with me to shoot some of the show's highlights. Here's what I saw:

- The BrocktronX Drum Suit. The inventor, Brock Seiler, demonstrated the drum suit to crowds of enthusiastic showgoers. He strapped drum suit pads around his waist and thighs, and wore percussion-sensing boots (to make "Stompercussion"). All the parts of the drum suit were MIDI-connected to electronic keyboards and drum machines. Brock created some impressive music by banging on different parts of his body with two wooden drumsticks.

- The Dynacord Rhythm-Stick. The Rhythm-Stick looks like a fancy electric guitar—minus the strings. The Rhythm-Stick is really not a guitar at all, but a drum in disguise. You play the Stick by thumping on it with your fingers. Different parts of the Stick can be programmed to make different drum sounds, including high-hats, hand claps, tom-toms, cymbal crashes, and so on.

- The Casio SK-1 Digital Sampling Keyboard. At $130, the SK-1 costs only one-tenth as much as professional sampling keyboards. Its sampling rate is low (only 9.3 kHz), but it creates credible sounds, and you can carry it anywhere and record any sound—such as a doorbell, telephone ring, or dog barking. Hal Leonard Publishing Company and Sight & Sound make book/cassette packages that give you dozens of prerecorded sample sounds to record on your keyboard. Once the sound is in the keyboard, it appears—at a different pitch—when you press each of the keyboard's 32 keys.

- The Roland Digital Piano. Three members of our family play the piano, and we were recently hunting for a good used baby grand. We found one, but we couldn't afford the price its owners wanted—$5,000. Now we've found an alternative: the Roland Digital Piano. An extra piano module fits in a box the size of a briefcase. The piano keyboard has 88 full-sized, weighted wooden keys and a stand. The entire unit costs only $1,400.

The sound that comes out of the piano is unreal—just like that of a nine-foot Steinway grand piano. And if you get tired of that, you can switch to a harpsichord or a barroom honkey-tonk. The instrument has a MIDI connection, so you can plug it into other electronic instruments (like the 64 or 128) in your music studio.

There were almost as many microcomputers at the NAMM Show as keyboard synthesizers and guitars. Here are some of the 64 and 128 products that I saw during my NAMM wanderings:

- *SampleScope.* You plug your AKAI S-612 sampler into your 64 or 128 and boot up *SampleScope* to edit sound samples by manipulating graphic images of the sampled sound wave on the computer display screen (Ultimate Media, 275 Magnolia Ave., Larkspur, CA 94939).

- **Korg 800 Series Software.** You connect your 64 or 128 to a Korg 800 series keyboard via a MIDI interface—for example, Korg/Passport's MH01-C or MH02-C, Music Data's MIDI Interface, or Dr. T's Interface (Korg USA, 89 Frost St., Westbury, NY 11590).

- **Passport Hardware/Software.** Passport Designs can be your "one-stop store" for MIDI hardware and software for 64 and 128 computers, including the Passport MIDI Starter Kit, the Passport MIDI Home Pac, the Education Pac, the MIDI Beatles Album, the MIDI Voice Librarian, and the Passport MIDI Interface (Passport Designs, 625 Miramontes St., Half Moon Bay, CA 94019).

- **Sonus Software.** Sonus's new "Super Sequencer 128" ($275) is a professional recording system and librarian for the Commodore 128. It lets you record up to 16 sequences (song parts) on eight sound tracks from your keyboard synthesizer to your 128. Sonus also publishes several other programs, including the "Super Sequencer 64," the "MIDI processor," and the "RX-11, 15, 21" for controlling a Yamaha drum machine (Sonus, 21430 Strathern, Suite H, Canoga Park, CA 91304).

- **Syntech's MIDI Interface and *Studio One*.** Syntech's MIDI Interface and *Studio One* turn your 64 or 128 into a multitrack recording studio (Syntech Corporation, 5699 Kanan Rd., Agoura, CA 91301).

- **Education Courseware Systems/MEI/Computers & Me.** ECS, MEI, and Computers & Me have a large array of 64 and 128 software and hardware products that teach music concepts and interpretation to music students (Electronic Courseware Systems, 1210 Lancaster Dr., Champaign, IL 61821; MEI, 328 E-1 1300 North, P. O. Box 599, Chesterton, IN 46304; Computers & Me, Ashbrook Rd., Exeter, NH 03833-9733).

machine language for beginners

The First Steps

Richard Mansfield
Senior Editor

Why do people learn machine language? The answer is that it's the best possible, most effective way to program a computer.

Other languages are compromises because they have to be *translated* before the computer can understand them. Machine language (ML) speaks directly to the computer's brain, its microprocessor, in the language the microprocessor itself uses. That's why it's called machine language.

The Most Effective Programs

Every other language has a provision for calling ML subroutines to help speed things up. In BASIC, the provision is the SYS command, which transfers control of the computer to an ML routine located at the address of the SYS. SYS 40000 will cause the BASIC interpreter to stop while the ML located at address 40000 executes. There's also a provision for returning control to BASIC: ML's RTS command.

But why all this concern about speed? Computers are fast, very fast. Why would we need to maximize their velocity?

Let's look at an example. Suppose you want to write a word processor. You make a list of the features you want to include and then program it in BASIC. When you run it, you find that it moves ponderously, taking 40 seconds to search for a particular word in a short document.

So, because it is by far the most effective way to program a computer, many people take the time to learn ML. As you'll see, it's only a mystery until you know a bit about it.

What You Need To Get Started

Now let's get going. You'll need a special tool to write a machine language program: an *assembler*. The Commodore 64 comes with BASIC built in, so you simply start typing in a BASIC program and it can be run without further ado. However, when you write an ML program, it must be analyzed by an assembler and translated into the raw numbers which make up an executable ML program. This month's program is a simple assembler, written in BASIC. After you type it in, you'll be ready for the first lesson. (Commodore 128 users should be in 40-column mode.)

Load the assembler into your 64 or 128. Then type RUN. You'll see a list of available options, but let's ignore them for now. When the assembler asks you where you want your finished ML program to start in memory, type in 1824 and press RETURN. The screen will go blank, and the assembler will print the first "line number," which is really the address where the assembler will POKE your first command. Next, type LDA #0 and press RETURN. The assembler will do two things: It will show you the numbers it translated (169 and 0) and POKE them into memory. Since we're storing this little ML program into screen RAM, you'll actually see the ML being built on the screen (you can put ML programs virtually wherever you want in RAM).

Next type in TAY. Then STA 1024Y. (If you're using a 64, you need to type an extra line—STA 55296Y—following STA 1024Y, to fill color memory.) Then type DEY, and after that, BNE 1827. Then RTS. And finally, END. In each case, press RETURN after typing in the instructions. If you make a mistake, the assembler will tell you and will reprint the address so you can try again. If you make a *big* mistake, such as hitting RUN/STOP–RESTORE, you'll need to run the assembler and start over.

Here's what you should see on your screen if you use a 128:

```
1824  ? LDA    #0        169 0
1826  ? TAY              168
1827  ? STA    1024Y     153 0 4
1830  ? DEY              136
1831  ? BNE    1827      208 250
1833  ? RTS              96
1834  ? END
    PROGRAM IS FROM 1824 TO 1834
```

And here's what you see if you use a 64:

```
1824  ? LDA    #0        169 0
1826  ? TAY              168
1827  ? STA    1024Y     153 0 4
1830  ? STA    55296Y    153 0 216
1833  ? DEY              136
1834  ? BNE    1827      208 247
1836  ? RTS              96
1837  ? END
    PROGRAM IS FROM 1824 TO 1837
```

Try It Out

After you type END, the assembler returns you to the normal BASIC mode. Those strange symbols near the bottom of the screen are the complete ML program. Be very careful that you don't type anything over them, and don't do anything that would cause the screen to scroll or clear. That would wipe out the program. When you run the program, it will print (no, *print* suggests that it will be slow; it will *slap*) 256 @ characters on your screen. Try it. Type SYS 1824.

Next month, we'll explain what these odd three-letter commands mean, and we'll build a useful utility program in ML.

See program listings on page 115.

Tom R. Halfhill, Staff Editor

simple answers to common questions

Each month, COMPUTE!'s GAZETTE *tackles some questions commonly asked by Commodore users. If you have a question you'd like to see answered here, send it to this column, c/o* COMPUTE!'s GAZETTE, *P.O. Box 5406, Greensboro, NC 27403.*

Q. Is there any way to change a disk name without reformatting and losing the rest of the material which might be on the disk? I haven't been able to locate any information that applies to this problem.

A. There's no easy way to do this. The disk name which you assign when formatting a disk (for example, OPEN 15,8,15:PRINT #15, NEW0:*diskname,ID*":CLOSE 15) remains in the disk directory until the disk is reformatted. And, of course, reformatting wipes out any existing information on the disk.

It is possible to change the disk name by using a special utility program which lets you examine and alter individual sectors and bytes on the disk. An example of this type of utility is "Disk Editor," published in the February 1986 issue. However, this technique is recommended only for advanced users, since it's very easy to scramble the directory and mess up the whole disk if you're not absolutely sure what you're doing.

A safer—though more tedious—way to change the disk name would be to copy all the files you want to save onto a scratch disk, reformat the original disk with the new disk name you desire, and then copy all the files back onto it from the scratch disk.

Q. Before acquiring my Plus/4, I had read that it would not run BASIC programs written for the Commodore 64. However, I now have several programs that were written for the 64 that run fine on the Plus/4. These range from a database manager to a tax record program. My question is, why not write all BASIC programs in a format that can be run on all Commodore 64s and Plus/4s? I know that machine language programs must be different, but BASIC programs can work for the 64, Plus/4, and 128 in 64 mode.

A. It *is* possible to write BASIC programs that will work on both the Commodore 64 and Plus/4. It's also possible to write BASIC programs that will run interchangeably on a Commodore, Apple, Atari, and IBM. But the programs would have to be very limited in scope and function. You'd have to restrict yourself to the fundamental commands which are common to nearly all versions of BASIC, avoiding the more advanced commands that take advantage of machine-specific features. Programs written for different computers have to consist entirely of text and use no fancy screen formatting, no graphics, no color, and no sound.

The BASIC programs you mention—including a database manager and a tax record program—apparently meet these requirements. So would many other simple BASIC programs. But more complex programs which attempt to make the most of a computer's special features just cannot be made to work in different dialects of BASIC. The commands for using these features have little or nothing in common.

For example, the Plus/4's BASIC includes several commands that make it relatively easy to create graphics and sound. But the Commodore 64's BASIC does not. Graphics and sound on the 64 must be accomplished with PEEKs and POKEs, commands which directly access memory locations within the computer. These PEEKs and POKEs won't work on the Plus/4 because the two computers have different memory layouts.

Also, parts of some BASIC programs are written in machine language to improve performance. It would be a lucky coincidence indeed if these machine language routines happened to work on a computer for which they weren't designed.

In a way, it's unfortunate that so few computers are compatible with each other, even when programs are written in what's supposed to be the same language. But this is what makes it possible to have many different computers to choose from, each with its own unique combination of special features.

Q. I want to answer the question from the reader about hooking up VIC-20/Commodore 64–style joysticks to the Plus/4 and Commodore 16 ["Simple Answers to Common Questions," June 1986]. There is a store in West Germany that has the proper interface cable. Its address is Conrad Electronic Center GmbH & Co., Kurfurstenstrasse 145, 1000 Berlin 30, West Germany. The aaapter costs 6.90 in German marks, which is about $3 in U.S. currency (excluding postage). If you write, ask for the Computerkatalog II. I'm enclosing an order form.

A. Our thanks to Axel R. Oberhauser of the Province of Buenos Aires, Argentina, for supplying this information. Incidentally, the order form he enclosed bears another address for this company: Conrad Electronic GmbH, Klaus-Conrad-Str. 1, Postfach 11 80, D-8452 Hirschau, West Germany. You might try writing either or both addresses. Write to the attention of the *Geschäftsführer* (proprietor), Klaus Conrad.

User Group Update

When writing to a user group for information, please remember to enclose a self-addressed envelope with postage that is appropriate for the country to which you're writing.

Send typed additions, corrections, and deletions for this list to:

COMPUTE! Publications
P.O. Box 5406
Greensboro, NC 27403
Attn: Commodore User Groups

User Group Notes

The Central Dakota Commodore Club's new address is Rt. 3, Captain Leach Dr. #12, Mandan, ND 58554.

The Champaign-Urbana Commodore Users Group's new address is 802 N. Parke St., Tuscola, IL 61935

The Tulsa Area Commodore User's Group (TACUG) has a new address: 2851 East 21st Place, Tulsa, OK 74114.

The Crossroads Commodore User's Group has changed its address to 609 Cambridge, Victoria, TX 77901.

New Listings

CALIFORNIA

Commodore Hayward Users Group, P.O. Box 2072, San Leandro, CA 94577

ILLINOIS

Pros and Newcomers in Commodore (PANIC), c/o Vermilion County ESDA, 2 E. South St., Danville, IL 61832

LOUISIANA

Acadiana Commodore Computer Club (ACCC), P.O. Box 31412, Lafayette, LA 70503

MISSOURI

Columbia Commodore Users, P.O. Box 7633, Columbia, MO 65205

NEW JERSEY

Garden State Commodore User Group, 89 Stratford Rd., Tinton Falls, NJ 07724

Hunterdon Commodore Users Group, P.O. Box 724, Flemington, NJ 08822

NEW YORK

St. Francis College Microcomputer Users' Group, Microcomputer Center, 180 Remsen St., Brooklyn, NY 11201

OHIO

Cuyahoga Falls Commodore Club, P.O. Box 3025, Cuyahoga Falls, OH 44225

TEXAS

General Users Group (GUG), P.O. Box 3348, Stinnett, TX 79083

WEST VIRGINIA

Mid-Ohio Valley Commodore Club, Inc. (MOVCC), P.O. Box 2222, Parkersburg, WV 26101-2222

Outside The U.S.

AUSTRALIA

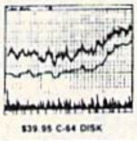
Yarra Valley Commodore Users Group, P.O. Box 176, Lilydale, Vic., 3140, Australia

BRAZIL

Curitiba Commodore Club (C.C.C), Rua Adolfo Stedile, 52, Bom Retiro, 80.520, Curitiba, Parana, Brasil

FINLAND

CSI, c/o Christer Nybergh, Ylapellontie 1.as.l., 39500 Ikaalinen, Finland



A Gateway To Machine Language

Richard Mansfield, Senior Editor

If you're thinking about learning—or starting to explore—machine language, here's an effective approach to this powerful and fascinating programming language. This small dictionary, arranged alphabetically, includes all the major BASIC commands. If you need to accomplish something in machine language—TAB, for example—look it up in this article to see how it's done. If you already understand BASIC, you'll find this an efficient way to quickly grasp the fundamentals of machine language, your computer's own native tongue. Excerpted from *128 Machine Language for Beginners* (COMPUTE! Books, 1986), some of the information here is 128-specific, but most applies directly to any Commodore computer.

The following machine language (ML) equivalents for BASIC commands represent only one of several possible solutions to each programming goal. There are usually many ways in ML to accomplish a given task because ML is so much freer than BASIC. Of these choices, one might work faster, one might take up less memory, and one might be easier to program and understand. For this article, example routines were selected to favor those which are easier to program and understand.

In any case, at ML's extraordinary speeds, and with the large amounts of RAM memory available to today's computerists, it will be rare that you will need to opt for velocity or memory efficiency.

CLR

In BASIC, this clears all variables. Its primary effect is to reset pointers. It is a somewhat abbreviated form of NEW since it does not "blank out" your program as NEW does.

CLR, in fact, is rarely used.

We might think of CLR, in ML, as the *initialization* phase of a program which erases (fills with zeros) the memory locations you've set aside to hold your ML flags, pointers, counters, and so on.

Before an ML program runs, you will usually want to be sure that some of its variables are set to zero. If they are in different places in memory, you will need to zero them individually:

```
2000  LDA  #$0
2002  STA  $1990    Put zero into one of the
                    "variables."
2005  STA  $1994    Zero another byte which needs to
                    be initialized.
```

On the other hand, if you've put all your variables together at the end of your ML program, the job is easy: Just loop through the list, putting zero in each variable. BASIC sets up a group of its variables (pointers) in zero page, so you can use a loop to zero them out:

```
2000  LDA  #$0
2002  LDY  #$0F     Y will be the counter. There are 15
                    bytes to zero out in this example.
2004  STA  $199,Y   The highest of the 15 bytes.
2007  DEY
2008  BNE  $2004    Let Y count down to zero, BNEing
                    until Y is zero, then the Branch if
                    Not Equal will let the program fall
                    through to the next instruction at
                    $200A.
```

CONT

This BASIC command allows your program to pick up where it left off after a STOP command. You might want to look at STOP, below. In ML, you can't usually get a running program to stop with the RUN/STOP key. If you like, you can write a subroutine which checks to see if a particular key is being held down on the keyboard and, if it is, BRK:

```
3000  JSR  $FFE4;  Routine to get the key currently
                   pressed.
3003  BEQ  3000;   If nothing is currently pressed,
                   keep looking.
3005  CMP  #13     This is the RETURN key on your
                   machine, but you'll want to CMP
                   here to the value that appears in
                   the "currently pressed" byte for
                   the key you select as your "stop"
                   key. It could be any key. If you
                   want to use A for your "stop" key,
                   try CMP #$41.
3007  BNE  $300A   If it's not your target key, jump to
                   RTS.
3009  BRK          If it is the target, BRK...
300A  RTS          back to the routine which called
                   this subroutine.
```

However, the above routine *requires* that some key be pressed. It will keep branching back to 3000 until someone touches a key. This is the kind of input you would use when you had printed a menu and wanted the program to pause until a selection was made.

There is, however, a location in zero page, the byte at $D4 on the 128, $CB on the 64, which detects keypresses on the fly. You could LDA $D4:CMP #10:BEQ FOUNDA (FOUNDA is the label you gave a routine that does something whenever the user presses the A). Notice that the code for the letter *A* has a value of ten here. Unlike a JSR $FFE4, the value returned from location $D4 is not regular ASCII. It's a different code, the "keyboard matrix code," and there's no use learning it or having a chart of it. Carriage return is 1, the letter *A* is 10; when no key is pressed, $D4 contains an 88 (the 64's $CB contains, appropriately, a 64). If you need to know sometime what value will be in $D4 for a particular keypress, just look at $D4 via BASIC with this simple program:

```
10 PRINT PEEK(212);:GOTO 10:REM USE PEEK(203) ON
   THE 64
```

and then press the key you're interested in.

Now back to CONT, the matter at hand. The 8502 or 6502 chips place the program counter (plus two) on the stack after a BRK. A close analogy to BASIC is the placement of BRK within ML code to cause a halt to program execution. Then, after examining registers or variables or *buffers* (places that hold input or output before it's received or sent), you can restart your program by using the monitor G (go) command. G is the equivalent of CONT.

DATA

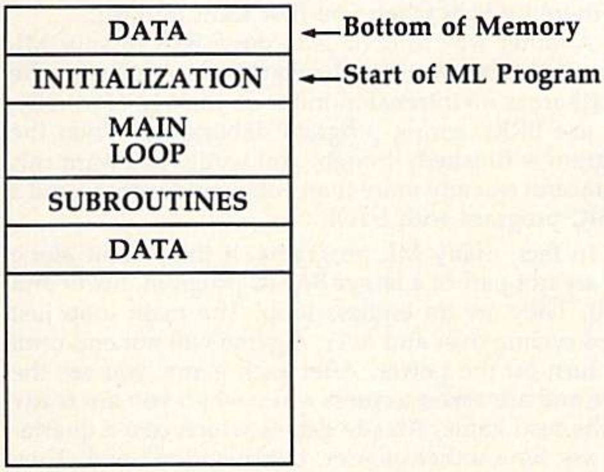
In BASIC, DATA announces that the items following the word DATA are to be considered pieces of information (as opposed to being thought of as parts of the program). That is, the program will probably *use* this data, but the data elements are not BASIC commands.

In ML, such a zone of "nonprogram" is called a *table*. It is unique only in that the program counter never starts trying to run through a table to carry out instructions. This never happens because you never transfer program control, using JMP, JSR, or a branching instruction, to anything within a table. (This is similar to the way that BASIC slides right over DATA lines.) There are no meaningful instructions inside a table.

To keep things simple, tables of data are usually stored together either above or below the program. Usually, tables are stored above, at the very end of the ML program (see figure below).

Tables can hold messages that are to be printed to the screen, hold variables, hold flags (temporary indicators), and so on. If you disassemble your BASIC in ROM, you'll find the words STOP, RUN, LIST, and so forth, gathered together in a table. You can suspect a data table when your disassembler starts giving lots of ??? error messages. It cannot find groups of meaningful opcodes within tables.

Typical ML Program Organization



DIM

With its automatic string handling, array management, and error messages, BASIC makes life easy for the programmer.

The price you pay for this hand holding is that it slows down the program when it's run. In ML, the DIMensioning of space in memory for variables is not explicitly handled by the computer. You must make a note that you are setting aside memory from $6000 to $6500, or whatever, to hold variables. It helps to make a simple map of this "dimensioned" memory so that you know where permanent strings, constants, variable strings, and variables, flags, and so on, are *within* the dimensioned zone. Because this set-aside memory will not contain meaningful ML instructions, it is generally placed at the end of the actual ML program.

A particular chunk of memory (where, and how much, is up to you) is reserved; that's all. There are no ML instructions which set aside the memory. That's why it's best to place tables at the end of your program. This way, they can be enlarged conveniently without affecting any other part of the program.

END

There are several ways to make a graceful exit from ML programs. You can JMP to the "warm start" address ($4003). Or you can go to the "cold start" address ($4000).

If you went into the ML *from* BASIC with a SYS, you can return to BASIC with an RTS. Recall that every JSR matches up with its own RTS. Every time you use a

JSR, it shoves its "return here" address onto the top of the stack. If the computer finds another JSR (before any RTS), it will shove another return address on top of the first one. So, after two JSRs, the stack contains two return addresses. When the first RTS is encountered, the top return address is lifted from the stack and put into the program counter so that the program returns control to the current instruction following the most recent JSR.

When the next RTS is encountered, it pulls *its* appropriate return (waiting for it on the stack), and so on. The effect of a SYS from BASIC is like a JSR from within ML. The return address to the correct spot *within BASIC* is put on the stack. In this way, if you are within ML and there is an RTS (without any preceding JSR), what's on the stack will be a return-to-BASIC address left there by SYS when you first went into ML.

Another way to END is to put a BRK in your ML code. This drops you into the machine's monitor on the 128 (there is no internal monitor on the 64). Normally, you use BRKs during program debugging. When the program is finished, though, you would not want this ungraceful exit any more than you would want to end a BASIC program with STOP.

In fact, many ML programs, if they stand alone and are not part of a larger BASIC program, never end at all. They are an endless loop. The main loop just keeps cycling over and over. A game will not end until you turn off the power. After each game, you see the score and are asked to press a key when you are ready for the next game. Arcade games which cost a quarter will ask for another quarter, but they don't end. They go into "attract mode." The game graphics are left running onscreen to interest new customers.

An ML word processor will cycle through its main loop, waiting for keys to be pressed, words to be written, format or disk instructions to be given. Here, too, it is common to find that the word processor takes over the machine, and you cannot stop it without turning the computer off. Among other things, such an endless loop protects software from being pirated. Since it takes control of the machine, this makes it harder for someone to save it or examine it once it's in RAM. Some such programs are "autobooting" in that they start themselves running as soon as they are loaded into the computer.

BASIC, itself an ML program, also loops endlessly until you power down. When a program is running, all sorts of things are happening. BASIC is an *interpreter*, which means that it must look up each word (like INT) it comes across during a RUN (interpreting, or *translating*, its meaning into machine-understandable JSRs). Then, BASIC executes the correct sequence of ML actions from its collection of routines.

In contrast to BASIC RUNs, BASIC spends 99 percent of its time waiting for you to *program* with it. This waiting for you to press keys is its endless loop, a tight, small loop indeed.

It would look like our "which key is pressed?" routine:

```
2000 LOOP LDA $D4; THE "WHICH KEY IS BEING
     PRESSED" LOCATION
2002 CMP #88; IF 88, KEEP LOOPING
2004 BEQ LOOP
```

If there is an 88 in $D4, this means that no key has been pressed. So, we keep looping until the value in address $D4 is something other than 88. This setup is similar to INPUT in BASIC because not only does it wait until a key is pressed, but it also leaves a unique value of the key in the accumulator when it's finished.

FOR-NEXT

Everyone has had to use delay loops in BASIC (FOR T = 1 TO 1000: NEXT T) which are also tight loops, sometimes called do-nothing loops because nothing happens between the FOR and the NEXT except the passage of time. For example, when you need to let the user read something on the screen, it's sometimes easier just to use a delay loop than to say, "When finished reading, press any key."

In any case, you'll sometimes need to use delay loops in ML just to *slow ML itself down*. In a game, the ball can fly across the screen. It can get so fast, in fact, that you can't see it. It just "appears" when it bounces off a wall. And, of course, you'll need to use loops in many other situations. Loops of all kinds are fundamental programming techniques.

In ML, you don't have that convenient little counter (T in the BASIC FOR-NEXT example above) which decides when to stop the loop. When T becomes 1000, go to the instructions beyond the word NEXT. Again, you must set up and check your *counter variable* by yourself.

If the loop is going to be smaller than 255 cycles, you can use the X register as the counter—Y is saved for the very useful *indirect indexed* addressing. Anyway, by using X, you can count down from 200 to 0:

```
2000  LDX   #200    (or $C8 hex)
2002  DEX
2003  BNE   $2002
```

For loops involving counters larger than 255, you'll need to use two bytes to count down, one going from 255 to 0 and then clicking (like a gear) the other (more significant) byte.

To count to 512:

```
2000  LDA  #$2
2002  STA  $6000   Put the 2 into address 6000, our
                   MSB, most significant byte,
                   counter.
2005  LDX  #$0     Set X to 0 so that its first DEX will
                   make it 255. Further DEXs will
                   count down again to 0, when it
                   will click the MSB down from 2 to
                   1 and then finally to 0.
2007  DEX
2008  BNE  $2007
200A  DEC  $6000   Click the number in address 6000
                   down 1.
200D  BNE  $2007
```

Here we used the X register as the LSB (least significant byte) and address 6000 as the MSB. Why use address 6000? Why not? Use any RAM byte you want that won't interfere with other things going on in the computer. In practice, you'll want to set aside a byte in your tables at the end of your ML program to be sure that it's not going to interfere with something. See DATA above for a discussion of tables.

We could use addresses $FA *and* $FB to hold the MSB/LSB if we wanted. This is commonly useful because then address $FA (or some available, two-byte space in zero page) can be used for LDA ($FA),Y. You would print a message to the screen using the combination of a zero-page counter and LDA (zero-page address),Y.

FOR-NEXT-STEP

Here you would just decrease your counter (usually X or Y) more than once. To create FOR I = 100 TO 1 STEP −2 you could use:

```
2000  LDX  #100
2002  DEX
2003  DEX
2004  BNE  $2002
```

For larger numbers you create a counter which uses two bytes, working together, to keep count of the events. Following our example above for FOR-NEXT, we could translate FOR I = 512 TO 0 STEP −2:

```
2000  LDA  #$2
2002  STA  COUNTER  This is going to hold our MSB.
2005  LDX  #$0      X is holding our LSB.
2007  DEX
2008  DEX           Here we click X down a second
                    time, for −2.
2009  BNE  $2007
200B  DEC  COUNTER
200E  BNE  $2007
COUNTER .BYTE 0; A single byte set aside in our
  Tables
```

To count up, use the CoMPare instruction. FOR I = 1 TO 50 STEP 3:

```
2000  LDX  #$0
2002  INX
2003  INX
2004  INX
2005  CPX  #$50
2007  BCS  $2002
```

For larger STEP sizes, you can use a *nested loop* within the larger one. This would avoid a whole slew of INXs. To write the ML equivalent of FOR I = 1 TO 50 STEP 10:

```
2000  LDX  #$0
2002  LDY  #$0
2004  INY
2005  CPY  #$0A
2007  BNE  $2004
2009  CPX  #$32
200B  BCS  $2002
```

GET

Every computer must have that important "which key is being pressed?" address, where it holds the value of a character typed in from the keyboard. To GET, you create a very small loop which tests this address. See a complete description of this technique under CONT above.

GOSUB

This is nearly identical to the command GOSUB inBASIC. Use JSR $*NNNN* and you will go to a subroutine at address *NNNN* instead of a line number as in BASIC. (*NNNN* just means that you can substitute any hex number for the *NNNN* that you want to. This is a form of math shorthand.)

Parameters are the number or numbers handed to a subroutine to give it information it needs. Quite often, BASIC subroutines work with the variables already established within the BASIC program. In ML, though, managing variables is up to you. Subroutines are useful because they can perform tasks repeatedly without needing to be written into the body of the program each time the task is to be carried out. Beyond this, they can be *generalized* so that a single subroutine can act in a variety of ways, depending upon the variable (the parameter) which is passed to it.

A delay loop to slow up a program could be general in the sense that the amount of delay is handed to the subroutine each time. The delay can, in this way, be of differing durations, depending on what it gets as a parameter from the main routine.

Let's say that we've decided to use address $40 to pass parameters to subroutines. We could pass a delay of five cycles of the loop by:

```
                  2000 LDA #$5
The Main Program  2002 STA $40
                  2004 JSR $5000
                  .
                  .
                  .
                  .
                  5000 DEC $40
                  5002 BEQ $500C If address $40 has
                                 counted all the
                                 way down from 5
                                 to 0, RTS back to
                                 the main program.
                  5004 LDY #$0
                  5006 DEY
The Subroutine    5007 BNE $5006
                  5009 JMP $5000
                  500C RTS
```

A delay which lasted twice as long as the above would merely require a single change to the calling routine: 2000 LDA #$0A.

GOTO

In ML, you use the JMP instruction. JMP is like JSR, except the address you leap away from is not saved anywhere. You jump, but cannot use an RTS to find your way back.

There are two basic kinds of branching in computing. A *conditional* branch would be CMP #0:BEQ 5000. The condition of equality is tested by BEQ, Branch if EQual. BNE tests a condition of inequality, Branch if Not Equal. Likewise, BCC (Branch if Carry is Clear) and the rest of these branches are testing conditions within the program.

GOTO and JMP do not depend on any conditions within the program, so they are *unconditional* branches. The question arises when you use a GOTO: Why did you write a part of your program that you must *always* (unconditionally) jump over? GOTO and JMP are sometimes used to patch up a program, but used without restraint, they can make your program hard to understand later. On the other hand, JMP can many times be the best solution to a programming problem. In fact, it is hard to imagine ML programming without it.

One additional note about JMP: It makes a program nonrelocatable. If you later want to move your whole ML program to a different part of memory, all the JMPs (and JSRs) need to be checked to be sure they aren't pointing to addresses which are no longer correct. (JMP or JSR into your BASIC ROMs will still be the same, but not those which are targeted to addresses *within* the ML program.)

```
2000   JMP   $2005
2003   LDY   #$3
2005   LDA   #$5
```

If you moved this little program up to $5000, everything would survive intact and work correctly except the JMP $2005. It would still say to jump to $2005, but it should say to jump to $5005, after the move. You have to go through with a disassembly and check for all these incorrect JMPs. To make your programs more "relocatable," you can use a special trick with unconditional branching which *will* move without needing to be fixed:

```
2000   LDY     #$0
2002   BEQ     $2005    Since we just loaded Y with a
                        zero, this Branch if EQual to
                        zero instruction will always be
                        true and cause a pseudo-JMP.
2004   NOP
2005   LDA     #$5
```

Your monitor includes a "moveit" routine, invoked with T (Transfer), which will take an ML program and relocate it somewhere else in memory for you. You can go into the monitor and type T 2000 2006 5000 (you give the monitor these numbers in hex). The third number is the target address. The first and second are the start and end of the program you want to move.

The best solution to relocatability, however, is an assembler like LADS. (*Editor's note: LADS is a label-based assembler included with* 128 Machine Language For Beginners.) With it, you never JMP to actual addresses; rather, you JMP or JSR or branch to labels. This way, relocating your program couldn't be simpler. You just change the start address with *= and reassemble. Everything is taken care of and the program reassembles to the new location flawlessly. With LADS, the example above is written like this:

```
100 JMP NEXTROUTINE
110 LDY #3
120 NEXTROUTINE LDA #5
```

(The numbers at the left are not addresses; they are line numbers for your convenience when writing the program, and they have no effect on the resulting ML code after assembly.)

IF-THEN

This familiar and fundamental computing structure is accomplished in ML by combining the CMP instruction with BNE or any other conditional branch: BEQ, BCC, and so forth. Sometimes, the IF half isn't even necessary. Here's how it would look:

```
2000   LDA   $57     What's in address $57?
2002   CMP   #$0F    Is it $0F, 15 decimal?
2004   BEQ   $200D   IF it is, branch up to $200D.
2006   LDA   #$0A    Or ELSE, put a $0A, 10 decimal,
                     into address $57.
2008   STA   $57
200A   JMP   $2011   And jump over the THEN part.
200D   LDA   #$14    THEN, put a $14, 20 decimal, into
                     address $57.
200F   STA   $57
2011                 Continue with the program....
```

Often, though, your flags are already set by an action, making the CMP unnecessary. For example, if you want to branch to $200D if the number in address $57 is zero, just LDA $57:BEQ $200D. This works because the act of loading the accumulator will affect the status register flags. You don't need to CMP #0 because the zero flag will be set if a zero was just loaded into the accumulator. It won't hurt anything to use a CMP, but you'll find many cases in ML programming where you can shorten and simplify your coding if you wish to. As you gain experience, you will see these patterns and learn what affects the status register flags.

INPUT

This is a series of GETs, echoed to the screen as they are typed in, which end when the typist hits the RETURN key. The reason for the echo (the symbol for each key typed is reproduced on the screen) is that few people enjoy typing without seeing what they've typed. This also allows for error correction using cursor control keys or DELETE and INSERT keys. To handle all of these actions, an INPUT routine must be fairly complicated. We don't want, for example, the DELETE to become a character within the string. We want it to act immediately on the string being entered during the INPUT, to erase a mistake.

Our INPUT routine must also be smart enough to know what to add to the string and what keys are intended only to modify it. Here is the basis for constructing your own ML INPUT. It simply receives a character from the keyboard, prints it to the screen, and ends when the RETURN key is pressed. We'll write this INPUT as a subroutine. That simply means that when the 13 (ASCII for carriage return) is encountered, we'll perform an RTS back to a point just following the main program address which JSRed to our INPUT routine. Let's do it in the LADS source code format, with line numbers instead of addresses:

```
10 *= $B00
20 .S
30 .O
40 LOOP JSR $FFE4:BEQ LOOP;   If we got a zero, no
                              key had been pressed.
50 JSR $FFD2;                 Print the character to
                              the screen.
60 CMP #13;                   Is it a carriage return?
70 BNE LOOP;                  If not, return for more
                              keypresses.
80 RTS;                       Otherwise return to the
                              calling routine.
```

If you try this out, you'll notice that even the cursor keys and delete, screen clear, and so forth, work correctly. This is because when you JSR $FFD2 (PRINT), it is just as if you printed any character from BASIC (with cursor control codes embedded in a string). However, if we wished, we could make this INPUT routine much more complex. As it stands, it will hold the string on the screen only. To save the string, you would need to store it in some buffer of yours in addition to its appearing on the screen. Nonetheless, if you're going to store the string into some safe location where you are keeping string variables, you'll need to refuse to store such things as the delete character or your stored string will be corrupted because it will include that delete. Or you might want to prevent the user from hitting a key like carriage return. In that case, just CMP #13:BEQ LOOP so that nothing is echoed to the screen or stored in your string when the user tries to enter that particular key.

The great freedom you have with ML is that you can redefine anything you want. You can *softkey*: define a key's meaning via software (have any key perform any task you want). You might even decide to use the $ key to DELETE.

Along with this freedom goes the responsibility for organizing, writing, and debugging these routines.

LET

Although this word is still available on most BASICs, it is a holdover from the early days of computing. It is supposed to remind you that statements like LET NAME = NAME + 4 is an *assignment* of a value to a variable, not an algebraic equation. The two numbers on either side of the equal sign, in BASIC, are not intended to be equal in the algebraic sense. Most people write NAME = NAME + 4 without using LET. The function of LET applies, though, to ML as well as to BASIC: Values must be assigned to variables.

In the 128, for example, where the RAM bank can change depending on how you configure the computer, there has to be a place where we can find out which bank is the current bank. Address $FF00 holds this information.

Likewise, a program will sometimes require that you *assign* meanings to string variables, counters, and the like. This can be part of the initialization process, the tasks performed before the real program, your main routine, gets started. Or it can happen during the execution of the main loop. In either case, there has to be an ML way to establish, to *assign*, variables. This also means that you must have zones of memory set aside to hold these variables unless, like the bank-switching location, the computer has already defined a variable's address. Normally, you will store your variables as a group at the end of an ML program.

For strings, you can think of LET as the establishment of a location in memory. In our INPUT example above, we might have included an instruction which would have sent the characters from the keyboard to a table of strings as well as echoing them to the screen. If so, there would have to be a way of managing these strings.

In general, you will probably find that you program in ML using somewhat fewer variables than in BASIC. There are three reasons for this:

1. You will probably not write many programs in ML like databases where you manipulate hundreds of names, addresses, and so forth. It might be somewhat inefficient to create an entire database management program, an inventory program for example, in ML. Keeping track of the variables would require careful programming.

The value of ML is its speed of execution, but its drawback is that it requires more precise programming and, at least for beginners, can take more time to write. So, for an inventory program, you could write the bulk of the program in BASIC and simply attach ML routines for *sorting* and *searching* tasks within the program where high speeds are desirable.

2. The variables in ML are often handled within a series of instructions (not held elsewhere as BASIC variables are). FOR I = 1 TO 10 : NEXT I becomes, in ML, LDY #1:INY:CPY #10:BNE.

Here, the BASIC variable is counted for you and stored outside the body of the program. The ML "variable," though, is counted by the program itself. ML has no *interpreter* which handles such things. If you want a loop, you must construct all of its components yourself.

3. In BASIC, it is tempting to assign values to variables at the start of the program and then to refer to them later by their variable names, as in 10 BALL = 79. Then, anytime you want to PRINT the BALL to the screen, you could say, PRINT CHR$(BALL). Alternatively, you might define it this way in BASIC: 10 BALL$ = "O". In either case, your program will later refer to the word BALL. In this example we are assuming that the number 207 will place a ball character on your screen (the letter *O).*

In ML we can use variable names precisely the same way if we are programming with an advanced assembler like LADS. However, with an elementary assembler, you will just LDA #207, STA (screen position) each time. Some people like to put the 207 into their zone of variables (that arbitrary area of memory set up at the end of a program to hold tables, counters, and important addresses). They can pull it out of that zone whenever it's needed. That is somewhat cumbersome, though, and slower. You would LDA 1015, STA (screen position), assuming you had put a 207 into this "ball" address, 1015, earlier.

Obviously, a value like BALL will always remain the same throughout a program. The ball will look like a ball in your game, whatever else happens. So, it's not a true variable; it does not *vary*. It is constant. A true variable *must* be located in your "zone of variables," your variable *table*.

It cannot be part of the body of your program itself (as in LDA #207) because it will change. You don't know when writing your program what the variable will be. So you can't use *immediate mode* addressing because it might not be a #207. You have to LDA 1015 from within your table of variables.

Elsewhere in the program you'll have programmed one or more STA 1015 or INC 1015 instruction, or some other manipulation of this address, which keeps updating this variable. In effect, ML makes you responsible for setting aside areas which are safe to hold variables if you are using the monitor assembler. What's more, you have to remember the addresses and update the variables in those addresses whenever necessary. This is why it is so useful to keep a piece of paper next to you when you are writing ML using the monitor. The paper lists the start and end addresses of the zone of variables, the table. You write down the specific address of each variable as you write your program.

LIST

This is done via a *disassembler*. It will not have line numbers (though, again, advanced assembler packages like LADS do allow line numbers). You will see the address of each instruction in memory. You can look over your work and plan debugging strategies, where to set BRKs into problem areas, and so on. The most common way to list and check your work, however, is to read over the *source code*. This does not require a disassembler. You write LADS source code as if it were a BASIC program and, thus, can LIST it and modify it as if it were a BASIC program. There is a subtle difference between studying source code and studying object code (via disassembly). The former is most useful for making modifications and for locating the more obvious bugs. The latter is useful when you need to patiently track down those last few stubborn bugs that no amount of reading over the source code seems to uncover.

LOAD

The method of saving and loading an ML program varies from computer to computer. You have two options: loading from within the monitor or from BASIC. When you finish working on a program, or a piece of a program, on the mini-assembler, you will know the starting and ending addresses of your work. Using these, you can save to disk or tape using the S monitor command (see SAVE below). To load, the simplest way is just to L "FILENAME",1 (for tape) or ,8 (for disk). You can also load ML when you're in BASIC mode by BLOAD. With both the monitor's L and BASIC's BLOAD commands, you can reassign your ML routine to a different target address (see your manual). However, this will not adjust the JSRs, and so on, so you haven't really relocated the program, and it probably would not run at the new location. To truly relocate it, you need to change the start address *= and reassemble it with LADS. However, loading in a version of your ML program to a different location with the L command and then loading in another version in its normal location does allow you to compare them with the monitor's C command.

NEW

In Microsoft BASIC, this has the effect of resetting some pointers which make the machine think that you are going to start over again. The next program line you type in will be put at the "start-of-a-BASIC-program" area of memory. Some computers, the Atari for example, even *wash* memory by filling it with zeros. There is no special command in ML for NEWing an area of memory, though the monitor has a "fill memory" option which will fill an area of memory as large as you want with whatever value you choose.

The reason that NEW is not found in ML is that you do not always write your programs in the same area of memory as you do in BASIC, building up from some predictable address. You might have a subroutine floating up in high memory, another way down low, your table of variables at the end, and your main program in the middle. Or you might not. We've been using $2000 as our starting address for many of the examples in this book and $5000 for subroutines, but this is entirely arbitrary.

To "NEW" in ML, just start assembling over the old program.

Alternatively, you could just turn the power off and then back on again. This would, however, have the disadvantage of wiping out your assembler along with your program.

ON-GOSUB

In BASIC, you are expecting to test values from among a group of numbers: 1, 2, 3, 4, 5, ... The value of X must fall within this narrow range: ON X GOSUB 100, 200, 300, ... (X must be 1 or 2 or 3 here). In other words, you could not conveniently test for widely separated values of X (18, 55, 220).

Using CMP, you can perform a *multiple branch* test:

```
2000  LDA  $96     Get a value, perhaps input from
                   the keyboard.
2002  CMP  #$50    Decimal 80
2004  BNE  $2009
2006  JSR  $5000   Where you would print "hot,"
                   following our example of CASE.
2009  CMP  #$64    Decimal 100
200B  BNE  $2011
200D  JSR  $5020   Print "very hot"
2010  CMP  #$78    Decimal 120
2012  BNE  $2017
2014  JSR  $5030   Print "intolerable"
```

This illustrates one way that bugs get into ML—by not cleanly entering and leaving subroutines. The potential problem here is triggering the CMPs more than once. Since you are JSRing and then will be RTSing back to *within* the multiple branch test above, you will have to be sure that the subroutines up at $5000 do not change the value of the accumulator. If the accumulator started out with a value of $50 and, somehow, the subroutine at $5000 left a $64 in the accumulator, you would print "hot" and then also print "very hot." One way around this would be to put a zero into the accumulator before returning from each of the subroutines (LDA #$0). This assumes that none of your tests, none of your cases, responds to a zero.

ON-GOTO

This is more common in ML than the ON-GOSUB structure above. It eliminates the need to worry about what is in the accumulator when you return from the subroutines. Instead of RTSing back, you jump back, *following all the branch tests.*

```
2000  LDA  $96
2002  CMP  #$50
2004  BNE  $2009
2006  JMP  $5000   Print "hot"
2009  CMP  #$64
200B  BNE  $2010
200D  JMP  $5020   Print "very hot"
2010  CMP  #$78
2012  BNE  $2017
2014  JMP  $5030   Print "intolerable"
```

```
2017              All the subroutines JMP $2017
                  when they finish.
```

Instead of RTS, each of the subroutines will JMP back to $2017, which lets the program continue without accidentally "triggering" one of the other tests with something left in the accumulator during the execution of one of the subroutines.

PRINT

You *could* print out a message in the following way:

```
2000  LDY  #$0
2002  LDA  #72       The letter H
2004  STA  $0400,Y   An address on the screen
2007  INY
2008  LDA  #69       The letter E
200A  STA  $0400,Y
200D  INY
200E  LDA  #76       The letter L
2010  STA  $0400,Y
2013  INY
2014  LDA  #76       The letter L
2016  STA  $0400,Y
2019  INY
201A  LDA  #79       The letter O
201C  STA  $0400,Y
```

But this is clearly a clumsy, memory-hungry way to go about it. In fact, it would be absurd to print out a long message in this fashion. The most common ML method involves putting message strings into a data table and ending each message with a zero. Zero is never a printing character in computers; to print the *number* zero, you use 176: LDA #$30, STA $0400. So, true zero (not the code for the character 0) can be used as a *delimiter* (a signal that something has ended) to let the printing routine know that you've finished the message. In a data table, we first put in the message "hello":

```
1000  $48  H
1001  $45  E
1002  $4C  L
1003  $4C  L
1004  $4F  O
1005  $00       The delimiter
1006  $48  H
1007  $49  I    Another message
1008  $00       Another delimiter
```

Such a message table can be as long as you need; it holds all your messages and they can be used again and again:

```
2000  LDY  #$0
2002  LDA  $1000,Y
2005  BEQ  $200F     If the zero flag is set, it must mean
                     that we've reached the delimiter,
                     so we branch out of this printing
                     routine.
2007  STA  $0400,Y   Put it on the screen.
200A  INY
200B  JMP  $2002     Go back and get the next letter in
                     the message.
200F                 Continue with the program.
```

Had we wanted to print HI, the only change necessary would have been to put $1006 into the LDA at address $2003. To change the location on the screen that the message starts printing, we could just put some other address into $2008. The message table, then, is just a mass of words, separated by zeros, in RAM memory.

The fastest way to print to the screen, especially if your program will be doing a lot of printing, is to create a subroutine which will print any of your messages. It can use some bytes in zero page (addresses 0–255) to hold the location of the message within your table of data.

To put an address into zero page, you will need to put it into two bytes. Addresses are too big to fit into one byte. With LADS, you can use the #< and #> pseudo-ops to extract the LSB and MSB of a label and thus store the address of your message into a zero page pointer:

```
 10 MSGADDRESS = 56
 20 SCREEN = $0400
100 LDA #<MESSAGE:STA MSGADDRESS; set up
    pointer
110 LDA #>MESSAGE:STA MSGADDRESS+1
120 JSR PRINTMSG; go to universal print subroutine
500 PRINTMSG LDY #0:LOOP LDA
    (MSGADDRESS),Y:BEQ END:STA SCREEN,Y
510 STA SCREEN,Y:INY:JMP LOOP
520 END RTS
```

First, you split the hex number in two. The left two digits, $10, are the MSB (most significant byte) and the right digits, $00, make up the LSB (least significant byte). If you are going to put this target address into zero page at 56 (decimal):

```
2000  LDA  #$00      LSB
2002  STA  $56
2004  LDA  #$10      MSB
2006  STA  $57
2008  JSR  $5000     Printout subroutine
5000  LDY  #$0
5002  LDA  ($56)Y
5004  BEQ  $5013     If zero, return from subroutine...
5006  STA  $0400,Y   to screen.
5009  INY
500A  JMP  $5002
500D  RTS
```

One drawback to this PRINT subroutine we've constructed is that it will always print any messages to the same place on the screen. That $0400 is frozen into your subroutine. Solution? Use another zero page pair of bytes to hold the screen address. Then, your calling routine sets up the message address as above, but also goes on to specify a screen address as well.

The 128's screen, in 40-column mode, starts at $0400 (1024 decimal), so you will want to put 0 and 4 into the LSB and MSB respectively for your screen pointer.

```
2000  LDA  #$00      LSB
2002  STA  $56       Set up message address
2004  LDA  #$10      MSB
2006  STA  $57
2008  LDA  #$0       LSB

200A  STA  $58       We'll just use the next two bytes in
                     zero page above our message
                     address for the screen address.
200C  LDA  #$4       MSB
200E  STA  $59
2010  JSR  $5000
5000  LDY  #$0
5002  LDA  ($56)Y
```

```
5004   BEQ   $500D   If zero, return from subroutine...
5006   STA   ($58),Y  to screen.
5009   INY
500A   JMP   $5002
500D   RTS
```

The easiest way to print messages to particular places on the screen, however, is to use the 128's built-in BASIC PRINT routine to send the characters, one by one, each to the next cursor position onscreen. The built-in routine updates and keeps track of the current cursor position for you. So, you can get around having to keep a screen pointer in zero page this way. In the example immediately above, just replace line 5006 with JSR $FFD2 (the 128's PRINT routine) and remove lines 2008–200E.

READ

There is no reason for a *read*ing of data in ML. Variables are not placed into "DATA statements." They are entered into a table when you are programming. The purpose of READ, in BASIC, is to assign variable names to raw data, or to take a group of data and move it somewhere, or to manipulate it into an array of variables. These things are handled by you, not by the computer, in ML programming.

If you need to access a piece of information, *you* set up the addresses of the datum and the target address to which you are moving it. As always, in ML you are expected to keep track of the locations of your variables. This pad of paper is always next to you as you program in ML. It would seem that you would need many notes, but in practice an average program of, say, 1000 bytes could be mapped out and commented on, using only one sheet.

Alternatively, with more sophisticated assemblers like LADS, the labels themselves within the program will keep track of things for you, and embedded comments serve to remind you of the use and function of all data.

REM

You do this on a pad of paper, too, when working with a simple assembler. If you want to comment or make notes about your program (and it can be a necessary, valuable explanation of what's going on), you can disassemble some ML code like a BASIC listing. If you have a printer, you can make notes on the printed disassembly. If you don't use a printer, make notes on your pad to explain the purpose of each subroutine, the parameters it expects to get, and the results or changes it effects.

The more sophisticated assemblers like LADS will permit comments within the source code. As you program, you can include REMarks by typing a semicolon, which is a signal to the assembler to ignore the REMarks when it is assembling your program. In these assemblers, you are working much closer to the way you work in BASIC. Your REMarks remain part of the source program, and can be listed out and studied.

RETURN

RTS works the same way that RETURN does in BASIC: It takes you back to *just after* the JSR (GOSUB) that sent control of the program away from the main program and into a subroutine. JSR pushes, onto the stack, the address which immediately follows the JSR itself. That address, then, sits on the stack, waiting until the next RTS is encountered. When an RTS occurs, the address is pulled from the stack and placed into the *program counter*. This has the effect of transferring program control back to the instruction just after the JSR.

RUN

There are several ways to start an ML program. If you are taking off into ML from BASIC, you just SYS to it by giving its address (in decimal) as the argument of the SYS. This acts just like JSR and will return control to BASIC, just as RETURN would, when there is an unmatched RTS in the ML program. By *unmatched*, we mean the first RTS which is not part of a JSR/RTS pair. SYS can take you into ML either in *immediate mode* (directly from the keyboard) or from within a BASIC program as one of the BASIC commands.

If you need to "pass" information from BASIC to ML, it is easiest to use integer numbers and just POKE them into some predetermined ML variable zone that you've set aside and noted on your notepad. Then just SYS to your ML routine, which will look into the set-aside, POKEd area when it needs the values from BASIC.

If you are not going between BASIC and ML, you can start (RUN) your ML program from within the built-in monitor. To enter the monitor, press f8. To run an ML program from within the monitor, type G 2000 (that's address 8192 in decimal; this presumes that you've either loaded in your ML program at that address or have just assembled one there).

The 128 expects to encounter a BRK instruction to end the run and return control to the monitor.

SAVE

When you save a BASIC program, the computer automatically handles it. The starting address and the ending address of your program are calculated for you. In ML, you must know the start and end addresses. From the monitor, you type S, then the name of your program, then 8 for disk or 1 for tape, the starting address, and the ending address. All these items are separated by commas:

```
S "FILENAME",8,2000,2010
```

(Note that these addresses are in hex. The addresses are 8192 and 8208, in decimal, but you must use hex from the monitor unless you specify otherwise. For more information about BSAVE and BLOAD, the ML save and load routines in BASIC, please see your *User's Guide*.)

STOP

BRK (or an RTS with no preceding JSR) throws you back into the monitor mode after running an ML program. BRK is most often used for debugging programs because you can set "breakpoints" in the same way that you would use STOP to examine variables when debugging a BASIC program.

SYS

This is BASIC's way of using a piece of ML code, an ML routine, as a subroutine. The only difference between SYS and GOSUB is that the computer is alerted to the fact that it needs to switch mental gears: The next series of instructions will be ML. In other words, the computer shouldn't try to interpret what it finds at the SYS address as more BASIC instructions. Later, when it comes upon an RTS instruction in the ML program which was not matched by a previous JSR instruction, it will then revert to the BASIC program and pick up where it left off, following the SYS instruction.

There are times when you want to write in ML and use it as a subroutine for a BASIC program. This can greatly speed up the execution of the BASIC program. To put an ML program in RAM where it will be safe from BASIC's dynamic variable storage (where it won't be overwritten by BASIC), you lower the "top-of-memory" pointer ($39,3A) to create some space in high RAM of which the computer is "unaware." This pointer contains the address (in the usual LSB,MSB format discussed earlier) beyond which BASIC is forbidden to intrude. If you're going to use only one page of memory (256 bytes), just DEC #3A which has the effect of making it point 256 bytes lower than it normally would. This pointer affects bank 1.

After resetting this pointer, you are free to load in your ML program into the now-safe RAM between where the pointer points and the true highest RAM byte in your computer.

Short ML routines can always be stored in the page between $B00 and $BFF without any special preliminaries.

String Handling

ASC

In BASIC, this will give you the number of the ASCII code which stands for the character you are testing. ?ASC("A") will result in a 65 being displayed. There is never any need for this in ML. If you are manipulating the character *A* in ML, you *are using ASCII already*. In other words, the letter *A is* 65 in ML programming. The Commodore ASCII code isn't standard ASCII; it stores character symbols in some nonstandard ways, so you will need to write a special program to be able to translate to standard ASCII if you are using a modem or some other peripheral which uses true ASCII.

CHR$

This is most useful in BASIC to let you use characters which cannot be represented within normal strings, will not show up on your screen, or cannot be typed from the keyboard.

For example, if you have a printer attached to your computer, you could send CHR$(13) to it, and it would perform a carriage return. The correct numbers which accomplish various things sometimes differ, though decimal 13—an ASCII code standard—is nearly universally recognized as carriage return, and the 128 uses this convention, too.

Or, you could send the combination CHR$(27) CHR$(8), and the printer would backspace.

There is no real use for CHR$ within ML. If you want to specify a carriage return, just LDA #13. In ML, you are not limited to the character values which can appear onscreen or within strings. Any value can be dealt with directly.

LEFT$

As usual in ML, *you* are in charge of manipulating data. Here's one way to extract a certain "substring" from the left side of a string as in the BASIC statement LEFT$(X$,5):

```
2000  LDY  #$5
2002  LDX  #$0       Use X as the offset for buffer
                     storage.
2004  LDA  $1000,Y   The location of X$.
2007  STA  $4000,X   The "buffer," or temporary
                     storage area, for the substring.
200A  INX
200B  DEY
200C  BNE  $2004
```

LEN

In some cases, you will already know the length of a string in ML. One of the ways to store and manipulate strings is to know beforehand the length and address of a string. Then you could use the subroutine given for LEFT$, above. More commonly, though, you will store your strings with delimiters (zeros) at the end of each string. To find out the length of a certain string:

```
2000  LDY  #$0
2002  LDA  $1000,Y   The address of the string you are
                     testing.
2003  BEQ  $2009     Remember, if you LDA a zero,
                     the zero flag is set. So you don't
                     really need to use a CMP #0 here
                     to test whether you've loaded the
                     zero delimiter.
2005  INY
2006  BNE  $2002     We are not using a JMP here be-
                     cause we assume that all your
                     strings are less than 256 charac-
                     ters long.
2008  BRK            If we still haven't found a zero
                     after 256 INYs, we avoid an end-
                     less loop by just BRKing out of
                     the subroutine.
2009  DEY            The LENgth of the string is now
                     in the Y register.
```

We had to DEY at the end because the final INY picked up the zero delimiter. So, the true count of the LENgth of the string is one less than Y shows, and we must DEY one time to make this adjustment.

MID$

To extract a substring which starts at the fourth character from within the string and is five characters long—MID$(X$,4,5):

```
2000  LDY  #$5       The size of the substring we're
                     after.
2002  LDX  #$0       X is the offset for storing the
                     substring.
2004  LDA  $1003,Y   To start at the fourth character
                     from within the X$ located at
                     $1000, simply add three to that
                     address. Instead of starting our
                     LDA,Y at $1000, skip to $1003.
                     This is because the first character
                     is not in position 1. Rather, it is
                     at the zeroth position, at $1000.
```

```
2007  STA  $4000,X   The temporary buffer to hold the
                     substring.
200A  INX
200B  DEY
200C  BNE  $2004
```

RIGHT$

This, too, is complicated because normally we do not know the LENgth of a given string. To find RIGHT$ (X$,5) if X$ starts at $1000, we should find the LEN first and then move the substring to our holding zone (buffer) at $4000:

```
2000  LDY  #$0
2002  LDX  #$0
2004  LDA  $1000,Y
2007  BEQ  $200D     The delimiting zero is found.
2009  INY
200A  JMP  $2004
200D  TYA            Put LEN into A so that we can
                     subtract the substring size from it.
200E  SEC            Always set carry before any
                     subtraction.
200F  SBC  #$5       Subtract the size of the substring
                     you want to extract.
2011  TAY            Put the offset back into Y, now
                     adjusted to point to five charac-
                     ters from the end of X$.
2012  LDA  $1000,Y
2015  BEQ  $201E     We found the delimiter, so end.
2017  STA  $4000,X
201A  INX
201B  DEY
201C  BNE  $2012
201E  RTS
```

TAB

This formatting instruction moves you to a specified column on a given line. TAB 10 moves you ten spaces from the left side of the screen.

In ML, you have more direct control over what happens: You would just add or subtract the amount you want to TAB over to. If you were printing to the screen and wanted ten spaces between A and B so it looked like

```
A          B
```

you could write:

```
2000  LDA  #$41      A
2002  STA  $0400     Screen RAM address
2005  LDA  #$42      B
2007  STA  $040A     You've added ten to the target
                     address.
```

Alternatively, you could add ten to the Y offset (this is LADS format):

```
 10 SCREEN = $0400
100 LDY #0:LDA #"A:STA SCREEN,Y:LDY #10:LDA
    #"B:STA SCREEN,Y
```

As an example, we are writing to the screen here, but in practice, you would print to the screen using $FFD2 as described below. The examples above, using Y as an offset, are more applicable to storing, say, items in a database or printing hardcopy.

Nonetheless, if you are printing out many columns of numbers and need a subroutine to space your printout correctly, you might want to use a subroutine which will add ten to the Y offset each time you call the subroutine:

```
5000  TYA
5001  CLC
5002  ADC #10
5004  TAY
5005  RTS
```

This subroutine directly adds ten to the Y register whenever you JSR $5000. However, it's more typical to rely on $FFD2 for screen printing since it will keep track of the cursor position for you. Just LDA with whatever character you want printed and then JSR $FFD2, and it will be printed at the next available space.

You can see that moving over ten spaces could be accomplished by LDA #32:JSR $FFD2 performed ten times. The 32 is the blank character. However, here, too, there is a more practical method.

Anything you can print from BASIC you can print from ML. So, all the cursor control characters can be printed—CLR screen, backspace, anything. Most control characters can be entered into LADS directly by typing #"*c* where *c* is the control code you desire:

```
5000  LDA  #"c
5001  JSR  $FFD2
```

Alternatively, you can put the actual Commodore ASCII value into the accumulator prior to JSR $FFD2. One way to find out the ASCII value to, for example, clear the screen, would be to go to BASIC and type CHR$(" ") to get it.

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

Each month, COMPUTE!'s GAZETTE publishes programs for the Commodore 128, 64, Plus/4, 16, and VIC-20. 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 back on, erasing whatever was in memory. So be sure to *save a copy of your 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, A̲ 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 A̲}), 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	S
{UP}	SHIFT	↑ CRSR ↓	●
{DOWN}		↑ CRSR ↓	Q
{LEFT}	SHIFT	←CRSR→	▌
{RIGHT}		←CRSR→	]
{RVS}	CTRL	9	R
{OFF}	CTRL	0	■
{BLK}	CTRL	1	■
{WHT}	CTRL	2	E
{RED}	CTRL	3	£
{CYN}	CTRL	4	◤

When You Read:	Press:		See:
{PUR}	CTRL	5	▒
{GRN}	CTRL	6	↑
{BLU}	CTRL	7	←
{YEL}	CTRL	8	π
{ 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	↑	π

For Commodore 64 Only

⁅ 1 ⁆	COMMODORE	1	♣
⁅ 2 ⁆	COMMODORE	2	◜
⁅ 3 ⁆	COMMODORE	3	☒
⁅ 4 ⁆	COMMODORE	4	▣
⁅ 5 ⁆	COMMODORE	5	┼
⁅ 6 ⁆	COMMODORE	6	▌
⁅ 7 ⁆	COMMODORE	7	◆
⁅ 8 ⁆	COMMODORE	8	▚

The Automatic Proofreader

Philip I. Nelson, Assistant Editor

"The Automatic Proofreader" helps you type in program listings for the 128, 64, Plus/4, 16, and VIC-20 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, 65526 for the Plus/4 and 16, and 64802 for the VIC). 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"
3Ø IF VEC=5Ø556 THEN PRINT "VI
   C-2Ø"
4Ø IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
5Ø IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
6Ø SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
7Ø FOR J=Ø TO 166:READ BYT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +BYT:NEXT
8Ø IF CHK<>2Ø57Ø THEN PRINT "*
   ERROR* CHECK TYPING IN DATA
    STATEMENTS":END
9Ø FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*HB)
1ØØ CHK=CHK+RF+LF+HF:POKE SA+L
   F,LB:POKE SA+HF,HB:NEXT
11Ø IF CHK<>22Ø54 THEN PRINT "
   *ERROR* RELOAD PROGRAM AND
   {SPACE}CHECK FINAL LINE":EN
   D
12Ø POKE SA+149,PEEK(772):POKE
    SA+15Ø,PEEK(773)
13Ø IF VEC=17165 THEN POKE SA+
   14,22:POKE SA+18,23:POKESA+
   29,224:POKESA+139,224
14Ø PRINT CHR$(147);CHR$(17);"
   PROOFREADER ACTIVE":SYS SA
15Ø POKE HI,PEEK(HI)+1:POKE (P
   EEK(LO)+256*PEEK(HI))-1,Ø:N
   EW
16Ø DATA 12Ø,169,73,141,4,3,16
   9,3,141,5,3
17Ø DATA 88,96,165,2Ø,133,167,
   165,21,133,168,169
18Ø DATA Ø,141,Ø,255,162,31,18
   1,199,157,227,3
19Ø DATA 2Ø2,16,248,169,19,32,
   21Ø,255,169,18,32
2ØØ DATA 21Ø,255,16Ø,Ø,132,18Ø
   ,132,176,136,23Ø,18Ø
21Ø DATA 2ØØ,185,Ø,2,24Ø,46,2Ø
   1,34,2Ø8,8,72
22Ø DATA 165,176,73,255,133,17
   6,1Ø4,72,2Ø1,32,2Ø8
23Ø DATA 7,165,176,2Ø8,3,1Ø4,2
   Ø8,226,1Ø4,166,18Ø
24Ø DATA 24,165,167,121,Ø,2,13
   3,167,165,168,1Ø5
25Ø DATA Ø,133,168,2Ø2,2Ø8,239
   ,24Ø,2Ø2,165,167,69
26Ø DATA 168,72,41,15,168,185,
   211,3,32,21Ø,255
27Ø DATA 1Ø4,74,74,74,74,168,1
   85,211,3,32,21Ø
28Ø DATA 255,162,31,189,227,3,
   149,199,2Ø2,16,248
29Ø DATA 169,146,32,21Ø,255,76
   ,86,137,65,66,67
3ØØ DATA 68,69,7Ø,71,72,74,75,
   77,8Ø,81,82,83,88
31Ø DATA 13,2,7,167,31,32,151,
   116,117,151,128,129,167,136
   ,137
```

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.

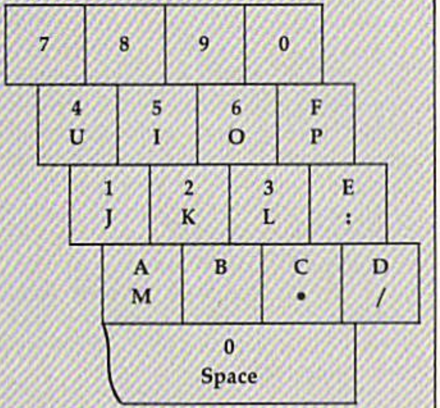
Invalid Characters Banned

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.

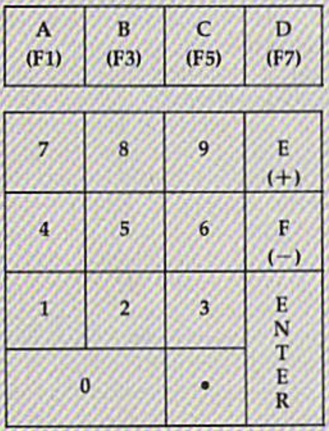
Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. To simplify typing, 128 MLX redefines the function keys and + and − keys on the numeric keypad so that you can enter data one-handed. (The 64 version incorporates 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 below show the keypad configurations for each version.

MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

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
        Ø
GF 19Ø  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$,1)<>"Y"TH
        EN22Ø
PG 21Ø  PRINT"{2 DOWN}{BLU}WORK
        ING...";:FORI=BS TO BS+
        EA-SA+7:POKE I,Ø:NEXT:P
        RINT"DONE"
DR 22Ø  PRINTTAB(1Ø)"{2 DOWN}
        {BLK}{RVS} MLX COMMAND
        {SPACE}MENU {DOWN}§4§":
        PRINT T$"{RVS}E{OFF}NTE
        R DATA"
BD 23Ø  PRINT T$"{RVS}D{OFF}ISP
        LAY DATA":PRINT T$"
        {RVS}L{OFF}OAD FILE"
JS 24Ø  PRINT T$"{RVS}S{OFF}AVE
         FILE":PRINT T$"{RVS}Q
        {OFF}UIT{2 DOWN}{BLK}"
JH 25Ø  GET A$:IF A$=N$ THEN25Ø
HK 26Ø  A=Ø:FOR I=1 TO 5:IF A$=
        MID$("EDLSQ",I,1)THEN A
        =I:I=5
FD 27Ø  NEXT:ON A GOTO42Ø,61Ø,6
        9Ø,7ØØ,28Ø:GOSUB1Ø6Ø:GO
        TO25Ø
EJ 28Ø  PRINT"{RVS} QUIT ":INPU
        T"{DOWN}§4§ARE YOU SURE
         [Y/N]";A$:IF LEFT$(A$,
        1)<>"Y"THEN22Ø
EM 29Ø  POKE SD+24,Ø:END
JX 3ØØ  IN$=N$:AD=Ø:INPUTIN$:IF
        LEN(IN$)<>4THENRETURN
KF 31Ø  B$=IN$:GOSUB32Ø:AD=A:B$
        =MID$(IN$,3):GOSUB32Ø:A
        D=AD*256+A:RETURN
PP 32Ø  A=Ø:FOR J=1 TO 2:A$=MID
        $(B$,J,1):B=ASC(A$)-C4+
        (A$>"@")*C7:A=A*C6+B
JA 33Ø  IF B<Ø OR B>15 THEN AD=
        Ø:A=-1:J=2
GX 34Ø  NEXT:RETURN
CH 35Ø  B=INT(A/C6):PRINT MID$(
        H$,B+1,1);:B=A-B*C6:PRI
        NT MID$(H$,B+1,1);:RETU
        RN
RR 36Ø  A=INT(AD/Z6):GOSUB35Ø:A
        =AD-A*Z6:GOSUB35Ø:PRINT
        ":";
BE 37Ø  CK=INT(AD/Z6):CK=AD-Z4*
        CK+Z5*(CK>Z7):GOTO39Ø
PX 38Ø  CK=CK*Z2+Z5*(CK>Z7)+A
JC 39Ø  CK=CK+Z5*(CK>Z5):RETURN
QS 4ØØ  PRINT"{DOWN}STARTING AT
        §4§";:GOSUB3ØØ:IF IN$<>
        N$ THEN GOSUB1Ø3Ø:IF F
        {SPACE}THEN4ØØ
EX 41Ø  RETURN
HD 42Ø  PRINT"{RVS} ENTER DATA
        {SPACE}":GOSUB4ØØ:IF IN
        $=N$ THEN22Ø
JK 43Ø  OPEN3,3:PRINT
SK 44Ø  POKE198,Ø:GOSUB36Ø:IF F
         THEN PRINT IN$:PRINT"
        {UP}{5 RIGHT}";
GC 45Ø  FOR I=Ø TO 24 STEP 3:B$
        =S$:FOR J=1 TO 2:IF F T
        HEN B$=MID$(IN$,I+J,1)
HA 46Ø  PRINT"{RVS}"B$L$;:IF I<
        24THEN PRINT"{OFF}";
HD 47Ø  GET A$:IF A$=N$ THEN47Ø
FK 48Ø  IF(A$>"/"ANDA$<":")OR(A
        $>"@"ANDA$<"G")THEN54Ø
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$="U")-1Ø*(A$="I
        ")-11*(A$="O")-12*(A$="
        P")
CM 487  A=A-13*(A$=S$):IF A THE
        N A$=MID$("ABCD123E456F
        Ø",A,1):GOTO 54Ø
MP 49Ø  IF A$=R$ AND((I=Ø)AND(J
        =1)OR F)THEN PRINT B$;:
        J=2:NEXT:I=24:GOTO55Ø
KC 5ØØ  IF A$="{HOME}" THEN PRI
        NT B$:J=2:NEXT:I=24:NEX
        T:F=Ø:GOTO44Ø
MX 51Ø  IF(A$="{RIGHT}")ANDF TH
        ENPRINT B$L$;:GOTO54Ø
GK 52Ø  IF A$<>L$ AND A$<>D$ OR
        ((I=Ø)AND(J=1))THEN GOS
        UB1Ø6Ø:GOTO47Ø
HG 53Ø  A$=L$+S$+L$:PRINT B$L$;
        :J=2-J:IF J THEN PRINT
        {SPACE}L$;:I=I-3
QS 54Ø  PRINT A$;:NEXT J:PRINT
        {SPACE}S$;
PM 55Ø  NEXT I:PRINT:PRINT"{UP}
        {5 RIGHT}";:INPUT#3,IN$
        :IF IN$=N$ THEN CLOSE3:
        GOTO22Ø
QC 56Ø  FOR I=1 TO 25 STEP3:B$=
        MID$(IN$,I):GOSUB32Ø:IF
         I<25 THEN GOSUB38Ø:A(I
        /3)=A
PK 57Ø  NEXT:IF A<>CK THEN GOSU
        B1Ø6Ø:PRINT"{BLK}{RVS}
        {SPACE}ERROR: REENTER L
        INE §4§":F=1:GOTO44Ø
HJ 58Ø  GOSUB1Ø8Ø:B=BS+AD-SA:FO
        R I=Ø TO 7:POKE B+I,A(I
        ):NEXT
QQ 59Ø  AD=AD+8:IF AD>EA THEN C
        LOSE3:PRINT"{DOWN}{BLU}
        ** END OF ENTRY **{BLK}
        {2 DOWN}":GOTO7ØØ
GQ 6ØØ  F=Ø:GOTO44Ø
QA 61Ø  PRINT"{CLR}{DOWN}{RVS}
        {SPACE}DISPLAY DATA ":G
        OSUB4ØØ:IF IN$=N$ THEN2
        2Ø
RJ 62Ø  PRINT"{DOWN}{BLU}PRESS:
         {RVS}SPACE{OFF} TO PAU
        SE, {RVS}RETURN{OFF} TO
         BREAK§4§{DOWN}"
KS 63Ø  GOSUB36Ø:B=BS+AD-SA:FOR
        I=BTO B+7:A=PEEK(I):GOS
        UB35Ø:GOSUB38Ø:PRINT S$
        ;
CC 64Ø  NEXT:PRINT"{RVS}";:A=CK
        :GOSUB35Ø:PRINT
KH 65Ø  F=1:AD=AD+8:IF AD>EA TH
        ENPRINT"{DOWN}{BLU}** E
        ND OF DATA **":GOTO22Ø
KC 66Ø  GET A$:IF A$=R$ THEN GO
        SUB1Ø8Ø:GOTO22Ø
EQ 67Ø  IF A$=S$ THEN F=F+1:GOS
        UB1Ø8Ø
AD 68Ø  ONFGOTO63Ø,66Ø,63Ø
CM 69Ø  PRINT"{DOWN}{RVS} LOAD
        {SPACE}DATA ":OP=1:GOTO
        71Ø
PC 7ØØ  PRINT"{DOWN}{RVS} SAVE
        {SPACE}FILE ":OP=Ø
RX 71Ø  IN$=N$:INPUT"{DOWN}FILE
        NAME§4§";IN$:IF IN$=N$
        {SPACE}THEN22Ø
PR 72Ø  F=Ø:PRINT"{DOWN}{BLK}
        {RVS}T{OFF}APE OR {RVS}
        D{OFF}ISK: §4§";
FP 73Ø  GET A$:IF A$="T"THEN PR
        INT"T{DOWN}":GOTO88Ø
HQ 74Ø  IF A$<>"D"THEN73Ø
HH 75Ø  PRINT"D{DOWN}":OPEN15,8
        ,15,"IØ:":B=EA-SA:IN$="
        Ø:"+IN$:IF OP THEN81Ø
SQ 76Ø  OPEN 1,8,8,IN$+",P,W":G
        OSUB86Ø:IF A THEN22Ø
FJ 77Ø  AH=INT(SA/256):AL=SA-(A
        H*256):PRINT#1,CHR$(AL)
        ;CHR$(AH);
PE 78Ø  FOR I=Ø TO B:PRINT#1,CH
        R$(PEEK(BS+I));:IF ST T
        HEN8ØØ
FC 79Ø  NEXT:CLOSE1:CLOSE15:GOT
        O94Ø
GS 8ØØ  GOSUB1Ø6Ø:PRINT"{DOWN}
        {BLK}ERROR DURING SAVE:
        §4§":GOSUB86Ø:GOTO22Ø
MA 81Ø  OPEN 1,8,8,IN$+",P,R":G
        OSUB86Ø:IF A THEN22Ø
GE 82Ø  GET#1,A$,B$:AD=ASC(A$+Z
        $)+256*ASC(B$+Z$):IF AD
        <>SA THEN F=1:GOTO85Ø
RX 83Ø  FOR I=Ø TO B:GET#1,A$:P
        OKE BS+I,ASC(A$+Z$):IF(
        I<>B)AND ST THEN F=2:AD
        =I:I=B
FA 84Ø  NEXT:IF ST<>64 THEN F=3
FQ 85Ø  CLOSE1:CLOSE15:ON ABS(F
        >Ø)+1 GOTO96Ø,97Ø
SA 86Ø  INPUT#15,A,A$:IF A THEN
         CLOSE1:CLOSE15:GOSUB1Ø
        6Ø:PRINT"{RVS}ERROR: "A
        $
GQ 87Ø  RETURN
EJ 88Ø  POKE183,PEEK(FA+2):POKE
        187,PEEK(FA+3):POKE188,
        PEEK(FA+4):IFOP=ØTHEN92
        Ø
HJ 89Ø  SYS 63466:IF(PEEK(783)A
        ND1)THEN GOSUB1Ø6Ø:PRIN
        T"{DOWN}{RVS} FILE NOT
        {SPACE}FOUND ":GOTO69Ø
CS 9ØØ  AD=PEEK(829)+256*PEEK(8
        3Ø):IF AD<>SA THEN F=1:
        GOTO97Ø
SC 91Ø  A=PEEK(831)+256*PEEK(83
        2)-1:F=F-2*(A<EA)-3*(A>
        EA):AD=A-AD:GOTO93Ø
KM 92Ø  A=SA:B=EA+1:GOSUB1Ø1Ø:P
        OKE78Ø,3:SYS 63338
JF 93Ø  A=BS:B=BS+(EA-SA)+1:GOS
        UB1Ø1Ø:ON OP GOTO95Ø:SY
        S 63591
AE 94Ø  GOSUB1Ø8Ø:PRINT"{BLU}**
         SAVE COMPLETED **":GOT
        O22Ø
XP 95Ø  POKE147,Ø:SYS 63562:IF
        {SPACE}ST>Ø THEN97Ø
FR 96Ø  GOSUB1Ø8Ø:PRINT"{BLU}**
         LOAD COMPLETED **":GOT
        O22Ø
DP 97Ø  GOSUB1Ø6Ø:PRINT"{BLK}
        {RVS}ERROR DURING LOAD:
        {DOWN}§4§":ON F GOSUB98
        Ø,99Ø,1ØØØ:GOTO22Ø
PP 98Ø  PRINT"INCORRECT STARTIN
        G ADDRESS (";:GOSUB36Ø:
        PRINT")":RETURN
GR 99Ø  PRINT"LOAD ENDED AT ";:
        AD=SA+AD:GOSUB36Ø:PRINT
         D$:RETURN
FD 1ØØØ PRINT"TRUNCATED AT END
        ING ADDRESS":RETURN
RX 1Ø1Ø AH=INT(A/256):AL=A-(AH
        *256):POKE193,AL:POKE1
        94,AH
FF 1Ø2Ø AH=INT(B/256):AL=B-(AH
        *256):POKE174,AL:POKE1
        75,AH:RETURN
FX 1Ø3Ø IF AD<SA OR AD>EA THEN
        1Ø5Ø
HA 1Ø4Ø IF(AD>511 AND AD<4Ø96Ø
        )OR(AD>49151 AND AD<53
        248)THEN GOSUB1Ø8Ø:F=Ø
        :RETURN
HC 1Ø5Ø GOSUB1Ø6Ø:PRINT"{RVS}
        {SPACE}INVALID ADDRESS
         {DOWN}{BLK}":F=1:RETU
        RN
AR 1Ø6Ø POKE SD+5,31:POKE SD+6
        ,2Ø8:POKE SD,24Ø:POKE
        {SPACE}SD+1,4:POKE SD+
        4,33
DX 1Ø7Ø FOR S=1 TO 1ØØ:NEXT:GO
        TO1Ø9Ø
PF 1Ø8Ø POKE SD+5,8:POKE SD+6,
        24Ø:POKE SD,Ø:POKE SD+
        1,9Ø:POKE SD+4,17
AC 1Ø9Ø FOR S=1 TO 1ØØ:NEXT:PO
        KE SD+4,Ø:POKE SD,Ø:PO
        KE SD+1,Ø:RETURN
```

Program 2: MLX For Commodore 128

```
AE 1ØØ TRAP 96Ø:POKE 4627,128:
       DIM NL$,A(7)
XP 11Ø Z2=2:Z4=254:Z5=255:Z6=2
       56:Z7=127:BS=256*PEEK(4
       627):EA=6528Ø
FB 12Ø BE$=CHR$(7):RT$=CHR$(13
       ):DL$=CHR$(2Ø):SP$=CHR$
       (32):LF$=CHR$(157)
KE 13Ø DEF FNHB(A)=INT(A/256):
       DEF FNLB(A)=A-FNHB(A)*2
       56:DEF FNAD(A)=PEEK(A)+
       256*PEEK(A+1)
JB 14Ø KEY 1,"A":KEY 3,"B":KEY
        5,"C":KEY 7,"D":VOL 15
       :IF RGR(Ø)=5 THEN FAST
FJ 15Ø PRINT"{CLR}"CHR$(142);C
       HR$(8):COLOR Ø,15:COLOR
        4,15:COLOR 6,15
GQ 16Ø 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 17Ø PRINT"{2 DOWN}
       {3 SPACES}COMPUTE!'S MA
       CHINE LANGUAGE EDITOR
       {2 DOWN}"
DK 18Ø PRINT"{BLK}STARTING ADD
       RESS[4]";:GOSUB 26Ø:IF
       {SPACE}AD THEN SA=AD:EL
       SE 18Ø
FH 19Ø PRINT"{BLK}{2 SPACES}EN
       DING ADDRESS[4]";:GOSUB
        26Ø:IF AD THEN EA=AD:E
       LSE 19Ø
MF 2ØØ PRINT"{DOWN}{BLK}CLEAR
       {SPACE}WORKSPACE [Y/N]?
       [4]":GETKEY A$:IF A$<>"
       Y" THEN 22Ø
QH 21Ø PRINT"{DOWN}{BLU}WORKIN
       G...";:BANK Ø:FOR A=BS
       {SPACE}TO BS+(EA-SA)+7:
       POKE A,Ø:NEXT A:PRINT"D
       ONE"
DC 22Ø PRINT TAB(1Ø)"{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 23Ø 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 24Ø GETKEY A$:A=INSTR("EDLS
       CQ",A$):ON A GOTO 34Ø,5
       5Ø,64Ø,65Ø,93Ø,94Ø:GOSU
       B 95Ø:GOTO 24Ø
SX 25Ø PRINT"STARTING AT";:GOS
       UB 26Ø:IF(AD<>Ø)OR(A$=N
       L$)THEN RETURN:ELSE 25Ø
BG 26Ø A$=NL$:INPUT A$:IF LEN(
       A$)=4 THEN AD=DEC(A$)
PP 27Ø IF AD=Ø THEN BEGIN:IF A
       $<>NL$ THEN 3ØØ:ELSE RE
       TURN:BEND
MA 28Ø IF AD<SA OR AD>EA THEN
       {SPACE}3ØØ
PM 29Ø IF AD>511 AND AD<6528Ø
       {SPACE}THEN PRINT BE$;:
       RETURN
SQ 3ØØ GOSUB 95Ø:PRINT"{RVS} I
       NVALID ADDRESS {DOWN}
       {BLK}":AD=Ø:RETURN
RD 31Ø CK=FNHB(AD):CK=AD-Z4*CK
       +Z5*(CK>Z7):GOTO 33Ø
DD 32Ø CK=CK*Z2+Z5*(CK>Z7)+A
AH 33Ø CK=CK+Z5*(CK>Z5):RETURN
QD 34Ø PRINT BE$;"{RVS} ENTER
       {SPACE}DATA ":GOSUB 25Ø
       :IF A$=NL$ THEN 22Ø
JA 35Ø BANK Ø:PRINT:F=Ø:OPEN 3
       ,3
BR 36Ø GOSUB 31Ø:PRINT HEX$(AD
       )+":";:IF F THEN PRINT
       {SPACE}L$:PRINT"{UP}
       {5 RIGHT}";
QA 37Ø FOR I=Ø TO 24 STEP 3:B$
       =SP$:FOR J=1 TO 2:IF F
       {SPACE}THEN B$=MID$(L$,
       I+J,1)
PS 38Ø PRINT"{RVS}"B$+LF$;:IF
       {SPACE}I<24 THEN PRINT"
       {OFF}";
RC 39Ø GETKEY A$:IF (A$>"/" AN
       D A$<":") OR(A$>"@" AND
        A$<"G") THEN 47Ø
AC 4ØØ IF A$="+" THEN A$="E":G
       OTO 47Ø
QB 41Ø IF A$="-" THEN A$="F":G
       OTO 47Ø
FB 42Ø IF A$=RT$ AND ((I=Ø) AN
       D (J=1) OR F) THEN PRIN
       T B$;:J=2:NEXT:I=24:GOT
       O 48Ø
RD 43Ø IF A$="{HOME}" THEN PRI
       NT B$:J=2:NEXT:I=24:NEX
       T:F=Ø:GOTO 36Ø
XB 44Ø IF (A$="{RIGHT}") AND F
        THEN PRINT B$+LF$;:GOT
       O 47Ø
JP 45Ø IF A$<>LF$ AND A$<>DL$
       {SPACE}OR ((I=Ø) AND (J
       =1)) THEN GOSUB 95Ø:GOT
       O 39Ø
PS 46Ø A$=LF$+SP$+LF$:PRINT B$
       +LF$;:J=2-J:IF J THEN P
       RINT LF$;:I=I-3
GB 47Ø PRINT A$;:NEXT J:PRINT
       {SPACE}SP$;
HA 48Ø NEXT I:PRINT:PRINT"{UP}
       {5 RIGHT}";:L$="
       {27 SPACES}"
DP 49Ø FOR I=1 TO 25 STEP 3:GE
       T#3,A$,B$:IF A$=SP$ THE
       N I=25:NEXT:CLOSE 3:GOT
       O 22Ø
BA 5ØØ A$=A$+B$:A=DEC(A$):MID$
       (L$,I,2)=A$:IF I<25 THE
       N GOSUB 32Ø:A(I/3)=A:GE
       T#3,A$
AR 51Ø NEXT I:IF A<>CK THEN GO
       SUB 95Ø:PRINT:PRINT"
       {RVS} ERROR: REENTER LI
       NE ":F=1:GOTO 36Ø
DX 52Ø PRINT BE$:B=BS+AD-SA:FO
       R I=Ø TO 7:POKE B+I,A(I
       ):NEXT I
XB 53Ø F=Ø:AD=AD+8:IF AD<=EA T
       HEN 36Ø
CA 54Ø CLOSE 3:PRINT"{DOWN}
       {BLU}** END OF ENTRY **
       {BLK}{2 DOWN}":GOTO 65Ø
MC 55Ø PRINT BE$;"{CLR}{DOWN}
       {RVS} DISPLAY DATA ":GO
       SUB 25Ø:IF A$=NL$ THEN
       {SPACE}22Ø
JF 56Ø BANK Ø:PRINT"{DOWN}
       {BLU}PRESS: {RVS}SPACE
       {OFF} TO PAUSE, {RVS}RE
       TURN{OFF} TO BREAK[4]
       {DOWN}"
XA 57Ø PRINT HEX$(AD)+":";:GOS
       UB 31Ø:B=BS+AD-SA
DJ 58Ø FOR I=B TO B+7:A=PEEK(I
       ):PRINT RIGHT$(HEX$(A),
       2);SP$;:GOSUB 32Ø:NEXT
       {SPACE}I
XB 59Ø PRINT"{RVS}";RIGHT$(HEX
       $(CK),2)
GR 6ØØ F=1:AD=AD+8:IF AD>EA TH
       EN PRINT"{BLU}** END OF
        DATA **":GOTO 22Ø
EB 61Ø GET A$:IF A$=RT$ THEN P
       RINT BE$:GOTO 22Ø
QK 62Ø IF A$=SP$ THEN F=F+1:PR
       INT BE$;
XS 63Ø ON F GOTO 57Ø,61Ø,57Ø
RF 64Ø PRINT BE$"{DOWN}{RVS} L
       OAD DATA ":OP=1:GOTO 66
       Ø
BP 65Ø PRINT BE$"{DOWN}{RVS} S
       AVE FILE ":OP=Ø
DM 66Ø F=Ø:F$=NL$:INPUT"FILENA
       ME[4]";F$:IF F$=NL$ THE
       N 22Ø
RF 67Ø PRINT"{DOWN}{BLK}{RVS}T
       {OFF}APE OR {RVS}D{OFF}
       ISK: [4]";
SQ 68Ø GETKEY A$:IF A$="T" THE
       N 85Ø:ELSE IF A$<>"D" T
       HEN 68Ø
SP 69Ø PRINT"DISK{DOWN}":IF OP
        THEN 76Ø
EH 7ØØ DOPEN#1,(F$+",P"),W:IF
       {SPACE}DS THEN A$=D$:GO
       TO 74Ø
JH 71Ø BANK Ø:POKE BS-2,FNLB(S
       A):POKE BS-1,FNHB(SA):P
       RINT"SAVING ";F$:PRINT
MC 72Ø FOR A=BS-2 TO BS+EA-SA:
       PRINT#1,CHR$(PEEK(A));:
       IF ST THEN A$="DISK WRI
       TE ERROR":GOTO 75Ø
GC 73Ø NEXT A:CLOSE 1:PRINT"
       {BLU}** SAVE COMPLETED
       {SPACE}WITHOUT ERRORS *
       *":GOTO 22Ø
RA 74Ø IF DS=63 THEN BEGIN:CLO
       SE 1:INPUT"{BLK}REPLACE
        EXISTING FILE [Y/N][4]
       ";A$:IF A$="Y" THEN SCR
       ATCH(F$):PRINT:GOTO 7ØØ
       :ELSE PRINT"{BLK}":GOTO
        66Ø:BEND
GA 75Ø CLOSE 1:GOSUB 95Ø:PRINT
       "{BLK}{RVS} ERROR DURIN
       G SAVE: [4]":PRINT A$:G
       OTO 22Ø
FD 76Ø DOPEN#1,(F$+",P"):IF DS
        THEN A$=DS$:F=4:CLOSE
       {SPACE}1:GOTO 79Ø
```

```
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
```

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.

COMPUTE!'s Gazette
Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE!'s Gazette
P.O. Box 10957
Des Moines, IA 50340

or call the Toll Free number listed below.

Change of Address. Please allow us 6–8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your Gazette subscription before we remind you to, send your current mailing label with payment or charge number, or call the Toll Free number listed below.

New Subscription. A one-year (12-month) U.S. subscription to *COMPUTE!'s Gazette* is $24 (2 years, $45; 3 years, $65. For subscription rates outside the U.S., see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of *COMPUTE!'s Gazette*, if you experience late delivery, or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!'s Gazette
1-800-247-5470
In IA 1-800-532-1272

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

BEFORE TYPING . . .

Before typing in programs, please refer to "How To Type In COMPUTE!'s GAZETTE Programs," which appears before the Program Listings.

Persuasion

(Article on page 48.)

```
RS 10 POKE 53269,0:POKE 53280,
      7:POKE 53281,7:PRINT"{CL
      R}{4 DOWN}{RED}"
SP 20 PRINT"{RVS}⁅K⁆{2 SPACES}
      {OFF}{DOWN}{3 LEFT}{RVS}
      ⁅K⁆{2 SPACES}{OFF}{DOWN}
      {3 LEFT}{RVS}⁅K⁆{OFF}⁅K⁆
      {RVS}⁅K⁆{OFF}⁅K⁆{DOWN}{4
       LEFT}{RVS}⁅K⁆{OFF}⁅K⁆{R
      VS}⁅K⁆{OFF}⁅K⁆{DOWN}{4 L
      EFT}{RVS}⁅K⁆{OFF}⁅K⁆{RVS
      }⁅K⁆{OFF}⁅K⁆{DOWN}{4 LEF
      T}{RVS}⁅K⁆{OFF}⁅K⁆{RVS}⁅
      K⁆{OFF}⁅K⁆";
SM 30 PRINT"{DOWN}{4 LEFT}{RVS
      }⁅K⁆{2 SPACES}{OFF}{DOWN
      }{3 LEFT}{RVS}⁅K⁆{2 SPAC
      ES}{OFF}{DOWN}{3 LEFT}{R
      VS}⁅K⁆{OFF}⁅K⁆{DOWN}{2 L
      EFT}{RVS}⁅K⁆{OFF}⁅K⁆{DOW
      N}{2 LEFT}{RVS}⁅K⁆{OFF}⁅
      K⁆{DOWN}{2 LEFT}{RVS}⁅K⁆
      {OFF}⁅K⁆{DOWN}{2 LEFT}{R
      VS}⁅K⁆{OFF}⁅K⁆{DOWN}{2 L
      EFT}{RVS}⁅K⁆{OFF}⁅K⁆{6 U
      P}"
MB 40 PRINT"{UP}{RVS}⁅K⁆{OFF}⁅
      K⁆ {RVS}⁅K⁆{2 SPACES}{OF
      F}⁅K⁆{RVS}⁅K⁆{2 SPACES}{
      OFF}{2 SPACES}{RVS}{2 SP
      ACES}{OFF} {RVS}⁅K⁆{OFF}
      ⁅K⁆{RVS}⁅K⁆{OFF}⁅K⁆ {RVS
      }⁅K⁆{OFF}⁅K⁆{2 SPACES}{R
      VS}{2 SPACES}{OFF}{2 SPA
      CES}{RVS}{2 SPACES}{OFF}
      {2 SPACES}{RVS}{2 SPACES
      }{OFF} {RVS}⁅K⁆{OFF}⁅K⁆{
      RVS}⁅K⁆{OFF}⁅K⁆ {RVS}⁅K⁆
      ";
PR 50 PRINT"{OFF}⁅K⁆ {RVS}⁅K⁆{
      OFF}⁅K⁆{2 SPACES}{RVS}⁅K
      ⁆{OFF}⁅K⁆{RVS}⁅K⁆{OFF}⁅K
      ⁆{RVS}⁅K⁆{OFF}⁅K⁆{RVS}⁅K
      ⁆{OFF}⁅K⁆{RVS}⁅K⁆{OFF}⁅K
      ⁆{RVS}⁅K⁆{OFF}⁅K⁆ {RVS}{
       2 SPACES}{OFF} {RVS}ƐKƎ{
       OFF}ƐKƎ{RVS}ƐKƎ{OFF}ƐKƎ
       {RVS}ƐKƎ{OFF}ƐKƎ {RVS}ƐK
       Ǝ{OFF}ƐKƎ{RVS}ƐKƎ{OFF}ƐK
       Ǝ{RVS}ƐKƎ ƐKƎ{OFF}ƐKƎ";
JK 6Ø PRINT" {RVS}ƐKƎ{OFF}ƐKƎ
       {RVS}ƐKƎ{OFF}ƐKƎ{2 SPACE
       S}{RVS}ƐKƎ{OFF}ƐKƎ{RVS}Ɛ
       KƎ{OFF}ƐKƎ{RVS}ƐKƎ{OFF}Ɛ
       KƎ{2 SPACES}{RVS}ƐKƎ{OFF
       }ƐKƎ{RVS}ƐKƎ{OFF}ƐKƎ{RVS
       }ƐKƎ{OFF}ƐKƎ{RVS}ƐKƎ{OFF
       }ƐKƎ{RVS}ƐKƎ{OFF}ƐKƎ{3 S
       PACES}{RVS}ƐKƎ{OFF}ƐKƎ {
       RVS}ƐKƎ{OFF}ƐKƎ{RVS}ƐKƎ{
       OFF}ƐKƎ{RVS}ƐKƎ{2 SPACES
       }";
CH 7Ø PRINT"{OFF}ƐKƎ {RVS}ƐKƎ{
       OFF}ƐKƎ {RVS}ƐKƎ {OFF}ƐK
       Ǝ {RVS}ƐKƎ{2 SPACES}{OFF
       }{2 SPACES}{RVS}{2 SPACE
       S}{OFF} {RVS}ƐKƎ{OFF}ƐKƎ
       {RVS}ƐKƎ{OFF}ƐKƎ{RVS}ƐKƎ
       {2 SPACES}{OFF}ƐKƎ {RVS}
       {2 SPACES}{OFF}{2 SPACES
       }{RVS}ƐKƎ{OFF}ƐKƎ {RVS}Ɛ
       KƎ{OFF}ƐKƎ{RVS}ƐKƎ{OFF}Ɛ
       KƎ{RVS}ƐKƎ{2 SPACES}{OFF
       }ƐKƎ";
HG 8Ø PRINT" {RVS}ƐKƎ{OFF}ƐKƎ
       {RVS}ƐKƎ{OFF}ƐKƎ{2 SPACE
       S}{RVS}ƐKƎ {OFF}ƐKƎ{3 SP
       ACES}{RVS}ƐKƎ{OFF}ƐKƎ{RV
       S}ƐKƎ{OFF}ƐKƎ{RVS}ƐKƎ{OF
       F}ƐKƎ{RVS}ƐKƎ{OFF}ƐKƎ{RV
       S}ƐKƎ{OFF}ƐKƎ{2 SPACES}{
       RVS}ƐKƎ{OFF}ƐKƎ {RVS}ƐKƎ
       {OFF}ƐKƎ {RVS}ƐKƎ{OFF}ƐK
       Ǝ{RVS}ƐKƎ{OFF}ƐKƎ{RVS}ƐK
       Ǝ{OFF}ƐKƎ{RVS} {OFF}";
GE 9Ø PRINT"ƐKƎ {RVS}ƐKƎ{OFF}Ɛ
       KƎ {RVS}ƐKƎ{OFF}ƐKƎ{2 SP
       ACES}{RVS}ƐKƎ{OFF}ƐKƎ{RV
       S} {OFF} {RVS}ƐKƎ{OFF}ƐK
       Ǝ{RVS}ƐKƎ{OFF}ƐKƎ{RVS}ƐK
       Ǝ{OFF}ƐKƎ{RVS}ƐKƎ{OFF}ƐK
       Ǝ{RVS}ƐKƎ{OFF}ƐKƎ{RVS}ƐK
       Ǝ{OFF}ƐKƎ{RVS}ƐKƎ{OFF}ƐK
       Ǝ{RVS}ƐKƎ{OFF}ƐKƎ {RVS}Ɛ
       KƎ{OFF}ƐKƎ {RVS}ƐKƎ{OFF}
       ƐKƎ{RVS}ƐKƎ{OFF}ƐKƎ";
GQ 1ØØ PRINT"{RVS}ƐKƎ{OFF}ƐKƎ{
        RVS}ƐKƎ{OFF}ƐKƎ {RVS}ƐK
        Ǝ{OFF}ƐKƎ {RVS}ƐKƎ{2 SP
        ACES}{OFF}ƐKƎ{RVS}ƐKƎ{O
        FF}ƐKƎ{RVS}ƐKƎ{OFF}ƐKƎ
        {RVS}{2 SPACES}{OFF}{2
        SPACES}{RVS}{2 SPACES}{
        OFF} {RVS}ƐKƎ{OFF}ƐKƎ{R
        VS}ƐKƎ{OFF}ƐKƎ {RVS}{2
        SPACES}{OFF}{2 SPACES}{
        RVS}{2 SPACES}{OFF}{2 S
        PACES}{RVS}{2 SPACES}{O
        FF} {RVS}";
RR 11Ø PRINT"ƐKƎ{OFF}ƐKƎ{RVS}Ɛ
        KƎ{OFF}ƐKƎ"CHR$(8)
KJ 12Ø POKE 53269,Ø:POKE 53249
        ,Ø:POKE 53251,Ø:PL=1
SA 13Ø FOR I=828 TO 892:READ A
        :POKE I,A:NEXT
HS 14Ø DIM C(1,15):FOR A=1TO 6
        :READ P:READ C(Ø,P),C(1
        ,P):NEXT:FOR A=Ø TO 6
BF 15Ø READ X1(A),X2(A):NEXT:F
        OR A=896 TO 959:READ B:
        POKE A,B:NEXT:POKE 2Ø4Ø
        ,14
PC 16Ø POKE 2Ø41,14:POKE 53269
        ,3:X(Ø)=5:Y(Ø)=Ø:X(1)=5
        :Y(1)=6:W(Ø)=6:W(1)=11
EP 17Ø POKE 53288,7:POKE 53287
        ,Ø
JA 18Ø PRINT"{CLR}{DOWN}":SYS
        828:PC=1Ø63:CP=55335
PR 19Ø DN$="{HOME}{23 DOWN}{RV
        S}Ɛ4Ǝ{7 SPACES}":D$="{R
        VS}P{DOWN}{LEFT}E{DOWN}
        {LEFT}R{DOWN}{LEFT}S{DO
        WN}{LEFT}U{DOWN}{LEFT}A
        {DOWN}{LEFT}S{DOWN}{LEF
        T}I{DOWN}{LEFT}O{DOWN}{
        LEFT}N"
CH 2ØØ E$="{RVS} {DOWN}{LEFT}
        {DOWN}{LEFT} {DOWN}{LEF
        T} {DOWN}{LEFT} {DOWN}{
        LEFT} {DOWN}{LEFT} {DOW
        N}{LEFT} {DOWN}{LEFT} {
        DOWN}{LEFT} "
HP 21Ø A$="{RED}{RVS}{2 SPACES
        }{DOWN}{2 LEFT}{2 SPACE
        S}{OFF}":B$="{YEL}N{2 S
        PACES}M":C$="{YEL}M{2 S
        PACES}N":SP=13:AC=3:FOR
         X=1 TO 4:PRINTSPC(SP);
XQ 22Ø FOR I=1 TO AC:PRINTA$"{
        DOWN}{3 LEFT}"B$"{2 UP}
        {3 SPACES}";:NEXT:SP=SP
        -3:AC=AC+1:PRINT"{2 DOW
        N}":NEXTX
CC 23Ø GOTO 26Ø
AS 24Ø SP=SP+6:AC=AC-2:PRINT"{
        2 UP}":FOR X=1 TO 3:PRI
        NT SPC(SP);:FOR I=1 TO
        AC
JH 25Ø PRINT A$"{DOWN}{3 LEFT}
        "B$"{2 UP}{3 SPACES}";:
        NEXT:SP=SP+3:AC=AC-1:PR
        INT"{2 DOWN}":NEXT
XK 26Ø SP=SP+5:AC=AC-2:PRINT"{
        3 UP}":FOR I=1 TO 3:PRI
        NT SPC(SP);:FOR X=1 TO
        AC
EM 27Ø PRINT C$"{DOWN}{3 LEFT}
        "A$"{3 SPACES}{2 UP}";:
        NEXT:SP=SP+3:AC=AC-1:PR
        INT"{2 DOWN}":NEXT
KP 28Ø PRINT"{HOME}{RVS}{BLU}{
        39 SPACES}":MT$="{RVS}
        {OFF}":POKE PC,16Ø
QX 29Ø POKE CP,6:T=12:M=11:AE=
        14:FOR Z=1 TO 1Ø
DX 3ØØ FOR I=1 TO T:PRINT MT$;
        :NEXT:PRINT"£"SPC(AE);
        :PRINT"Ɛ*Ǝ";
SK 31Ø FOR I=1 TO M:PRINT MT$;
        :NEXT
HG 32Ø PC=PC+4Ø:CP=CP+4Ø:POKE
        PC,16Ø:POKE CP,6:T=T-1:
        M=M-1:AE=AE+2:PRINT
FA 33Ø PRINT"{UP}":NEXT:PRINT"
        {RVS}{3 SPACES}"SPC(34)
        "{RVS}{2 SPACES}{OFF}":
        PRINT:POKE PC+4Ø,16Ø:PO
        KE CP+4Ø,6
MX 34Ø POKE PC+8Ø,16Ø:POKE CP+
        8Ø,11:PC=PC+8Ø:CP=CP+8Ø
        :PRINT"{2 UP}";
MR 35Ø PRINT"Ɛ4Ǝ{RVS}{3 SPACES
        }"SPC(34)"{RVS}{2 SPACE
        S}{OFF}":T=T+1:M=M+1:AE
        =AE-2:FOR Z=1 TO 1Ø:FOR
         I=1 TOT
BJ 36Ø PRINT MT$;:NEXT:PRINT"{
        RVS}Ɛ*Ǝ"SPC(AE);:PRINT"
        {RVS}£";:FOR I=1 TO M:
        PRINT MT$;:NEXT
CD 37Ø PC=PC+4Ø:CP=CP+4Ø:POKE
        PC,16Ø:POKE CP,11:T=T+1
        :M=M+1:AE=AE-2:PRINT:PR
        INT"{UP}"
GJ 38Ø NEXT:PRINT"{RVS}{39 SPA
        CES}"
FA 39Ø PRINT"{RVS}{39 SPACES}{
        HOME}"
JH 4ØØ POKE 1983,16Ø:POKE 2Ø23
        ,16Ø:POKE 56255,11:POKE
         56295,11
ED 41Ø POKE 53281,Ø:SP=53248:P
        RINT"{HOME}{3 DOWN}{RVS
        }{BLU} NORTH{2 DOWN}{4
        LEFT}"SC(Ø)"{LEFT}{RVS}
         "
KE 42Ø PRINT"{12 DOWN}{RVS}Ɛ4Ǝ
         SOUTH{2 DOWN}{4 LEFT}"
        SC(1)"{LEFT}{RVS} "
HA 43Ø JX(5)=-1:JY(5)=-1:JX(6)
        =-1:JY(6)=1:JX(9)=1:JY(
        9)=-1:JX(1Ø)=1:JY(1Ø)=1
QQ 44Ø GOSUB 73Ø
KQ 45Ø PL=-(PL=Ø):TS=PEEK(5328
        7):POKE 53287,PEEK(5328
        8):POKE 53288,TS
JK 46Ø IFPL=1THENPF=11:PRINT"{
        HOME}{DOWN}{BLU}"SPC(39
        )E$:PRINT"{HOME}{13 DOW
        N}Ɛ4Ǝ"SPC(39);:GOTO 48Ø
DX 47Ø PRINT"{HOME}{13 DOWN}Ɛ4
        Ǝ"SPC(39)E$:PF=6:PRINT"
        {HOME}{DOWN}{BLU}"SPC(3
        9);
RP 48Ø POKE 646,PF:PRINT D$"{H
        OME}"
JB 49Ø JV=PEEK(5632Ø):J=15-(JV
        AND15):X=X(PL):Y=Y(PL):
        X=X+JX(J):Y=Y+JY(J)
CA 5ØØ IF Y<Ø OR Y>6 THEN 49Ø
QQ 51Ø IF X<X1(Y) OR X>X2(Y) O
        R JY(J)=Ø THEN 49Ø
DX 52Ø X(PL)=X:Y(PL)=Y:S=5538Ø
        +X*3+Y*12Ø:C=PEEK(S)AND
        15:QQ=1:GOSUB 73Ø
DR 53Ø IF C(PL,C)=W(PL) THEN S
        C(PL)=SC(PL)+1:QQ=2
XS 54Ø IF C(PL,C)=2 THEN SC(PL
        )=SC(PL)-1
DK 55Ø GOSUB 69Ø
QM 56Ø IF C(PL,C)=15 THEN POKE
         S,15:POKE S+1,15:POKE
        S+4Ø,11:POKE S+41,11:GO
        TO 59Ø
AR 57Ø IF C(PL,C)=14 THEN POKE
         S,14:POKE S+1,14:POKE
        S+4Ø,6:POKE S+41,6:GOTO
         59Ø
KR 58Ø FOR A=Ø TO 1:FOR B=Ø TO
         1:POKE S+A+B*4Ø,C(PL,C
        ):NEXTB,A
GR 59Ø PRINT"{HOME}{3 DOWN}{RV
        S}{BLU} NORTH{2 DOWN}{4
         LEFT}"SC(Ø)"{LEFT}{RVS
        } "
DF 6ØØ PRINT"{11 DOWN}{RVS}Ɛ4Ǝ
         SOUTH{2 DOWN}{4 LEFT}"
        SC(1)"{LEFT}{RVS} "
SF 61Ø IF SC(Ø)=15 AND SC(1)=1
        5 THEN 68Ø
RG 62Ø IF SC(Ø)=16 OR SC(1)=16
         THEN 64Ø
JQ 63Ø GOTO 45Ø
AR 64Ø IF SC(1)>SC(Ø) THEN PRI
        NTDN$" THE SOUTH IS THE
         WINNER{12 SPACES}{HOME
        }":GOTO66Ø
PD 65Ø PRINT DN$" THE NORTH IS
         THE WINNER{12 SPACES}{
        HOME}"
FX 66Ø PRINT DN$"{DOWN}{RVS}{L
        EFT}PRESS ANY KEY TO PL
        AY AGAIN{HOME}":POKE 19
        8,Ø:WAIT 198,1
JE 67Ø PRINT"{CLR}":POKE 53281
        ,7:RUN 12Ø
EB 68Ø PRINT DN$"{8 SPACES}IT'
        S A TIE{HOME}":GOTO 66Ø
FJ 69Ø FOR O=1 TO QQ:FOR L=542
        72 TO 54295
```

```
ME 7ØØ POKE L,Ø:NEXT:POKE 5429
       6,15:POKE 54277,8:POKE
       54278,255:POKE 54276,23
PB 71Ø POKE 54287,4Ø:POKE 5427
       8,15:FOR I=1 TO 1Ø:POKE
        54273,I:NEXTI,O:RETURN
SH 72Ø POKE 646,PF:RETURN
JM 73Ø H1=51+X(Ø)*24:H2=51+X(1
       )*24:POKE SP+1,Ø:POKE S
       P+3,Ø
JD 74Ø POKE SP,H1 AND 255:POKE
        SP+2,H2 AND 255:MS=INT
       (H1/255)+2*INT(H2/255)
SB 75Ø POKE SP+16,MS:POKE SP+1
       ,64+Y(Ø)*24:POKE SP+3,6
       4+Y(1)*24:RETURN
PX 76Ø DATA 169,27,141,17,2Ø8,
       169,127,141,13,22Ø,169
DF 77Ø DATA 86,141,2Ø,3,169,3,
       141,21,3,169,129
AC 78Ø DATA 141,26,2Ø8,96,169,
       1,141,25,2Ø8,162,5
DC 79Ø DATA 16Ø,11,173,18,2Ø8,
       2Ø1,1Ø,176,4,162,146
DS 8ØØ DATA 16Ø,6,14Ø,32,2Ø8,1
       42,18,2Ø8,173,13,22Ø
DP 81Ø DATA 41,1,24Ø,3,76,49,2
       34,76,188,254
BF 82Ø DATA 2,5,5,5,14,15,14,6
       ,5,15,5,11,6,2,6,11,11
XH 83Ø DATA 2,3,7,2,8,1,9,Ø,1Ø
       ,1,9,2,8,3,7
ME 84Ø DATA Ø,Ø,Ø,Ø,Ø,Ø,31,255
MD 85Ø DATA 252,31,255,252,28,
       Ø,28,28
PK 86Ø DATA 255,156,28,193,156
       ,28,193,156
DJ 87Ø DATA 28,193,156,28,193,
       156,28,255
EB 88Ø DATA 156,28,192,28,28,1
       92,28,28
QB 89Ø DATA 192,28,28,192,28,2
       8,192,28
JC 9ØØ DATA 28,Ø,28,31,255,252
       ,31,255
SH 91Ø DATA 252,Ø,Ø,Ø,Ø,Ø,Ø,Ø
```

Ringside Karate

See instructions in article on page 52 before typing in.

```
3ØØØ:ØØ ØF CØ ØØ 15 5Ø ØØ 3F 65
3ØØ8:FØ ØØ 3F AØ ØØ 3F BØ ØØ 31
3Ø1Ø:3B A8 ØØ 3A AØ ØØ 2A AØ D5
3Ø18:ØØ 2A ØØ ØØ 3A ØØ ØØ 5E 33
3Ø2Ø:7Ø Ø1 57 5C Ø1 75 57 Ø1 37
3Ø28:5D 55 Ø3 57 5A ØØ D5 AA 8B
3Ø3Ø:ØØ 36 A8 ØØ 1D 7Ø ØØ 15 F2
3Ø38:7Ø ØØ 15 7Ø ØØ 3F FØ ØF 68
3Ø4Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ AØ
3Ø48:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ A8
3Ø5Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ BØ
3Ø58:ØØ ØØ ØØ ØØ ØØ ØØ Ø2 ØØ BC
3Ø6Ø:ØØ Ø2 8Ø ØØ ØA CØ ØØ 6B 1Ø
3Ø68:ØØ ØØ 6C ØØ ØØ 7Ø ØØ ØØ 18
3Ø7Ø:CØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 31
3Ø78:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØB E3
3Ø8Ø:ØØ 15 7Ø ØØ 55 5C ØØ 55 A5
3Ø88:5C Ø1 55 57 Ø1 55 57 Ø1 84
3Ø9Ø:55 57 Ø5 5F 55 Ø5 7Ø 15 BC
3Ø98:Ø5 7Ø 15 15 CØ Ø5 15 CØ 9Ø
3ØAØ:Ø5 15 CØ Ø5 17 ØØ Ø1 17 Ø3
3ØA8:ØØ Ø1 17 ØØ Ø1 17 ØF Ø1 AF
3ØBØ:17 3F C1 2B FF FE 3F CØ 92
3ØB8:3F ØØ ØØ ØØ ØØ ØØ ØØ ØF C7
3ØCØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 21
3ØC8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 29
3ØDØ:ØØ ØØ CØ ØØ ØØ CØ ØØ ØØ 4C
3ØD8:CØ ØØ ØØ 7Ø ØØ ØØ 7Ø ØØ 81
3ØEØ:ØØ 7Ø ØØ ØØ 7Ø ØØ ØØ 7Ø 51
3ØE8:ØØ ØØ 7Ø ØØ ØØ 7Ø ØØ ØØ 19
3ØFØ:7Ø ØØ ØØ AC ØØ ØØ FC ØØ 4E
3ØF8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØF 68
31ØØ:ØØ 15 7Ø ØØ 55 5C ØØ 55 27
31Ø8:5F Ø1 55 57 Ø1 55 57 Ø1 87
311Ø:55 57 Ø1 5F 55 Ø1 7Ø 15 AD
3118:Ø1 7Ø 15 Ø1 7Ø Ø5 Ø1 7Ø D3
312Ø:Ø5 Ø1 7Ø Ø5 Ø1 5C Ø1 Ø1 2Ø
3128:5C Ø1 ØØ 57 Ø1 ØØ 5B C1 EE
313Ø:ØØ 6F F1 ØØ BF FE Ø3 FØ 9D
3138:ØF ØØ ØØ ØØ ØØ ØØ ØØ ØF 31
314Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ A2
3148:FC ØØ Ø1 55 ØØ Ø3 FF ØØ AA
315Ø:Ø3 FA ØØ Ø3 FB ØØ Ø3 BA C3
3158:ØØ Ø3 AA ØØ Ø2 AA ØØ Ø2 8D
316Ø:AØ ØØ Ø3 AØ ØØ Ø5 E7 ØØ 61
3168:15 75 ØØ 1D 55 ØØ 1D 55 BE
317Ø:ØØ 17 55 ØØ 15 A9 ØØ 35 C7
3178:A9 ØØ ØF 57 ØØ Ø1 57 ØØ B9
318Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ E2
3188:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ EA
319Ø:ØØ ØØ ØØ ØØ ØØ 8Ø ØØ ØØ F4
3198:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FA
31AØ:ØØ ØØ ØØ ØØ FF C2 8Ø 55 64
31A8:6A 8Ø 55 6A 8Ø 5F CØ ØØ B4
31BØ:7Ø ØØ ØØ CØ ØØ ØØ ØØ ØØ 57
31B8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 1B
31CØ:ØØ Ø1 57 ØØ Ø3 FF ØØ Ø1 67
31C8:57 ØØ Ø1 57 ØØ Ø1 57 ØØ 1F
31DØ:Ø1 55 ØØ Ø1 55 ØØ Ø1 55 1B
31D8:ØØ Ø1 5D ØØ Ø1 5C ØØ Ø1 A1
31EØ:5C ØØ Ø5 5C ØØ 15 7Ø ØØ ØD
31E8:55 CØ Ø1 57 ØØ Ø5 5C ØØ 88
31FØ:15 7Ø ØF ØA 7Ø FF Ø2 AF B3
31F8:FF ØF FØ ØØ ØØ ØØ ØØ ØØ 3D
32ØØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 64
32Ø8:ØØ ØØ ØØ ØØ CØ ØØ ØØ 7Ø E2
321Ø:ØØ ØØ 5C ØØ ØØ 57 ØØ ØØ 5D
3218:55 CØ ØØ 15 CØ ØØ 35 CØ D9
322Ø:ØØ 35 CØ ØØ Ø5 CØ ØØ Ø5 1A
3228:CØ ØØ Ø5 CØ ØØ Ø5 CØ ØØ 2F
323Ø:C5 CØ ØØ C5 CØ ØØ FA BØ BØ
3238:ØØ FF FØ ØØ ØØ ØØ ØØ ØØ BA
324Ø:ØØ 3F ØØ ØØ 55 4Ø ØØ FF 2Ø
3248:CØ ØØ FE 8Ø ØØ FE CØ ØØ 72
325Ø:EE AØ ØØ EA 8Ø ØØ AA 8Ø DC
3258:ØØ A8 ØØ ØØ A8 ØØ ØØ EA 17
326Ø:D5 ØØ 7A D5 Ø1 5F 55 Ø1 8D
3268:55 5Ø Ø5 55 5Ø Ø5 55 5Ø 13
327Ø:15 D5 54 17 C5 57 15 A5 2C
3278:5D Ø5 A1 75 ØØ ØØ D5 ØØ Ø4
328Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ E4
3288:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ EC
329Ø:AØ ØØ ØØ AØ ØØ Ø2 BØ ØØ B8
3298:6A CØ ØØ 6B ØØ ØØ 6C ØØ F1
32AØ:ØØ 7Ø ØØ ØØ CØ ØØ 18 ØØ 57
32A8:Ø1 58 ØØ 15 58 ØØ 55 5C BE
32BØ:Ø5 55 5C 15 55 FØ 55 5C 3F
32B8:ØØ 55 7Ø ØØ 55 CØ ØØ ØØ 2E
32CØ:ØØ Ø3 55 ØØ ØØ 55 ØØ ØØ E5
32C8:55 ØØ ØØ 55 ØØ Ø1 55 ØØ DB
32DØ:Ø1 57 ØØ Ø1 5C ØØ Ø1 5Ø DØ
32D8:ØØ Ø5 7Ø ØØ Ø5 7Ø ØØ Ø5 7B
32EØ:7Ø ØØ 15 CØ ØØ 15 CØ ØØ Ø2
32E8:15 CØ ØØ 15 CØ ØØ 17 ØØ 8D
32FØ:ØØ 57 ØØ ØØ 57 ØØ ØØ AB 91
32F8:ØF ØØ FF FF ØØ ØØ ØØ ØØ E4
33ØØ:57 ØØ ØØ 7C ØØ ØØ 7Ø ØØ BA
33Ø8:ØØ CØ ØØ ØØ ØØ ØØ ØØ ØØ 9E
331Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 76
3318:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 7E
332Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 86
3328:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 8E
333Ø:ØØ ØØ ØØ ØF FC ØØ FF FF 6F
3338:FF FC ØF FF ØØ ØØ ØØ ØØ BF
334Ø:ØØ FC ØØ Ø1 55 ØØ Ø3 FF A6
3348:ØØ Ø3 FA ØØ Ø3 FB ØØ Ø3 D9
335Ø:BA 8Ø Ø3 AA ØØ Ø2 AA ØØ 9C
3358:Ø2 AØ ØØ Ø3 AØ ØØ Ø1 EØ FF
336Ø:ØØ Ø5 75 55 Ø5 55 55 Ø5 39
3368:D5 FF Ø5 75 CØ Ø1 5F CØ 3B
337Ø:Ø3 55 6A ØØ D5 6A ØØ 55 A8
3378:CØ ØØ 55 CØ ØØ FF CØ ØØ 77
338Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ E6
3388:ØØ ØØ ØØ ØØ BØ ØØ ØØ AC 21
339Ø:ØØ ØØ BØ ØØ ØØ BØ ØØ ØØ CF
3398:BØ ØØ ØØ BØ ØØ ØØ BØ ØØ C3
33AØ:ØØ BØ ØØ ØØ BØ ØØ ØØ ØØ B8
33A8:ØØ ØØ ØØ ØØ ØØ FØ ØØ ØØ D2
33BØ:AØ ØØ ØØ 8Ø ØØ ØØ 2Ø ØØ AF
33B8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØB 2A
33CØ:ØØ 15 7F ØØ 55 55 ØØ 55 B1
33C8:55 ØØ 55 55 Ø1 55 55 Ø1 E2
33DØ:55 FF Ø1 5F ØØ Ø1 5C ØØ B4
33D8:Ø1 7Ø ØØ Ø1 7Ø ØØ Ø1 7Ø E1
33EØ:ØØ Ø1 7Ø ØØ Ø1 7Ø ØØ Ø1 6Ø
33E8:7Ø ØØ Ø1 7Ø ØØ Ø1 7Ø ØØ 93
33FØ:Ø1 7Ø ØF Ø2 B3 FF Ø3 FF 99
33F8:FØ ØØ ØØ ØØ ØØ ØØ ØØ ØF E6
34ØØ:ØØ ØØ ØØ ØØ ØØ ØØ 5Ø ØØ Ø9
34Ø8:ØØ 5Ø ØØ ØØ 5Ø ØØ ØØ 5Ø 57
341Ø:ØØ ØØ 5Ø ØØ ØØ 5Ø ØØ ØØ C3
3418:5Ø ØØ ØØ AØ ØØ ØØ 2Ø ØØ F2
342Ø:ØØ 2Ø ØØ ØØ ØØ ØØ ØØ ØØ 9Ø
3428:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 9Ø
343Ø:ØØ ØØ ØØ CØ ØØ ØØ FØ ØØ 86
3438:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ AØ
344Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ A8
3448:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ BØ
345Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ B8
3458:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ CØ
346Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ C8
3468:Ø3 FØ ØØ Ø5 54 ØØ ØF FC 9C
347Ø:ØØ ØF E8 ØØ ØF EC ØØ ØE F3
3478:EA ØØ ØE A8 ØØ ØA A8 Ø2 1E
348Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ E8
3488:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FØ
349Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F8
3498:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø1
34AØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø9
34A8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 11
34BØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 19
34B8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø2 23
34CØ:ØØ ØA 8Ø ØØ ØE 8Ø ØØ Ø7 35
34C8:97 ØØ Ø5 D5 ØØ 15 55 ØØ F9
34DØ:55 55 ØØ 55 5C Ø1 5D 5C 8C
34D8:Ø3 C5 5C ØD 75 5C ØD 5F 27
34EØ:FF 35 55 55 35 D5 55 35 77
34E8:F5 55 35 C5 55 B5 CØ ØØ A7
34FØ:B5 CØ FF BA CF FF FF FF 8E
34F8:FF ØØ ØØ ØØ ØØ ØØ ØØ Ø2 63
35ØØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 6A
35Ø8:ØØ CØ ØØ ØØ 7Ø ØØ ØØ 5F 85
351Ø:ØØ ØØ 55 A8 ØØ 15 A8 ØØ 55
3518:ØF ØØ ØØ 35 CØ ØØ D5 7Ø 7F
352Ø:ØØ 55 5Ø ØØ 55 5C ØØ 5Ø 56
3528:57 ØØ 4Ø 15 CØ ØØ 15 AØ 68
353Ø:FF Ø5 A8 FF Ø1 FC FF FF EC
3538:FC ØØ ØØ ØØ ØØ ØØ ØØ Ø2 23
354Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ AA
3548:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ B2
355Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ BA
3558:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ C2
356Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ CA
3568:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ D2
357Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ DA
3578:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ E2
358Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ EA
3588:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F2
359Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FA
3598:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø3
35AØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØB
35A8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 13
35BØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 1B
35B8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 23
35CØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 2B
35C8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 33
35DØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 3B
35D8:ØØ ØØ ØØ ØØ 8Ø ØØ 1E AØ 24
35EØ:ØØ DE A1 55 DE AB 55 DF BC
35E8:EA D5 DF AB 5F DF ED 55 AØ
35FØ:FF FF 55 ØØ Ø3 D5 ØØ ØØ 75
35F8:FF ØØ ØØ ØØ ØØ ØØ ØØ ØØ 63
36ØØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 6C
36Ø8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 74
361Ø:ØØ ØØ ØØ ØØ ØØ ØØ Ø7 ØØ 8A
3618:ØØ 15 CØ ØØ 55 7Ø Ø1 55 A5
362Ø:7Ø D5 5D 7Ø D5 71 7Ø D7 1A
3628:C1 7Ø D5 7F 7Ø 75 55 F2 3B
363Ø:5D 55 5E 57 F5 5A D6 AD 56
3638:5A F6 AF FF 3F FØ ØØ ØØ 43
364Ø:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ AC
```

```
3648:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ B4
3650:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ BC
3658:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ C4
3660:BØ ØØ Ø2 BØ ØØ ØØ AC ØØ C9
3668:ØØ 2B ØØ ØØ ØB ØØ ØØ Ø3 FA
3670:ØØ ØØ Ø3 ØØ ØØ ØØ ØØ ØØ 3D
3678:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØB EF
3680:Ø3 FØ ØØ Ø5 54 ØØ ØF FC B8
3688:ØØ ØA FC ØØ ØE FC ØØ 2A A5
3690:EC ØØ ØA AC ØØ ØA A8 ØØ F8
3698:ØØ A8 ØØ ØØ A4 ØØ Ø7 9F Ø2
36AØ:CØ 1F 7F CØ 7F F7 FØ FF EE
36A8:DF FØ AF 7F FØ AA FD CØ 1E
36BØ:2A 77 CØ Ø3 DF CØ Ø3 FF 6Ø
36B8:ØØ Ø3 FF ØØ Ø1 57 ØØ ØF 5A
36CØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 2D
36C8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 35
36DØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 3D
36D8:ØØ ØØ ØØ ØØ ØØ Ø2 ØØ ØØ 4D
36EØ:Ø2 ØØ ØØ Ø2 ØØ ØØ Ø2 ØØ 72
36E8:ØØ Ø2 ØØ ØØ Ø2 ØØ ØØ Ø2 E7
36FØ:ØØ ØØ Ø2 ØØ ØØ Ø5 ØØ ØØ B1
36F8:ØF ØØ ØØ ØØ ØØ ØØ ØØ ØF FB
3700:Ø2 AB ØØ ØA AA CØ ØA AA 12
3708:CØ 2A AA BØ 2A AA BØ 2A 49
3710:AA BØ AA FA AC AB Ø2 AC C9
3718:AB Ø2 AC AC ØØ AB AC ØØ 45
3720:AB AC ØØ AB BØ ØØ 2B BØ D6
3728:ØØ 2B BØ ØØ 2B BØ FØ 2B AØ
3730:B3 FC 2B 7F FF 17 FC Ø3 6E
3738:FF ØØ ØØ ØØ ØØ ØØ ØØ ØF B5
3740:Ø2 AB ØØ ØA AA CØ ØA AA 52
3748:CØ 2A AA BØ 2A AA BØ 2A 89
3750:AA BØ AA FA BØ AB Ø2 BØ 2E
3758:AB Ø2 BØ AC Ø2 BØ AC Ø2 2C
3760:BØ AC Ø2 BØ BØ ØA BØ BØ 5D
3768:ØA BØ BØ 2A CØ BF 1A CØ BA
3770:BF C6 CØ 7F FD CØ FØ ØF 64
3778:CØ ØØ ØØ ØØ ØØ ØØ ØØ ØF 56
3780:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ EE
3788:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F6
3790:ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø2 Ø1
3798:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø7
37AØ:ØØ ØØ ØØ ØØ Ø2 81 55 Ø2 D1
37A8:AB FF Ø2 AB FF ØØ Ø1 5F 49
37BØ:ØØ ØØ Ø7 ØØ ØØ Ø1 ØØ ØØ Ø4
37B8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 27
37CØ:ØØ ØØ ØØ ØØ ØØ ØØ 3F ØØ AD
37C8:ØØ 55 4Ø ØØ FF CØ ØØ AF 47
37DØ:CØ ØØ EF CØ ØØ AE CØ ØØ E5
37D8:AA CØ ØØ AA 8Ø ØØ ØA 8Ø 1Ø
37EØ:ØØ ØA 4Ø ØØ 79 FØ ØØ F7 61
37E8:FC ØØ FF DF ØØ FF DF ØØ 93
37FØ:FF 7F ØØ EA FF ØØ EA F4 B8
37F8:ØØ 3F 5Ø ØØ 3F FØ ØØ ØØ FE
3800:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 7Ø
3808:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 78
3810:ØØ Ø2 ØØ ØØ ØA ØØ ØØ 2A 7B
3818:ØØ ØØ AA ØØ ØØ AB ØØ ØØ 8C
3820:AC ØØ ØØ AC ØØ ØØ AC ØØ ØB
3828:ØØ AC ØØ ØØ AC ØØ ØØ AC D5
3830:ØØ ØØ AC ØØ ØØ AC ØØ Ø1 E9
3838:5F ØØ Ø3 FF ØØ ØØ ØØ ØØ B8
3840:2A BØ ØØ 15 7Ø ØØ 2A BØ CB
3848:ØØ 2A BØ ØØ 2A BØ ØØ AA 18
3850:BØ ØØ AA BØ ØØ AA BØ ØØ 85
3858:BA BØ ØØ CA BØ ØØ ØA BØ 49
3860:ØØ ØA AC ØØ Ø2 AB ØØ ØØ A7
3868:AA CØ ØØ 2A BØ ØØ ØA AC 47
3870:FØ Ø2 A8 FF Ø2 5C FF F5 66
3878:7C ØØ ØF FC ØØ ØØ ØØ ØØ D8
3880:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FØ
3888:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F8
3890:ØA ØØ ØØ ØA CØ ØØ Ø2 BØ 61
3898:ØØ ØØ AD ØØ ØØ 2B ØØ ØØ 6B
38AØ:ØB ØØ ØØ Ø1 1B CØ ØØ 1A 9C
38A8:BC ØØ 1A AB ØØ ØA AA FØ E3
38BØ:ØA AA AC ØØ AA AB ØØ ØA 74
38B8:AA ØØ Ø2 AA ØØ ØØ AA ØØ BE
38CØ:Ø3 FØ ØØ Ø5 54 ØØ ØF FC FC
38C8:ØØ ØA FC ØØ ØE FC ØØ 2A E9
38DØ:EC ØØ ØA AC ØØ ØA A8 ØØ 3D
38D8:ØØ A8 ØØ ØØ A8 ØØ 55 A9 ØD
38EØ:ØØ FD A7 CØ FF 5F CØ 5F 3Ø
38E8:FF FØ ØF FF FØ ØF FF FC 38
```

```
38FØ:3F FF FC 7F FF 7C 9F FA C4
38F8:FC A7 FA F4 A9 FC 5Ø ØØ 62
3900:ØØ ØØ 2A ØØ ØØ ØA ØØ ØØ DF
3908:Ø2 ØØ ØØ ØØ ØØ ØØ ØØ ØØ 7B
3910:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 82
3918:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 8A
3920:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 92
3928:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 9A
3930:ØØ ØØ ØØ ØØ ØF FC FF FF ØF
3938:FF FF CØ 3F ØØ ØØ ØØ ØØ B6
3940:AA CØ ØØ AA CØ ØØ AA CØ FE
3948:ØØ AA CØ ØØ AA BØ ØØ 2A BF
3950:BØ ØØ ØA BØ ØØ ØA BØ ØØ FØ
3958:Ø2 AC ØØ Ø2 AC ØØ Ø2 AC 2D
3960:ØØ ØØ AB ØØ ØØ AB ØØ ØØ F6
3968:AB ØØ ØØ AB ØØ ØØ 2B ØØ C1
3970:ØØ 2A ØØ ØØ 2A ØØ FF D5 94
3978:ØØ FF FF ØØ ØØ ØØ ØØ ØØ EA
3980:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F2
3988:ØØ ØØ ØØ ØØ ØØ ØØ ØB ØØ 11
3990:ØØ 2B ØØ ØØ ØB ØØ ØØ ØB 31
3998:ØØ ØØ ØB ØØ ØØ ØB ØØ ØØ 98
39AØ:ØB ØØ ØØ ØB ØØ ØØ ØB ØØ 5F
39A8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØF 2A
39BØ:ØØ ØØ ØA ØØ ØØ Ø2 ØØ ØØ 6C
39B8:Ø8 ØØ ØØ ØØ ØØ ØØ ØØ ØB 3A
39CØ:ØØ 3F ØØ ØØ 55 4Ø ØØ FF AE
39C8:CØ ØØ AF CØ ØØ EF CØ Ø2 EØ
39DØ:AE CØ ØØ AA CØ ØØ AA 8Ø 51
39D8:ØØ ØA 8Ø ØØ ØA 4Ø 55 59 33
39EØ:FØ FF F7 FØ FF FF FØ 55 11
39E8:FD FØ ØØ F7 FØ 35 5F FØ 22
39FØ:AF FF 7Ø AF FF FØ Ø5 55 67
39F8:CØ Ø3 FF CØ Ø1 55 CØ ØØ 77
3A00:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 74
3A08:ØA ØØ ØØ ØA ØØ ØØ ØA ØØ 36
3A10:ØØ ØA ØØ ØØ ØA ØØ ØØ ØA 61
3A18:ØØ ØØ ØA ØØ ØØ Ø5 ØØ ØØ E1
3A20:Ø4 ØØ ØØ Ø4 ØØ ØØ ØØ ØØ D6
3A28:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 9C
3A30:ØØ ØØ ØØ ØØ ØØ Ø3 ØØ ØØ BØ
3A38:ØF ØØ ØØ ØØ ØØ ØØ ØØ ØØ 34
3A40:FE AB ØØ AA AA CØ AA AA 22
3A48:CØ AA AA CØ AA AA BØ CØ 4B
3A50:AA BØ CØ ØA BØ CØ ØA BØ 4C
3A58:CØ Ø2 BØ CØ Ø2 BØ ØØ Ø2 A4
3A60:BØ ØØ Ø2 BØ ØØ Ø2 BØ ØØ E1
3A68:Ø2 BØ ØØ Ø2 BØ ØØ Ø2 BØ 64
3A70:3F C2 BØ FF FD 7Ø CØ FF 7E
3A78:FØ ØØ ØØ ØØ ØØ ØØ ØØ ØF 74
3A80:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F4
3A88:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FC
3A90:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø5
3A98:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØD
3AAØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 15
3AA8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 1D
3ABØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 25
3AB8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØB 38
3ACØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 35
3AC8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 3D
3ADØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 45
3AD8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 4D
3AEØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØF 64
3AE8:CØ ØØ 15 5Ø ØØ 3F FØ ØØ 44
3AFØ:2B FØ ØØ 3B FØ ØØ AB BØ 7A
3AF8:ØØ 2A BØ ØØ 2A AØ ØØ Ø2 E3
3B00:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 76
3B08:ØØ ØØ ØØ Ø1 ØØ ØØ Ø7 ØØ 9C
3B10:28 ØF ØØ 2A AD ØØ 2A A4 67
3B18:ØØ ØØ FØ ØØ ØØ AC ØØ Ø2 61
3B20:AB ØØ ØA AA ØØ ØA AA ØØ D5
3B28:2A FA ØØ AB Ø2 Ø5 AB ØØ A8
3B30:15 AC ØF 3F BØ FF 3F FF 36
3B38:FF ØØ ØØ ØØ ØØ ØØ ØØ Ø2 BØ
3B40:Ø2 AØ ØØ Ø2 9Ø ØØ 5E 7Ø B1
3B48:ØØ FD F4 ØØ FF FD ØØ 5F 34
3B50:FF 4Ø ØF FF CØ ØF F7 DØ BB
3B58:ØF F1 DØ ØF FE BØ F5 7A FE
3B60:B4 AA AA AC AA AB AC AA Ø4
3B68:AF AC AA A3 AC ØØ Ø3 AE 8B
3B70:FF F3 AE FF FF 5E FF FF 33
3B78:FF ØØ ØØ ØØ ØØ ØØ ØØ Ø2 FØ
3B80:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F6
3B88:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FE
3B90:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø7
```

```
3B98:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØF
3BAØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 17
3BA8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 1F
3BBØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 27
3BB8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØB 3A
3BCØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 37
3BC8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 3F
3BDØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 47
3BD8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 4F
3BEØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 57
3BE8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 5F
3BFØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 67
3BF8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 6F
3C00:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 78
3C08:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 8Ø
3C10:ØØ ØØ ØØ ØØ ØØ ØØ 2C ØØ EØ
3C18:ØØ AB ØØ Ø2 AA CØ Ø2 AA A2
3C20:BF Ø2 BA A9 Ø2 8E A9 Ø2 8A
3C28:83 E9 Ø2 FE A9 4F AA A5 92
3C30:7A AA 9A 5A AF EA 5A BA 22
3C38:A5 FF FA AF ØØ ØF FC ØØ 14
3C40:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ B8
3C48:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ CØ
3C50:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ C8
3C58:ØØ ØØ ØØ ØØ Ø2 ØØ ØØ ØA EA
3C60:B4 FF CA B7 FF 6A B7 FD 1F
3C68:AB F7 5F 6A F7 FF DB F7 B6
3C70:FF FF FF FF CØ ØØ FF ØØ EE
3C78:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FØ
3C80:ØØ 82 2Ø A3 4Ø 2Ø 96 4Ø C7
3C88:A9 ØØ 8D 3C Ø3 8D 3D Ø3 17
3C90:8D 3E Ø3 8D 3F Ø3 A9 FF F1
3C98:8D 15 DØ 8D 1C DØ A9 Ø1 88
3CAØ:8D 25 DØ A9 ØØ 8D 26 DØ 31
3CA8:A2 ØØ A9 ØA 9D 27 DØ E8 5C
3CBØ:EØ Ø8 DØ F6 A9 CØ 8D F8 89
3CB8:Ø7 A9 C1 8D F9 Ø7 A9 C2 32
3CCØ:8D FA Ø7 A9 C3 8D FB Ø7 8D
3CC8:A9 D9 8D FC Ø7 A9 DA 8D 3Ø
3CDØ:FD Ø7 A9 DB 8D FE Ø7 A9 1D
3CD8:DC 8D FF Ø7 A9 B4 8D Ø5 D3
3CEØ:DØ 8D Ø7 DØ 8D ØD DØ 8D E2
3CE8:ØF DØ A9 AØ 8D Ø1 DØ 8D FB
3CFØ:Ø3 DØ 8D Ø9 DØ 8D ØB DØ Ø5
3CF8:A9 51 8D ØØ DØ 8D Ø4 DØ E1
3D00:A9 69 8D Ø2 DØ 8D Ø6 DØ 15
3D08:A9 B4 8D Ø8 DØ 8D ØC DØ 5C
3D10:A9 CC 8D ØA DØ 8D ØE DØ 8E
3D18:AD ØØ DC 29 1Ø DØ 35 A5 6B
3D20:FC DØ 31 AD ØØ DC 29 ØF 23
3D28:C9 ØA DØ Ø8 A9 E2 2Ø D1 8F
3D30:3E 4C BF 3D C9 Ø9 DØ Ø8 C4
3D38:A9 EA 2Ø D1 3E 4C BF 3D 43
3D40:C9 ØB DØ Ø8 A9 DE 2Ø D1 D7
3D48:3E 4C BF 3D A9 E6 2Ø D1 BB
3D50:3E 4C BF 3D AD ØØ DC 29 19
3D58:1Ø FØ Ø4 A9 ØØ 85 FC AD EF
3D60:ØØ DC 29 ØF C9 ØF DØ Ø3 57
3D68:4C BF 3D 38 C9 Ø8 9Ø 3Ø E3
3D70:AD ØE DØ 18 C9 28 BØ Ø3 34
3D78:4C BF 3D AD Ø8 DØ 38 ED 6D
3D80:ØØ DØ C9 14 BØ Ø3 4C BF 93
3D88:3D A2 Ø5 CE Ø8 DØ CE ØA Ø3
3D90:DØ CE ØC DØ CE ØE DØ CA DØ
3D98:DØ F1 EE FF Ø7 4C BF 3D FB
3DAØ:AD ØE DØ 38 C9 EB 9Ø Ø3 35
3DA8:4C BF 3D A2 Ø5 EE Ø8 DØ CF
3DBØ:EE ØA DØ EE ØC DØ EE ØE BD
3DB8:DØ CA DØ F1 EE FF Ø7 AD BA
3DCØ:Ø1 DC 29 1Ø DØ 35 A5 FB BB
3DC8:DØ 31 AD Ø1 DC 29 ØF C9 31
3DDØ:Ø6 DØ Ø8 A9 C9 2Ø B1 3E 8E
3DD8:4C 6C 3E C9 Ø7 DØ Ø8 A9 2E
3DEØ:C5 2Ø B1 3E 4C 6C 3E C9 BA
3DE8:Ø5 DØ Ø8 A9 D1 2Ø B1 3E 66
3DFØ:4C 6C 3E A9 CD 2Ø B1 3E 9F
3DF8:4C 6C 3E AD Ø1 DC 29 1Ø 35
3E00:FØ Ø4 A9 ØØ 85 FB AD Ø1 A3
3E08:DC 29 ØF C9 ØF DØ Ø3 4C C9
3E10:6C 3E 38 C9 Ø8 9Ø 25 AD 7Ø
3E18:ØØ DØ 18 C9 2D BØ Ø3 4C E6
3E20:6C 3E A2 Ø5 CE ØØ DØ CE ED
3E28:Ø2 DØ CE Ø4 DØ CE Ø6 DØ 92
3E30:CA DØ F1 EE FA Ø7 EE FA 4Ø
3E38:Ø7 4C 6C 3E AD Ø8 DØ 38 24
```

```
3E40:ED ØØ DØ C9 14 BØ Ø3 4C 2Ø
3E48:6C 3E AD ØØ DØ 38 C9 DC 18
3E5Ø:9Ø Ø3 4C 6C 3E A2 Ø5 EE 9B
3E58:ØØ DØ EE Ø2 DØ EE Ø4 DØ 22
3E6Ø:EE Ø6 DØ CA DØ F1 EE FA C3
3E68:Ø7 EE FA Ø7 AD 1E DØ A2 1E
3E7Ø:64 2Ø BD 3F AD 1E DØ AA Ø5
3E78:29 12 C9 12 FØ 73 8A 29 FB
3E8Ø:48 C9 48 FØ 6C 8A 29 18 A3
3E88:C9 18 FØ 65 8A 29 42 C9 AB
3E9Ø:42 FØ 5E A5 FB A8 A9 CØ 27
3E98:2Ø B1 3E 98 85 FB A5 FC 47
3EAØ:A8 A9 D9 2Ø D1 3E 98 85 57
3EA8:FC A2 32 2Ø BD 3F 4C 18 3Ø
3EBØ:3D A2 ØØ 9D F8 Ø7 18 69 CB
3EB8:Ø1 E8 EØ Ø4 DØ F5 A9 Ø1 FE
3ECØ:85 FB AD F8 Ø7 C9 CØ FØ 16
3EC8:Ø7 C9 D5 FØ Ø3 2Ø 6D 4Ø B8
3EDØ:6Ø A2 ØØ 9D FC Ø7 18 69 9D
3ED8:Ø1 E8 EØ Ø4 DØ F5 A9 Ø1 1F
3EEØ:85 FC AD FC Ø7 C9 EE FØ 13
3EE8:E7 C9 D9 FØ E3 2Ø 6D 4Ø DØ
3EFØ:6Ø AD F8 Ø7 C9 CD FØ 7Ø 7Ø
3EF8:AD FC Ø7 C9 E6 FØ 69 AD 84
3FØØ:F8 Ø7 C9 CØ DØ 51 AD FC 26
3FØ8:Ø7 C9 D9 FØ 4A 2Ø 47 4Ø 68
3F1Ø:A2 ØA CE ØØ DØ CE Ø2 DØ D2
3F18:CE Ø4 DØ CE Ø6 DØ CA DØ DF
3F2Ø:F1 EE FA Ø7 EE FA Ø7 A9 3E
3F28:D5 2Ø B1 3E A9 D9 2Ø D1 7A
3F3Ø:3E 2Ø 13 4Ø A2 FF 2Ø BD 4F
3F38:3F A2 FF 2Ø BD 3F A2 FF 31
3F4Ø:2Ø BD 3F A9 Ø3 CD 3F Ø3 91
3F48:FØ Ø3 4C B4 3C A9 E8 CD FC
3F5Ø:3E Ø3 DØ F6 4C E8 3F AD 6A
3F58:FC Ø7 C9 D9 DØ ØD AD F8 FC
3F6Ø:Ø7 C9 CØ DØ Ø6 4C 93 3E CØ
3F68:4C 93 3E AD FC Ø7 C9 D9 Ø6
3F7Ø:DØ F6 AD F8 Ø7 C9 CØ FØ 2C
3F78:EF 2Ø 47 4Ø A2 ØA EE Ø8 Ø7
3F8Ø:DØ EE ØA DØ EE ØC DØ EE A9
3F88:ØE DØ CA DØ F1 A9 EE 2Ø DC
3F9Ø:D1 3E A9 CØ 2Ø B1 3E 2Ø 2D
3F98:FF 3F A2 FF 2Ø BD 3F A2 54
3FAØ:FF 2Ø BD 3F A2 FF 2Ø BD E5
3FA8:3F A9 Ø3 CD 3D Ø3 FØ Ø3 49
3FBØ:4C B4 3C A9 E8 CD 3C Ø3 9E
3FB8:DØ F6 4C C6 3F AØ FF 88 58
3FCØ:DØ FD CA DØ F8 6Ø A2 Ø5 21
3FC8:AØ ØB 18 2Ø FØ FF AØ ØØ 28
3FDØ:B9 A3 45 C9 ØØ FØ Ø6 2Ø 4A
3FD8:D2 FF C8 DØ F3 A9 ØØ 85 B2
3FEØ:C6 A5 C6 FØ FC 4C 82 3C 6E
3FE8:A2 Ø5 AØ ØB 18 2Ø FØ FF E1
3FFØ:AØ ØØ B9 BØ 45 C9 ØØ FØ 44
3FF8:E4 2Ø D2 FF C8 DØ F3 18 D5
4ØØØ:AD 3C Ø3 69 64 8D 3C Ø3 32
4ØØ8:AD 3D Ø3 69 ØØ 8D 3D Ø3 59
4Ø1Ø:4C 24 4Ø 18 AD 3E Ø3 69 1F
4Ø18:64 8D 3E Ø3 AD 3F Ø3 69 FF
4Ø2Ø:ØØ 8D 3F Ø3 A2 Ø8 AØ 22 B4
4Ø28:18 2Ø FØ FF AD 3D Ø3 AE F1
4Ø3Ø:3C Ø3 2Ø CD BD A2 ØD AØ A3
4Ø38:22 18 2Ø FØ FF AD 3F Ø3 1B
4Ø4Ø:AE 3E Ø3 2Ø CD BD 6Ø A9 D9
4Ø48:ØF 8D 18 D4 8D ØC D4 A9 F3
4Ø5Ø:Ø7 8D ØD D4 A9 1C 8D Ø8 87
4Ø58:D4 A9 ØØ 8D Ø7 D4 A9 81 E6
4Ø6Ø:8D ØB D4 A2 ØA 2Ø BD 3F BA
4Ø68:A9 8Ø 8D ØB D4 A9 E1 8D DE
4Ø7Ø:ØØ D4 A9 1E 8D Ø1 D4 A9 Ø1
4Ø78:44 8D Ø5 D4 A9 ØØ 8D Ø6 DA
4Ø8Ø:D4 A9 Ø7 8D 18 D4 A9 81 78
4Ø88:8D Ø4 D4 A2 ØA 2Ø BD 3F 21
4Ø9Ø:A9 8Ø 8D Ø4 D4 6Ø A2 ØØ 65
4Ø98:A9 ØØ 9D ØØ D4 E8 EØ 18 C5
4ØAØ:DØ F8 6Ø A9 ØØ 8D 21 DØ B7
4ØA8:A9 ØF 8D 2Ø DØ AØ ØØ B9 38
4ØBØ:DF 4Ø 2Ø D2 FF C8 DØ F7 1F
4ØB8:B9 DF 41 2Ø D2 FF C8 DØ 31
4ØCØ:F7 B9 DF 42 2Ø D2 FF C8 EØ
4ØC8:DØ F7 B9 DF 43 2Ø D2 FF 25
4ØDØ:C8 DØ F7 B9 DF 44 FØ Ø6 7C
4ØD8:2Ø D2 FF C8 DØ F5 6Ø 93 5D
4ØEØ:2Ø 12 Ø5 2Ø 2Ø 2Ø 2Ø 2Ø 7A
```

```
4ØE8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 69
4ØFØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 71
4ØF8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 79
41ØØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 82
41Ø8:2Ø 92 2Ø 12 2Ø 92 2Ø 2Ø 1Ø
411Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 84
4118:97 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 56
412Ø:2Ø 2Ø 2Ø 2Ø 92 2Ø 2Ø 2Ø 36
4128:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 9C
413Ø:Ø5 2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø 73
4138:2Ø 2Ø 12 2Ø 2Ø 92 2Ø 2Ø C2
414Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 B4
4148:97 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 86
415Ø:2Ø 2Ø 2Ø 2Ø 92 2Ø 2Ø 2Ø 66
4158:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 CC
416Ø:Ø5 2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø A3
4168:2Ø 2Ø 12 2Ø 2Ø 92 2Ø 2Ø F2
417Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø DF B2
4178:12 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø F3
418Ø:2Ø 2Ø 92 A9 2Ø 2Ø 2Ø 2Ø E9
4188:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 Ø5 D3
419Ø:2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø AF
4198:2Ø 12 2Ø 2Ø 92 2Ø 2Ø 2Ø 2B
41AØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 23
41A8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2B
41BØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 33
41B8:2Ø 2Ø AØ 12 Ø5 2Ø 92 2Ø 76
41CØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 2Ø 27
41C8:2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø E7
41DØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 53
41D8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 5B
41EØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 63
41E8:12 Ø5 2Ø 92 2Ø 12 57 48 23
41FØ:49 54 45 92 2Ø 12 2Ø 2Ø A8
41F8:92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø B4
42ØØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 84
42Ø8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 8C
421Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 86
4218:2Ø 92 2Ø 12 53 43 4F 52 ØF
422Ø:45 92 2Ø 12 2Ø 2Ø 92 2Ø D7
4228:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø AC
423Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø B4
4238:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø BC
424Ø:2Ø 2Ø 2Ø 2Ø 12 2Ø 92 FE
4248:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 BE
425Ø:2Ø 2Ø 92 2Ø 96 D1 2Ø D1 4F
4258:2Ø D1 2Ø D1 2Ø D1 2Ø D1 DC
426Ø:2Ø D1 2Ø D1 2Ø D1 2Ø D1 E4
4268:2Ø D1 2Ø D1 2Ø D1 2Ø D1 EC
427Ø:2Ø D1 12 Ø5 2Ø 92 2Ø 2Ø B7
4278:3Ø 3Ø 3Ø 2Ø 2Ø 12 2Ø 2Ø D2
428Ø:92 96 D1 2Ø D1 2Ø D1 2Ø Ø3
4288:D1 2Ø D1 2Ø D1 2Ø D1 2Ø ØD
429Ø:D1 2Ø D1 2Ø D1 2Ø D1 2Ø 15
4298:D1 2Ø D1 2Ø D1 2Ø D1 2Ø 1D
42AØ:12 Ø5 2Ø 92 2Ø 2Ø 2Ø 2Ø 7E
42A8:2Ø 2Ø 2Ø 12 2Ø 2Ø 92 2Ø 31
42BØ:96 D1 2Ø D1 2Ø D1 2Ø D1 7Ø
42B8:2Ø D1 2Ø D1 2Ø D1 2Ø D1 3D
42CØ:2Ø D1 2Ø D1 2Ø D1 2Ø D1 45
42C8:2Ø D1 2Ø D1 2Ø D1 12 Ø5 64
42DØ:2Ø 92 2Ø 12 9B 42 4C 41 EE
42D8:43 4B 92 2Ø 12 Ø5 2Ø 2Ø 2B
42EØ:92 96 D1 2Ø 12 95 A9 2Ø EA
42E8:92 A9 9E B7 B7 B7 B7 B7 34
42FØ:B7 B7 B7 B7 B7 B7 B7 B7 75
42F8:B7 B7 B7 B7 B7 B7 B7 95 5B
43ØØ:DF 12 2Ø DF 92 96 D1 2Ø AF
43Ø8:12 Ø5 2Ø 92 2Ø 12 9B 53 D9
431Ø:43 4F 52 45 92 2Ø 12 Ø5 E8
4318:2Ø 2Ø 92 2Ø 9E CE 96 D1 3A
432Ø:12 95 2Ø 92 96 2Ø D1 2Ø 3B
4328:D1 2Ø D1 2Ø D1 2Ø D1 2Ø AE
433Ø:D1 2Ø D1 2Ø D1 2Ø D1 2Ø B6
4338:D1 2Ø D1 95 12 2Ø 92 96 1Ø
434Ø:D1 9E CD 96 D1 12 Ø5 2Ø 7B
4348:92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø Ø8
435Ø:12 2Ø 2Ø 92 9E CE 96 D1 44
4358:9E CE 12 95 2Ø 92 9E B7 BD
436Ø:B7 B7 B7 B7 B7 B7 B7 B7 E6
4368:B7 B7 B7 B7 B7 B7 B7 B7 EE
437Ø:B7 B7 B7 B7 B7 95 12 2Ø 8B
4378:92 9E CD 96 D1 9E CD 12 C9
438Ø:Ø5 2Ø 92 2Ø 2Ø 3Ø 3Ø 3Ø 38
4388:2Ø 2Ø 12 2Ø 2Ø 92 96 D1 B5
```

```
439Ø:9E CE 1C D1 12 95 2Ø 92 74
4398:1C D1 2Ø D1 2Ø D1 2Ø D1 1D
43AØ:2Ø D1 2Ø D1 2Ø D1 2Ø D1 27
43A8:2Ø D1 2Ø D1 2Ø D1 2Ø 95 F2
43BØ:12 2Ø 92 1C D1 9E CD 96 97
43B8:D1 12 Ø5 2Ø 92 2Ø 2Ø 2Ø C4
43CØ:2Ø 2Ø 2Ø 2Ø 12 2Ø 2Ø 92 49
43C8:9E CE 1C D1 12 9B A9 2Ø 65
43DØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 57
43D8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 5F
43EØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø DF 27
43E8:92 1C D1 9E CD 12 Ø5 2Ø C4
43FØ:92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø BØ
43F8:12 2Ø 2Ø 92 1C D1 12 9B A5
44ØØ:A9 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 4D
44Ø8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 9Ø
441Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 98
4418:2Ø 2Ø 2Ø DF 92 1C D1 12 75
442Ø:Ø5 2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø 69
4428:2Ø 2Ø 12 2Ø 2Ø 92 12 9B 18
443Ø:A9 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 7D
4438:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø CØ
444Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø C8
4448:2Ø 2Ø 2Ø 2Ø 2Ø DF 92 12 A6
445Ø:Ø5 2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø 99
4458:2Ø 2Ø 12 2Ø 2Ø 9B 2Ø 2Ø ØD
446Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø E8
4468:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø FØ
447Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø F8
4478:2Ø 2Ø 2Ø 2Ø Ø5 2Ø 92 2Ø ØD
448Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 2Ø EC
4488:2Ø 9B 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø EF
449Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 19
4498:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 21
44AØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 29
44A8:Ø5 2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø F1
44BØ:2Ø 2Ø 12 2Ø 2Ø 98 2Ø 2Ø 59
44B8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 41
44CØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 49
44C8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 51
44DØ:2Ø 2Ø 2Ø 2Ø Ø5 2Ø 92 2Ø 65
44D8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 12 2Ø 45
44EØ:2Ø 92 2Ø 97 DF 12 2Ø 92 B5
44E8:A9 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 36
44FØ:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 79
44F8:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø DF 41
45ØØ:12 2Ø 92 A9 2Ø 12 Ø5 2Ø FB
45Ø8:92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø CB
451Ø:12 2Ø 2Ø 92 2Ø 97 2Ø 12 8A
4518:2Ø 2Ø 2Ø 2Ø 2Ø 52 49 4E EB
452Ø:47 53 49 44 45 2Ø 2Ø 4B C6
4528:41 52 41 54 45 2Ø 2Ø 2Ø 6Ø
453Ø:2Ø 2Ø 92 2Ø 2Ø 12 Ø5 2Ø 9A
4538:92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø FB
454Ø:12 2Ø 2Ø 92 2Ø 97 2Ø 12 BA
4548:2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 6F
455Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø DA
4558:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø E2
456Ø:2Ø 2Ø 12 2Ø 92 2Ø 2Ø 12 AE
4568:Ø5 2Ø 92 2Ø 2Ø 2Ø 2Ø 2Ø B3
457Ø:2Ø 2Ø 12 2Ø 92 2Ø 12 Ø5 95
4578:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø Ø3
458Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø ØB
4588:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 13
459Ø:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 1B
4598:2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 91 91 77
45AØ:92 8E ØØ 9C 57 48 49 54 A4
45A8:45 2Ø 57 49 4E 53 21 ØØ 5F
45BØ:9C 42 4C 41 43 4B 2Ø 57 96
45B8:49 4E 53 21 ØØ ØØ ØØ ØØ F7
```

Menu System

(Article on page 74.)

```
RK 1Ø DIM P$(2Ø),L$(2Ø),E$(2Ø)
      ,S$(2Ø),R$(2Ø):Q$=CHR$(3
      4):C$=","
DM 2Ø M$="INSERT DISK AND PRES
      S A KEY":GOSUB56Ø
DE 3Ø S1$="5 DIMP$(2Ø),L$(2Ø),
      J$(2Ø),S$(2Ø),R$(2Ø),E$(
      2Ø)"
SF 4Ø S2$="1Ø X=1:F$="+CHR$(34
      )
```

```
PR 50  S2$=S2$+"{CLR}{22 DOWN}S
       ELECT{HOME}"
MF 60  S3$="20 FORN=1TO20:E$(N)
       ="+CHR$(34)+",8"+CHR$(34
       )
EJ 70  S3$=S3$+":NEXT:OPEN2,8,2
       ,"+CHR$(34)
GC 80  S3$=S3$+"Ø:.M,S,R"+CHR$(
       34)+":M$="+CHR$(34)+",8,
       1"+CHR$(34)+":Q$=CHR$(34
       )"
JR 90  S4$="30 FORN=1TO20:INPUT
       #2,P$(N),L$(N),J$(N),S$(
       N),R$(N):IFJ$(N)="+CHR$(
       34)
XA 100  S4$=S4$+"1"+CHR$(34)+"T
        HENE$(N)=M$"
MJ 110  S5$="40 NEXT:CLOSE2:PRI
        NTF$:FORN=1TO20:PRINTN;
        TAB(4)
HG 120  S5$=S5$+";P$(N):NEXT:IN
        PUT"+CHR$(34)+"
        {7 RIGHT}{DOWN}"+CHR$(3
        4)+";K"
EP 130  S6$="50 PRINT"+CHR$(34)
        +"{CLR}{3 DOWN}LOAD"+CH
        R$(34)+";Q$;P$(K);Q$;E$
        (K):IFS$(K)>"
RP 140  S6$=S6$+CHR$(34)+"Ø"+CH
        R$(34)+"THENPRINT"+CHR$
        (34)+"{4 DOWN}SYS"+CHR$
        (34)
MD 150  S6$=S6$+";S$(K):X=X+1"
FD 160  S7$="60 IFR$(K)="+CHR$(
        34)+"Y"+CHR$(34)+"THENP
        RINT"+CHR$(34)+"
        {4 DOWN}RUN"
KJ 170  S7$=S7$+CHR$(34)+":X=X+
        1:IFS$(K)>"+CHR$(34)+"Ø
        "+CHR$(34)+"THENPRINT"
RD 180  S7$=S7$+CHR$(34)+"
        {2 UP}RUN"+CHR$(34)
XB 190  S8$="70 FORZ=ØTOX:B=631
        +Z:POKEB,13:POKE198,Z:N
        EXT:PRINT"+CHR$(34)+"
        {HOME}"
EP 200  S8$=S8$+CHR$(34)
QF 210  PRINT"{CLR}{WHT}
        {3 DOWN}{14 SPACES}MENU
         SYSTEM"
KF 220  PRINT"{5 DOWN}
        {8 SPACES}{RVS} F1
        {OFF}...CREATE MENU FIL
        E"
PG 230  PRINT"{DOWN}{8 SPACES}
        {RVS} F2 {OFF}...LOAD M
        ENU FILE"
PG 240  PRINT"{DOWN}{8 SPACES}
        {RVS} F3 {OFF}...EDIT M
        ENU FILE"
CG 250  PRINT"{DOWN}{8 SPACES}
        {RVS} F4 {OFF}...SAVE M
        ENU FILE"
BR 260  PRINT"{DOWN}{8 SPACES}
        {RVS} F5 {OFF}...WRITE
        {SPACE}MENU DRIVER"
AJ 270  GETK$:IFK$=CHR$(133)THE
        NGOSUB560:GOTO390
GH 280  IFK$=CHR$(137)THEN330
AR 290  IFK$=CHR$(134)THEN390
QQ 300  IFK$=CHR$(138)THEN360
JF 310  IFK$=CHR$(135)THEN600
AJ 320  GOTO270
MJ 330  GOSUB570:OPEN2,8,2,"Ø:.
        M,S,R"
CP 340  FORN=1TO20:INPUT#2,P$(N
        ),L$(N),E$(N),S$(N),R$(
        N):NEXT
QF 350  CLOSE2:GOTO390
PP 360  GOSUB570:OPEN2,8,2,"@Ø:
        .M,S,W"
XP 370  FORN=1TO20:PRINT#2,P$(N
        )C$L$(N)C$E$(N)C$S$(N)C
        $R$(N)C$:NEXT
BC 380  CLOSE2:GOTO210
XJ 390  PRINT"{CLR}{RVS}EDIT OP
        TION{DOWN}":FORN=1TO9:P
        RINTN;LEFT$(P$(N),16):N
        EXT
SQ 400  PRINT"1Ø ";LEFT$(P$(1Ø)
        ,16);"{1Ø UP}":FORN=11T
        O2Ø:PRINTTAB(19)N;
PC 410  PRINTLEFT$(P$(N),16):NE
        XT:PRINT"{DOWN} {RVS}E
        {OFF} TO EDIT OR {RVS}R
        ETURN{OFF} FOR MENU"
CK 420  GETK$:IFK$=""THEN420
SF 430  IFK$=CHR$(13)THEN210
BK 440  IFK$<>"E"THEN420
EP 450  PRINT"{HOME}{13 DOWN}
        {39 SPACES}{UP}"
MR 460  INPUT"WHICH DIGIT TO ED
        IT";N$:N=VAL(N$):IFN<1O
        RN>20THEN450
EF 470  INPUT"{DOWN}ENTER PROGR
        AM NAME";P$(N):P$(N)=LE
        FT$(P$(N),35)
JD 480  INPUT"{DOWN}ENTER BOOT
        {SPACE}NAME";L$(N):L$(N
        )=LEFT$(L$(N),16)
BD 490  INPUT"{DOWN}ENTER {RVS}
         1 {OFF} FOR 8,1 OR
        {RVS} Ø {OFF} FOR ,8";E
        $(N)
MX 500  INPUT"{DOWN}ENTER SYS#
        {SPACE}OR RETURN";S$(N)
JH 510  IF S$(N)=CHR$(13)ORS$(N
        )=""THENS$(N)=" ":GOTO5
        3Ø
SA 520  IFS$(N)=""THENS$(N)=" "
MC 530  INPUT"{DOWN}RUN COMMAND
         NEEDED";R$(N):R$(N)=LE
        FT$((R$(N)),1)
SG 540  IFR$(N)<>"N"ANDR$(N)<>"
        Y"THENPRINT"{3 UP}":GOT
        O53Ø
KK 550  GOTO39Ø
AA 560  FORN=1TO2Ø:P$(N)=" ":L$
        (N)=" ":E$(N)=" ":S$(N)
        =" ":R$(N)=" ":NEXT:RET
        URN
SM 570  PRINT"{CLR}{8 DOWN}
        {6 RIGHT}{RVS}";M$
MF 580  GETK$:IFK$=""THEN58Ø
AP 590  RETURN
EK 600  PRINT"{CLR}{3 DOWN}NEW
        {2 DOWN}"
RR 610  PRINTS1$:PRINTS2$:PRINT
        S3$:PRINTS4$:PRINTS5$:P
        RINTS6$:PRINTS7$:PRINTS
        8$
CC 620  PRINT"SAVE"+CHR$(34)+"@
        Ø:MENU"+CHR$(34)+",8:SY
        S64738{HOME}"
ME 630  FORZ=ØTO1Ø:POKE631+Z,13
        :POKE198,Z+1:NEXT:END
```

Math Attack

(Article on page 56.)

```
EP 10  FOR I=54272TO54296:POKEI
       ,Ø:NEXT
AH 20  POKE51,127:POKE52,62:POK
       E55,127:POKE56,62:CLR:GO
       TO4Ø
RB 30  PRINT"{HOME}";:FORC=1TO2
       1:PRINTC$:NEXT:PRINT"
       {HOME}":RETURN
AP 40  C$="{39 SPACES}":PRINTCH
       R$(142);CHR$(8)
ER 50  PRINT"{CLR}{21 DOWN}
       {BLU}";:POKE5328Ø,Ø:POKE
       53281,Ø:V=53248:POKE5428
       1,75
QE 60  POKE54296,15:F=54273:AD=
       F+4:SR=F+5:W=F+3:F2=F+7:
       W2=W+7:POKEW+8,Ø:POKEW+9
       ,24Ø
MM 70  D$="{HOME}{24 DOWN}":TB$
       =D$+"{2Ø RIGHT}"
BB 80  PRINT"{3 SPACES}【P】【O】
       {2 SPACES}【I】 【@】
       {2 SPACES}【O】 【I】
       {2 SPACES}【@】【P】【O】【I】
       【I】【O】【P】【@】{3 SPACES}
       【O】【@】【P】 【P】【@】【I】
       {2 SPACES}【@】"
FM 90  PRINT"{RVS}{RIGHT} 【T】
       {2 SPACES}【U】【Y】 【U】 【Y】
       【U】 【T】 【U】【T】{4 SPACES}
       【T】{4 SPACES}【U】【Y】【T】
       {3 SPACES}【U】{3 SPACES}
       【Y】【U】 "
MH 100  PRINT"{RVS}{RIGHT}
        {38 SPACES}":BC=PEEK(V+
        31)
MM 110  PRINT"{HOME}{2 DOWN}
        {RIGHT}{PUR}TYPE: {RVS}
        +{OFF} (ADDITION)"
XD 120  PRINT"{2 DOWN}{7 RIGHT}
        {RVS}-{OFF} (SUBTRACTIO
        N)"
DR 130  PRINT"{2 DOWN}{7 RIGHT}
        {RVS}X{OFF} (MULTIPLICA
        TION)"
QK 140  GETT$:IFT$<>"+"ANDT$<>"
        -"ANDT$<>"X"THEN14Ø
GM 150  GOSUB3Ø:DIMN(1Ø,7):POKE
        56334,PEEK(56334)AND254
        :POKE1,PEEK(1)AND251
EK 160  CS=53632:FORX=ØTO9:FORY
        =ØTO7:N(X,Y)=PEEK(CS):C
        S=CS+1:NEXTY,X
QH 170  SN=-5344Ø*(T$="X")-5359
        2*(T$="+")-536Ø8*(T$="-
        ")
HF 180  FORY=ØTO7:N(1Ø,Y)=PEEK(
        SN):SN=SN+1:NEXT
CA 190  POKE1,PEEK(1)OR4:POKE56
        334,PEEK(56334)OR1
SR 200  PRINT"{2 DOWN}{2 RIGHT}
        {YEL}LEVEL:{3 SPACES}
        {RVS}1{6 RIGHT}2
        {6 RIGHT}3"
DX 210  PRINT"{2 DOWN}(LIMITS)
        {3 SPACES}9{5 SPACES}99
        {5 SPACES}99"
JF 220  PRINT"{9 SPACES}"T$" 9
        {4 SPACES}"T$" 9
        {4 SPACES}"T$"99"
FA 230  PRINT"{9 SPACES}【3 T】
        {4 SPACES}【3 T】
        {4 SPACES}【3 T】"
CM 240  GETL$:IFL$<>"1"ANDL$<>"
        2"ANDL$<>"3"THEN24Ø
KD 250  L=VAL(L$):L1=-9*(L=1)-9
        9*(L=2)-99*(L=3):L2=-9*
        (L=1)-9*(L=2)-99*(L=3)
RM 260  GOSUB3Ø:POKEV,1ØØ:POKEV
        +2,175:POKEV+4,25Ø:POKE
        V+1,Ø:POKEV+3,Ø:POKEV+5
        ,Ø
SP 270  FORX=16ØØØTO16191:READD
        :POKEX,D:NEXT:POKEV+39,
        5:POKEV+4Ø,3:POKEV+41,8
MR 280  DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø
GQ 290  DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø
EQ 300  DATAØ,Ø,Ø,16,Ø,Ø,74,Ø
BF 310  DATA2,82,64,2,74,64,2,8
        2
RE 320  DATA64,17,36,136,9,36,1
        44,41
DS 330  DATA36,148,36,145,36,18
        ,137,72
FC 340  DATA73,82,146,36,137,36
        ,18,82
```

```
JG 35Ø  DATA72,17Ø,137,85,Ø,84,
        Ø,Ø
KS 36Ø  DATAØ,17Ø,Ø,5,85,64,21,
        85
QR 37Ø  DATA8Ø,42,17Ø,168,85,85
        ,84,85
HE 38Ø  DATA85,84,85,85,84,42,1
        7Ø,168
PB 39Ø  DATA21,85,8Ø,5,85,64,Ø,
        17Ø
FG 4ØØ  DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø
BF 41Ø  DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø
RF 42Ø  DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø
KE 43Ø  DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø
EQ 44Ø  DATA96,Ø,1,96,64,2,16,3
        5
RP 45Ø  DATA4,15,144,24,1Ø,126,
        48,4
SP 46Ø  DATA173,96,4,3Ø,18,7Ø,3
        5,192
PR 47Ø  DATA53,65,72,36,129,4Ø,
        2Ø,129
HF 48Ø  DATA72,28,129,192,36,2,
        66,124
JF 49Ø  DATA1Ø2,98,162,12Ø,8Ø,5
        3,144,152
DK 5ØØ  DATA1,73,48,2,6,8Ø,4,2
EA 51Ø  DATA2ØØ,8,1,4,144,Ø,Ø,Ø
JJ 52Ø  FORX=16192TO16384:POKEX
        ,Ø:NEXT
ME 53Ø  POKE2Ø4Ø,253:POKE2Ø41,2
        54:POKE2Ø42,255
GB 54Ø  FORSN=ØTO2:GOSUB1Ø8Ø:NE
        XT:POKEV+21,7:POKEV+31,
        Ø:GOSUB3Ø:POKE198,Ø
QF 55Ø  Y(Ø)=Ø:Y(1)=3Ø:Y(2)=1Ø:
        PRINTD$"ANSWERS:ƐPƷ";TA
        B(19);"BOMBS YOU MISSED
        : Ø";
RX 56Ø  FORX=ØTO2:Y(X)=Y(X)+1:N
        EXT
BS 57Ø  POKEV+1,Y(Ø):POKEV+3,Y(
        1):POKEV+5,Y(2):BC=PEEK
        (V+31)
PC 58Ø  POKEW2,17:POKEF2,Y(Ø):P
        OKEF2,Y(1):POKEF2,Y(2):
        POKEW2,Ø
SC 59Ø  FORD=ØTO(L-1)*1ØØ:NEXT
AM 6ØØ  GETN$:IF(N$=>"Ø"ANDN$<=
        "9")ORN$="-"ORN$=CHR$(2
        Ø)ORN$=CHR$(13)THEN7ØØ
AS 61Ø  BC=PEEK(V+31)AND7:IFBC=
        ØTHEN56Ø
CC 62Ø  IF(BCAND1)=1THENSN=Ø:GO
        SUB66Ø
AG 63Ø  IF(BCAND2)=2THENSN=1:GO
        SUB66Ø
JQ 64Ø  IF(BCAND4)=4THENSN=2:GO
        SUB66Ø
XX 65Ø  GOTO56Ø
EX 66Ø  BH=BH+1:POKEV+1+SN*2,22
        1:POKE2Ø4Ø+SN,25Ø:POKEW
        ,Ø:POKEAD,12:POKEF,1
RB 67Ø  POKEW,129:FORD=1TO1ØØ:N
        EXT:POKEW,Ø:POKEW,129:P
        OKE2Ø4Ø+SN,251:FORD=1TO
        1ØØ
PB 68Ø  NEXT:GOSUB1Ø8Ø:PRINTTB$
        ;TAB(36)BH;:Y(SN)=3Ø:IF
        BH>2THEN89Ø
EJ 69Ø  POKE2Ø4Ø+SN,253+SN:RETU
        RN
MD 7ØØ  IFN$=CHR$(2Ø)ANDTB>ØTHE
        NTB=TB-1:PRINT"
        {2 LEFT}ƐPƷ{LEFT}";:AN$
        =LEFT$(AN$,LEN(AN$)-1)
XD 71Ø  IFN$=CHR$(2Ø)THEN61Ø
JG 72Ø  IFN$<>"-"THEN75Ø
XQ 73Ø  IFTB=ØTHENTB=1:AN$="-":
        PRINTLEFT$(TB$,33)"-ƐPƷ
        {LEFT}";
ED 74Ø  GOTO61Ø
JD 75Ø  IFN$=CHR$(13)THEN78Ø
DG 76Ø  IFTB<4THENPRINTLEFT$(TB
        $,33+TB);N$;"ƐPƷ{LEFT}"
        ;:TB=TB+1:AN$=AN$+N$
JF 77Ø  GOTO61Ø
KD 78Ø  AN=VAL(AN$):PRINTD$;"
        {8 RIGHT}ƐPƷ{6 SPACES}"
        ;:AN$="":TB=Ø
XP 79Ø  CF=Ø:FORXX=ØTO2:IFA(XX)
        =ANANDY(XX)<221THENGOSU
        B86Ø
AA 8ØØ  NEXT:IFTC>=2ØTHEN1ØØØ
PJ 81Ø  IFCF>ØTHEN85Ø
QB 82Ø  WT=WT+1:FORX=ØTO2:Y(X)=
        Y(X)+(3-L)*1Ø:IFY(X)>2Ø
        ØTHENY(X)=2ØØ
RM 83Ø  NEXT:L=L-1:POKEW,Ø:POKE
        AD,Ø:POKESR,24Ø
AP 84Ø  POKEW,17:FORD=255TO1ØØS
        TEP-2:POKEF,D:NEXT:POKE
        W,Ø:POKESR,Ø
FJ 85Ø  GOTO61Ø
BH 86Ø  TC=TC+1:AT=AT+1:POKE2Ø4
        Ø+XX,252:SN=XX:POKEW,Ø:
        POKEAD,144:POKEF,25Ø
EG 87Ø  POKEW,129:FORD=1TO1ØØ:N
        EXT:POKE2Ø4Ø+XX,253+XX:
        GOSUB1Ø8Ø:Y(XX)=3Ø:CF=1
AR 88Ø  BC=PEEK(V+31):RETURN
HD 89Ø  POKEV+1,Ø:POKEV+3,Ø:POK
        EV+5,Ø
DB 9ØØ  PRINT"{CLR}{3 DOWN}
        {5 RIGHT}{RED}MATH PROB
        LEMS INVADE THE CITY":G
        OTO93Ø
MH 91Ø  PRINT"{CLR}{3 DOWN}
        {5 RIGHT}{RED}YOU'VE BE
        ATEN THE MATH PROBLEMS"
RK 92Ø  PRINT"{2 DOWN}{3 RIGHT}
         {9 RIGHT}AT THEIR OWN
        {SPACE}GAME"
CA 93Ø  PRINT"{2 DOWN}{3 RIGHT}
        Ɛ6ƷGAME RESULTS:":PRINT
        "{DOWN}{4 RIGHT}{GRN}RI
        GHT ANSWERS:"AT
MH 94Ø  PRINT"{DOWN}{6 RIGHT}
        {PUR}WRONG ANSWERS:"WT
XQ 95Ø  PRINT"{DOWN}{8 RIGHT}
        Ɛ1ƷBOMBS REACHING THE C
        ITY:"BH
GP 96Ø  PRINT"{3 DOWN}Ɛ7Ʒ
        {3 RIGHT}DO YOU WANT AN
        OTHER CHANCE? (Y/N)?"
HX 97Ø  GETC$:IFC$<>"Y"ANDC$<>"
        N"THEN97Ø
MG 98Ø  IFC$="Y"THENRUN
RP 99Ø  PRINT"{CLR}Ɛ7Ʒ";:POKE53
        28Ø,14:POKE53281,6:POKE
        V+21,Ø:END
BS 1ØØØ FORX=ØTO2:Y(X)=Ø:POKEV
         +1+2*X,Ø:NEXT:IFL1+L2>
         1ØØØTHEN91Ø
HK 1Ø1Ø PRINT"{HOME}{3 DOWN}
         {6 RIGHT}{PUR}"TC"CORR
         ECT ANSWERS":TC=Ø
MM 1Ø2Ø PRINT"Ɛ3Ʒ{DOWN}
         {3 RIGHT}YOU MAY GO ON
          . . . ."
GX 1Ø3Ø PRINT"{DOWN}{4 RIGHT}
         {17 SPACES}TO THE NEXT
          LEVEL"
RD 1Ø4Ø L1$=STR$(L1):L2$=STR$(
         L2):IFL2<L1THENL2$=L2$
         +"9":GOTO1Ø6Ø
GK 1Ø5Ø L1$=L1$+"9"
QR 1Ø6Ø L1=VAL(L1$):L2=VAL(L2$
         ):PRINT"{2 DOWN}
         {4 RIGHT}{YEL}YOUR NEW
          LIMITS ARE: "L1"AND"L
         2
EA 1Ø7Ø GOTO53Ø
CM 1Ø8Ø U$=STR$(INT(RND(Ø)*L1)
         ):B$=STR$(INT(RND(Ø)*L
         2)):B$=RIGHT$(B$,2)
HP 1Ø9Ø IFT$="X"THENA(SN)=VAL(
         U$)*VAL(B$)
KK 11ØØ IFT$="+"THENA(SN)=VAL(
         U$)+VAL(B$)
JD 111Ø IFT$="-"THENA(SN)=VAL(
         U$)-VAL(B$)
BE 112Ø B$=T$+B$:U$=RIGHT$(U$,
         3):SY=SN*2+1+V:IFLEN(U
         $)=2THENU$=" "+U$
QK 113Ø SS=(SN+253)*64:POKEW2,
         Ø:POKEW2,65:FORY=ØTO2:
         C1$=MID$(U$,Y+1,1)
BA 114Ø FORX=ØTO7:POKESS+X*3+Y
         ,N(VAL(C1$),X):IFC1$=C
         HR$(32)THENPOKESS+X*3+
         Y,Ø
AA 115Ø POKESY,ABS(PEEK(SY)-3)
         :POKEF2,1ØØ-1Ø*X:NEXTX
         ,Y
XM 116Ø FORY=ØTO2:C1$=MID$(B$,
         Y+1,1):FORX=8TO15
AH 117Ø POKESS+X*3+Y,N(VAL(C1$
         ),X-8):POKEF2,1Ø*X:POK
         ESY,ABS(PEEK(SY)-3)
FD 118Ø IFC1$=" "THENPOKESS+X*
         3+Y,Ø
XJ 119Ø IFC1$=T$THENPOKESS+X*3
         +Y,N(1Ø,X-8)
KP 12ØØ NEXTX,Y:POKEW2,Ø
XE 121Ø POKESS+48,255:POKESS+4
         9,255:POKESS+5Ø,255
XB 122Ø FORX=SS+51TOSS+62:POKE
         X,Ø:NEXT:RETURN
```

Sequential File Converter For *SpeedScript*

See instructions in article on page 76 before typing in.

```
Ø8Ø1:25 Ø8 64 ØØ 53 4E B2 32 9D
Ø8Ø9:37 31 36 3A 53 4C B2 32 CE
Ø811:37 33 36 3A 44 4E B2 32 E6
Ø819:37 33 39 3A 44 4C B2 32 47
Ø821:37 35 39 ØØ 48 Ø8 6E ØØ 8Ø
Ø829:99 22 93 11 11 Ø5 22 A6 99
Ø831:31 32 29 22 43 4F 44 45 D2
Ø839:2Ø 43 4F 4E 56 45 52 53 B8
Ø841:49 4F 4E 11 11 22 ØØ 7E 34
Ø849:Ø8 78 ØØ 99 22 5Ø 4C 45 45
Ø851:41 53 45 2Ø 49 4E 5Ø 55 FA
Ø859:54 2Ø 53 4F 55 52 43 45 BA
Ø861:2Ø 5Ø 52 4F 47 52 41 4D 28
Ø869:27 53 2Ø 44 49 53 4B 2Ø 78
Ø871:4E 41 4D 45 3A 22 3A 85 4B
Ø879:2Ø 53 4E 24 ØØ B1 Ø8 82 D3
Ø881:ØØ 8B 2Ø C3 28 53 4E 24 Ø4
Ø889:29 B1 31 36 2Ø A7 2Ø 99 9D
Ø891:22 54 48 41 54 2Ø 4E 41 E5
Ø899:4D 45 2Ø 49 53 2Ø 54 4F 4D
Ø8A1:4F 2Ø 4C 4F 4E 47 2E 2E F9
Ø8A9:2E 22 3A 89 31 32 3Ø ØØ EB
Ø8B1:C4 Ø8 8C ØØ 53 4E 24 B2 86
Ø8B9:53 4E 24 AA 22 2C 53 2C CA
Ø8C1:52 22 ØØ F2 Ø8 96 ØØ 81 CE
Ø8C9:2Ø 49 B2 3Ø 2Ø A4 2Ø C3 2D
Ø8D1:28 53 4E 24 29 AB 31 3A 6B
Ø8D9:97 2Ø 53 4E AA 49 2C C6 A6
Ø8E1:28 CA 28 53 4E 24 2C 49 97
Ø8E9:AA 31 2C 31 29 29 3A 82 19
Ø8F1:ØØ 28 Ø9 AØ ØØ 99 22 11 F2
Ø8F9:5Ø 4C 45 41 53 45 2Ø 49 3B
Ø9Ø1:4E 5Ø 55 54 2Ø 44 45 53 2E
Ø9Ø9:54 49 4E 41 54 49 4F 4E 2A
Ø911:2Ø 5Ø 52 4F 47 52 41 4D D9
Ø919:27 53 2Ø 4E 41 4D 45 22 68
Ø921:3A 85 2Ø 44 4E 24 ØØ 5B 58
Ø929:Ø9 AA ØØ 8B 2Ø C3 28 44 C7
Ø931:4E 24 29 B1 31 36 2Ø A7 FD
Ø939:2Ø 99 22 54 48 41 54 2Ø 5B
Ø941:4E 41 4D 45 2Ø 49 53 2Ø B5
Ø949:54 4F 4F 2Ø 4C 4F 4E 47 C8
```

```
0951:2E 2E 2E 22 3A 89 31 36 7E
0959:30 00 6E 09 B4 00 44 4E 5E
0961:24 B2 44 4E 24 AA 22 2C DB
0969:50 2C 57 22 00 9C 09 BE FE
0971:00 81 20 49 B2 30 20 A4 B7
0979:20 C3 28 44 4E 24 29 AB D6
0981:31 3A 97 20 44 4E AA 49 A9
0989:2C C6 28 CA 28 44 4E 24 28
0991:2C 49 AA 31 2C 31 29 29 16
0999:3A 82 00 C3 09 C8 00 97 A8
09A1:20 53 4C 2C C3 28 53 4E 98
09A9:24 29 AA 32 3A 97 20 44 45
09B1:4C 2C C3 28 44 4E 24 29 BC
09B9:AA 32 3A 9E 20 32 35 30 43
09C1:31 00 00 00 A2 08 A9 0F 04
09C9:A8 20 BA FF A9 02 A2 C8 F2
09D1:A0 0A 20 BD FF 20 C0 FF 98
09D9:A9 0F 20 C3 FF AD B0 0A E6
09E1:A2 9A A0 0A 20 BD FF A9 42
09E9:02 A2 08 A8 20 BA FF 20 3D
09F1:C0 FF AD C7 0A A2 B1 A0 75
09F9:0A 20 BD FF A9 03 A2 08 77
0A01:A8 20 BA FF 20 C0 FF A9 76
0A09:00 85 90 A2 03 20 C9 FF E6
0A11:A9 01 20 D2 FF A9 08 20 42
0A19:D2 FF 20 CC FF A2 02 20 16
0A21:C6 FF 20 CF FF 85 FB A4 4C
0A29:90 84 FC 20 CC FF A5 FB F5
0A31:C9 0D D0 04 A9 1F D0 3E 71
0A39:C9 20 B0 04 A9 3F D0 36 B2
0A41:C9 40 90 32 C9 60 B0 06 B6
0A49:38 E9 40 4C 77 0A C9 80 B8
0A51:B0 06 38 E9 20 4C 77 0A 10
0A59:C9 A0 B0 05 A9 3F 4C 77 3B
0A61:0A C9 C0 B0 06 38 E9 40 35
0A69:4C 77 0A C9 FF D0 04 A9 54
0A71:5E D0 03 38 E9 80 85 FB 25
0A79:A2 03 20 C9 FF A5 FB 20 EE
0A81:D2 FF 20 CC FF A4 FC C0 1D
0A89:00 F0 92 A9 02 20 C3 FF DE
0A91:A9 03 20 C3 FF 20 CC FF 95
0A99:60 30 3A 20 20 53 42 2E 34
0AA1:44 49 53 4B 20 52 45 50 6E
0AA9:4F 52 54 2C 53 2C 52 14 4B
0AB1:30 3A 20 20 53 53 20 44 DE
0AB9:49 53 4B 20 52 45 50 4F 4A
0AC1:52 54 2C 50 2C 57 14 49 CE
0AC9:30 00 00 00 00 00 00 00 F5
```

Mastering Sound & Music On The 128

(Article on page 78.)

Program 1: The Magic Flute

```
MR 10  VOL 8:TEMPO 13:PLAY"X0U9
       ":A$="V1O5IGIFQEV2O4QGV3
       QCQCV2QGV1O5QEQRQE"
FR 20  B$="V1O5QFV2O4QGV3O3QBQB
       V2O4QGV1O5QFQRIFIE"
AA 30  C$="V1O5QDV2O4QGV3O3QBQB
       V2O4QGV1O5QDQRQD"
HH 40  D$="V1O5HEV2O4QGV3QCQCV2
       QGV1QRO5IGIF":E$="V1O5HE
       V2O4HGV3HCHCV2HGV1O5QEQE
       "
JJ 50  F$="V1O5.QFV2O4WAV3WFV1O
       5IGQAQF":G$="V1O5QEV2O4H
       GV1O5QEQDV2O4HGV1O5QD"
HF 60  H$="V1O5HCV2O4QCQEQCV1QR
       "
QR 70  FOR T=0 TO 9:PRINT USING
       "{CLR}{7 DOWN}{TAB}
       {LEFT}THIS IS ENVELOPE N
       UMBER #";T
HQ 80  READ EN$(T):PRINT USING"
       {7 DOWN}{TAB}{5 SPACES}#
       #############";EN$(T);:T
       $=STR$(T)
JK 90  PLAY"V1T":PLAY T$:PLAY"V
       2T":PLAY T$:PLAY"V3T":PL
       AY T$
GK 100 PLAY A$:PLAY B$:PLAY C$
       :PLAY D$:PLAY E$:PLAY F
       $:PLAY G$:PLAY H$:NEXT:
       END
BK 110 DATA "{3 SPACES}PIANO",
       " ACCORDION","
       {2 SPACES}CALLIOPE","
       {4 SPACES}DRUM","
       {3 SPACES}FLUTE","
       {3 SPACES}GUITAR","HARP
       SICHORD","{3 SPACES}ORG
       AN","{2 SPACES}TRUMPET"
       ," XYLOPHONE"
```

Program 2: American Patrol

```
JD 10  PRINT"{CLR}{5 DOWN}{TAB}
       {4 SPACES}AMERICAN PATRO
       L":PRINT"{5 DOWN}{2 TAB}
       {3 SPACES}BY":PRINT"
       {5 DOWN}{TAB}{5 SPACES}F
       . W. MEACHAM"
AJ 20  TEMPO 20:VOL 8:PLAY"X0U9
       ":PU$="V1O4IASBO5S#C"
RE 30  PLAY"V1T2V2T2V3T2":REM *
       **
XJ 40  A$="V1O5.IDV3O3QDO4S#FV2
       SAV1O5.IDV2SRO4SAV3S#FSR
       "
KX 50  B$="V3O3QAV1O5IDI#CIDV2O
       4QAV3Q#FV1O5IE"
DJ 60  C$="V1O5.I#FV3O3QDO4S#FV
       2SAV1O5.I#FV2SRO4SAV3S#F
       SR"
GA 70  D$="V3O3QAV1O5I#FIFI#FV3
       O4Q#FV2QAV1O5IG"
HC 80  E$="V1O5.IAV3O3QDO4S#FV2
       SAV1O5.IAV2SRO4SAV3S#FSR
       "
QH 90  F$="V3O3QAV1O5IAI#GIAV3O
       4Q#FV2QAV1O6ID"
CM 100 G$="V1O5HAV3O3QDQ#FV2O4
       SASRSASR":H$="V2QRV3O3Q
       GQ#FV2O4QAV1O5.I#F"
KG 110 I$="V3O3QAV1O5.IGV3O4Q#
       CV2QGV1O5IGI#F":J$="V1O
       5.IEV3O3QEO4Q#CV2QGV1O5
       .IG"
DP 120 K$="V3O3QDV1O5.I#FV3O4Q
       #FV2QAV1O5I#FIE":L$="V1
       O5.IDV3O3QAO4Q#FV2QAV1O
       5.I#F"
XD 130 M$="V3O4IEV1O5.IEV3O4I#
       FIEV2Q#GV1.IBV3ID":N$="
       V3O4Q#CV1O5.I#CV3O3QBV2
       O4Q#GV1O5.ID"
DD 140 O$="V3O3IAV1O5HEV3O3IBI
       AV2O4QGV3O3IG":P$="V3O3
       Q#FQEV2O4QGV1O4IASBO5S#
       C"
QQ 150 W$="V1O5HAV3O3.IDSRO3QD
       V2O4QA":X$="V3O3QEQ#FV2
       O4QAV1O5.I#F"
QH 160 Y$="V3O3QGV1O5.IBSR.IAV
       3O4SDV2SBSRSBV3SDSR":Z$
       ="V3O3QDV1O5.IGV3O4QGV2
       QBV1O5.I#F"
SK 170 AA$="V3O3QEV1O5.IESR.ID
       V3O4S#GV2SBSRSBV3S#GSR"
       :AB$="V3O3Q#GV1O5.I#CSR
       .IDV3O4QEV2QB"
RM 180 AC$="V3O3QAV1O5QEI#FV3O
       4Q#CV2QEV1O5IG":AD$="V1
       O5.I#FV3O2QAO4Q#CV2QGV1
       O5.IE"
PX 190 AE$="V3O3QDV1O5.IDV3O4Q
       DV2Q#FV1IAIB":AF$="V1O5
       I#CV3O3QAV1O5IDIEV3O4QD
       V2Q#FV1O5I#F"
JJ 200 AG$="V1O5.IGV3O3QAO4S#C
       V2SEV1O5.IGV2SRO4SEV3S#
       CSR"
BB 210 AH$="V3O3QEV1O5.IGV3O4Q
       #CV2QEV1O5.IG"
AR 220 AI$="V3O3QAV1O5IGI#FHGV
       3O4S#CV2SESRSEV3S#CSR":
       AJ$="V3O3QEO4Q#CV2QAV1O
       5Q#G"
XM 230 AK$="V1O5.IAV3O3QDO4SDV
       2S#FV1O5.IAV2SRO4S#FV3S
       DSR"
DA 240 AL$="V3O3QAV1O5IAIGI#FV
       3O4QDV2QAV1O5IG":AM$="V
       1O5HAV3O3QDO4SDV2S#FSRS
       #FV3SDSR":AN$="V3O3Q#FO
       4QDV2QAV1O5Q#F"
KR 250 AV$="V3O3Q#FO4QDV2QAV1O
       5QA":AW$="V1O5.IBV3O3QG
       O4SDV2SGV1O5.IBV2SRO4SG
       V3SDSR"
SR 260 AX$="V3O2QBV1O6.QDV3O4Q
       DV2QGV1O5IB"
PE 270 AY$="V1O5.IAV3O3QDO4SDV
       2S#FV1O5.IAV2SRO4S#FV3S
       DSR":AZ$="V3O3QAV1O5IAI
       G.I#FV3O4QDV2QA"
BK 280 BA$="V3O3Q#FV1O5.IGV3O4
       SEV2S#AV1O5.IGV2SRO4S#A
       V3SESR"
JB 290 BB$="V3O3Q#CV1O5.I#CV3O
       4Q#FV2Q#AV1O5.IG":BC$="
       V3O2QBV1O5.I#FV3O4SDV2S
       BV1O5.I#FV2SRO4SBV3SDSR
       "
HR 300 BD$="V3O3Q#FV1O5I#FIE.I
       DV3O4Q#FV2QBV1SR":BH$="
       V3O3Q#FV1O5IAIG.I#FV3O4
       QDV2QA"
MF 310 BI$="V3O3QAV1O5.IGV3O4S
       #CV2SEV1O5.IGV2SRO4SEV3
       S#CSR"
AC 320 BJ$="V3O3QEV1O5.I#CV3O4
       Q#CV2QAV1O5.IE":BK$="V3
       O3QDV1O5.IDV3O4S#FV2SAV
       1O5.IDV2SRO4SAV3S#FSR"
CE 330 BL$="V3O3QAV2O4Q#FV1O5.
       IDSRS#FSRS#FSR":BM$="V1
       O5W#FV3QRO3I#FV2O4Q#A":
       BN$="V2O4QBV3O3I#GIRI#A
       V2O5Q#C"
PG 340 BO$="V2O5QDV3O3IBV1QRO5
       IDI#C":BP$="V1O5IDIEI#F
       IG":BQ$="V1O5WAV3QRO3IA
       V2O5Q#C"
JG 350 BR$="V2O5QDV3O3IBIRO4I#
       CV2O5QE":BS$="V2O5Q#FV3
       O4IDV1QRO5I#FI#E":BT$="
       V1O5I#FI#GIAIB"
MQ 360 BU$="V1O6W#CV3QRO4I#CV2
       O5Q#E":BV$="V2O5Q#FV3O4
       I#DIRI#EV2O5Q#G":BW$="V
       2O5QAV3O4I#FV1QRO5IAI#G
       "
KJ 370 BX$="V1O5IAIBO6I#CID":B
       Y$="V1O6WEV3QRO4S#GV2O6
       SDSRSDV3O4S#GSR":BZ$="V
       3O4I#GV2O6IDIRIDV3O4I#G
       IR"
AK 380 CA$="V3O4IAV2O6Q#CV1QEQ
       R":CB$="V1O2QAV3HAV2HA"
       :CC$="V3O3IAV1O5IDV2O4I
       #FIRI#FV1O5IDV3O3IAIR":
       CD$="V3O3IAV2O4I#FV1O5I
       D"
BK 390 FOR R=1TO2:PLAY PU$:PLA
       Y A$:PLAY B$:PLAY C$:PL
       AY D$:PLAY E$:PLAY F$:P
       LAY G$:PLAY H$
BE 400 PLAY I$:PLAY J$:PLAY K$
       :PLAY L$:PLAY M$:PLAY N
       $:PLAY O$:PLAY P$:Q$=A$
       :R$=B$:S$=C$:T$=D$
BE 410 PLAY Q$:PLAY R$:PLAY S$
       :PLAY T$:U$=E$:V$=F$:PL
       AY U$:PLAY V$:PLAY W$:P
       LAY X$:PLAY Y$:PLAY Z$:
         PLAY AA$:PLAY AB$
PP 42Ø PLAY AC$:PLAY AD$:PLAY
         {SPACE}AE$:PLAY AF$:PLA
         Y AG$:PLAY AH$:PLAY AI$
         :PLAY AJ$:PLAY AK$:PLAY
          AL$:PLAY AM$:PLAY AN$
PE 43Ø AO$=AG$:AP$=AH$:AQ$=AI$
         :AR$=AJ$:PLAY AO$:PLAY
         {SPACE}AP$:PLAY AQ$:PLA
         Y AR$:AS$=AK$:AT$=AL$:A
         U$=AM$
CF 44Ø PLAY AS$:PLAY AT$:PLAY
         {SPACE}AU$:PLAY AV$:PLA
         Y AW$:PLAY AX$:PLAY AY$
         :PLAY AZ$:PLAY BA$:PLAY
          BB$:PLAY BC$:PLAY BD$
DC 45Ø BE$=AW$:BF$=AX$:BG$=AY$
         :PLAY BE$:PLAY BF$:PLAY
          BG$:PLAY BH$:PLAY BI$:
         PLAY BJ$:IF R=2 THEN48Ø
GG 46Ø PLAY BK$:PLAY BL$:PLAY
         {SPACE}BM$:PLAY BN$:PLA
         Y BO$:PLAY BP$:PLAY BQ$
         :PLAY BR$:PLAY BS$:PLAY
          BT$
HJ 47Ø PLAY BU$:PLAY BV$:PLAY
         {SPACE}BW$:PLAY BX$:PLA
         Y BY$:PLAY BZ$:PLAY CA$
         :VOL 15:PLAY CB$:VOL 8
HB 48Ø NEXT R:PLAY CC$:PLAY CD
         $:END
```

Machine Language For Beginners

(Article on page 87.)

```
CC 1Ø H=Ø:REM IF H = Ø THEN AS
        SEMBLY IS IN DECIMAL
FJ 5Ø HE$="Ø123456789ABCDEF":S
        Z=1:ZO$="ØØØ"
JD 1ØØ PRINT"{3 SPACES}SIMPLE
         {3 SPACES}ASSEMBLER
         {2 SPACES}CONVENTIONS:"
RQ 11Ø DIMM$(56),TY(56),OP(56)
JK 12Ø FORI=1TO56:READM$(I)
PC 122 ROP$=MID$(M$(I),4,1):TY
         (I)=VAL(ROP$)
PS 124 OP$=RIGHT$(M$(I),3):OP(
         I)=VAL(OP$)
SC 126 M$(I)=LEFT$(M$(I),3)
PJ 14Ø NEXTI: PRINT
KB 15Ø PRINT"IMMEDIATE
         {5 SPACES}LDA #15
KS 155 PRINT"ABSOLUTE
         {6 SPACES}LDA 15ØØ
RM 16Ø PRINT"ZERO PAGE
         {5 SPACES}LDA 15
EJ 165 PRINT"ACCUMULATOR
         {3 SPACES}ASL
EJ 17Ø PRINT"INDIRECT X
         {4 SPACES}LDA (15X)
XB 175 PRINT"INDIRECT Y
         {4 SPACES}LDA (15)Y
GM 177 PRINT"ZERO PAGE X
         {3 SPACES}LDA 15X
PC 179 PRINT"ZERO PAGE Y
         {3 SPACES}LDX 15Y
CQ 18Ø PRINT"ABSOLUTE X
         {4 SPACES}LDA 15ØØX
JH 185 PRINT"ABSOLUTE Y
         {4 SPACES}LDA 15ØØY
FP 189 PRINT:PRINT"{4 SPACES}E
         NTER ALL NUMBERS IN ";
HE 19Ø IFH=1 THENPRINT"HEX":GO
         TO2ØØ
AC 195 PRINT"DECIMAL"
HM 2ØØ PRINT:PRINT"PLEASE INPU
         T STARTING ADDRESS FOR
         {SPACE}ML PROGRAM":INPU
         T SA$
BA 21Ø IFH=1THENH$=SA$:GOSUB5Ø
         ØØ:SA=DE:GOTO22Ø
XE 215 SA=VAL(SA$)
GR 22Ø TA=SA:PRINT"{CLR}":REM
         {SPACE}CLEAR THE SCREEN
FX 23Ø IFH=1THENDE=SA:SZ=3:GOS
         UB4ØØØ:PRINTH$;:GOTO24Ø
PQ 235 PRINTSA" ";
KJ 24Ø INPUTMN$:PRINT"{UP}"SPC
         (2Ø);:REM GO UP ONE LIN
         E AND OVER 2Ø SPACES
KH 241 REM ADD NEW PSEUDO-OPS
         {SPACE}HERE
DB 242 IFRIGHT$(MN$,7)="FORWAR
         D"THENFB=SA
RH 243 IFRIGHT$(MN$,7)="RESOLV
         E"THENFR=SA-FB:POKEFB+1
         ,FR-2:PRINT"{2 SPACES}O
         K":GOTO23Ø
XK 244 IFRIGHT$(MN$,4)="POKE"T
         HENPRINT"ADDR,NUMBER(DE
         C)";
CH 245 IFRIGHT$(MN$,4)="POKE"T
         HENINPUTADR,NUM:POKEADR
         ,NUM:GOTO23Ø
XA 25Ø IFMN$="END"THENPRINT:PR
         INT"{6 SPACES}PROGRAM I
         S FROM"TA"TO"SA:END
BG 26Ø L=LEN(MN$):L$=LEFT$(MN$
         ,3)
QS 27Ø FORI=1TO56:IFL$=M$(I)TH
         EN3ØØ
EH 28Ø NEXTI
FK 29Ø GOTO85Ø
KK 3ØØ REM PRIMARY OPCODE CATE
         GORIES
CH 3Ø1 TY=TY(I):OP=OP(I)
XR 3Ø5 IFFB=SATHENTN=Ø:GOTO2Ø1
         Ø
SB 31Ø IFTY=ØTHENGOTO1ØØØ
ER 32Ø IFTY=3THENTY=1:IFL=3THE
         NOP=OP+8:GOTO1ØØØ
RX 33Ø R$=RIGHT$(MN$,L-4):IFH=
         1THENGOSUB6ØØØ
GK 34Ø LR$=LEFT$(R$,1):LL=LEN(
         R$):IFLR$="#"THEN48Ø
JK 35Ø IFLR$="("THEN52Ø
FK 36Ø IFTY=8THEN6ØØ
KC 37Ø IFTY=3THENOP=OP+8:GOTO1
         ØØØ
RM 38Ø IFRIGHT$(R$,1)="X"ORRIG
         HT$(R$,1)="Y"THEN63Ø
GG 39Ø IFLEFT$(L$,1)="J"THEN82
         Ø
SQ 4ØØ TN=VAL(R$):IFTN>255THEN
         43Ø
DK 41Ø IFTY=1ORTY=3ORTY=4ORTY=
         5THENOP=OP+4
ER 42Ø GOTO2ØØØ
SR 43Ø H%=TN/256:L%=TN-256*H%:
         IFTY=2ORTY=7THENOP=OP+8
         :GOTO47Ø
SE 44Ø IFTY=1ORTY=3ORTY=4ORTY=
         5THENOP=OP+12:GOTO47Ø
AJ 45Ø IFTY=6ORTY=9THEN47Ø
SE 46Ø GOTO85Ø
KB 47Ø GOTO3ØØØ
JS 48Ø TN=VAL(RIGHT$(R$,LL-1))
SM 49Ø IFTY=1THENOP=OP+8:GOTO2
         ØØØ
BX 5ØØ IFTY=4ORTY=5THENGOTO2ØØ
         Ø
FG 51Ø GOTO85Ø
XS 52Ø IFRIGHT$(R$,2)=")Y"THEN
         54Ø
DQ 53Ø IFRIGHT$(R$,2)="X)"THEN
         57Ø
AG 54Ø TN=VAL(MID$(R$,2,LL-3))
RF 55Ø IFTY=1THENOP=OP+16:GOTO
         2ØØØ
KM 56Ø GOTO85Ø
QB 57Ø TN=VAL(MID$(R$,2,LL-3))
GG 58Ø IFTY=1THENGOTO2ØØØ
FQ 59Ø GOTO85Ø
BX 6ØØ TN=VAL(R$):TN=TN-SA-2:I
         FTN<-128ORTN>127THENPRI
         NT"TOO FAR ";:GOTO85Ø
PC 61Ø IFTN<ØTHENTN=TN+256
XK 62Ø GOTO2ØØØ
SK 63Ø IFRIGHT$(R$,2)=")Y"THEN
         54Ø
MA 64Ø IFRIGHT$(R$,1)="X"THEN7
         2Ø
JE 65Ø REM *ZERO Y
XB 66Ø TN=VAL(LEFT$(R$,LL-1)):
         IFTN>255THEN68Ø
PG 67Ø IFTY=2ORTY=5THEN73Ø
PA 675 IFTY=1THEN76Ø
PC 68Ø GOSUB77Ø:IFTY=1THENOP=O
         P+24:GOTO71Ø
HF 69Ø IFTY=5THENOP=OP+28:GOTO
         71Ø
JD 7ØØ GOTO85Ø
SX 71Ø GOTO3ØØØ
MQ 72Ø TN=VAL(LEFT$(R$,LL-1)):
         IFTN>255THENGOSUB77Ø:GO
         TO78Ø
HA 73Ø IFTY=2THENOP=OP+16:GOTO
         76Ø
AC 74Ø IFTY=1ORTY=3ORTY=5THENO
         P=OP+2Ø:GOTO76Ø
EH 75Ø GOTO85Ø
BD 76Ø GOTO2ØØØ
SS 77Ø H%=TN/256:L%=TN-256*H%:
         RETURN
MF 78Ø IFTY=2THENOP=OP+24:GOTO
         81Ø
CJ 79Ø IFTY=1ORTY=3ORTY=5THENO
         P=OP+28:GOTO81Ø
RK 8ØØ GOTO85Ø
GE 81Ø GOTO3ØØØ
FK 82Ø TN=VAL(R$)
SK 83Ø GOSUB77Ø
BJ 84Ø GOTO71Ø
QC 85Ø PRINT"{RVS} ERROR ":GOT
         O23Ø
KP 1ØØØ REM 1 BYTE INSTRUCTION
          S
HF 1Ø1Ø POKESA,OP:SA=SA+1:IFH=
          1THEN 1Ø3Ø
HH 1Ø2Ø PRINTOP:GOTO23Ø
EG 1Ø3Ø DE = OP:GOSUB4ØØØ:PRIN
          TH$:GOTO23Ø
EP 2ØØØ REM 2 BYTE INSTRUCTION
          S
MB 2ØØ5 IFTN>255THENPRINT" INC
          ORRECT ARGUMENT.":GOTO
          23Ø
FA 2Ø1Ø POKESA,OP:POKESA+1,TN:
          SA=SA+2:IFH=1THEN2Ø3Ø
FQ 2Ø2Ø PRINTOP;TN:GOTO23Ø
GR 2Ø3Ø DE = OP:GOSUB4ØØØ:PRIN
          TH$" ";
EG 2Ø4Ø DE = TN:GOSUB4ØØØ:PRIN
          TH$:GOTO23Ø
DM 3ØØØ REM 3 BYTE INSTRUCTION
          S
AX 3Ø1Ø POKESA,OP:POKESA+1,L%:
          POKESA+2,H%:SA=SA+3:IF
          H=1THEN3Ø3Ø
SA 3Ø2Ø PRINTOP;L%;H%:GOTO23Ø
DB 3Ø3Ø DE = OP:GOSUB4ØØØ:PRIN
          TH$" ";
MH 3Ø4Ø DE = L%:GOSUB4ØØØ:PRIN
          TH$" ";
KF 3Ø5Ø DE = H%:GOSUB4ØØØ:PRIN
          TH$:GOTO23Ø
HM 4ØØØ REM{2 SPACES}DECIMAL T
          O HEX (DE TO H$)
BM 4Ø1Ø H$="":FORM=SZTOØSTEP-1
          :N%=DE/(16↑M):DE=DE-N%
          *16↑M:H$=H$+MID$(HE$,N
          %+1,1)
DA 4Ø2Ø NEXT:SZ=1:RETURN
CD 5ØØØ REM{2 SPACES}HEX TO DE
          CIMAL (H$ TO DE)
```

```
DD 5010 D=0:Q=3:FORM=1TO4:FORW
        =0TO15:IFMID$(H$,M,1)=
        MID$(HE$,W+1,1)THEN503
        0
CS 5020 NEXTW
AR 5030 D1=W*(16↑(Q)):D=D+D1:Q
        =Q-1:NEXTM:DE=INT(D):R
        ETURN
JP 6000 REM ACCEPT HEX OPCODE
        {SPACE}INPUT AND TRANS
        LATE IT TO DECIMAL
GX 6010 IFLEFT$(R$,1)="#"THENH
        $="00"+RIGHT$(R$,2):GO
        SUB5000:R$="#"+STR$(DE
        ):RETURN
JB 6020 LS=LEN(R$):AZ$=LEFT$(R
        $,1):ZA$=MID$(R$,LS,1)
        :IFAZ$<>"("THEN6050
QX 6030 IFZA$="Y"THENH$="00"+M
        ID$(R$,2,2):GOSUB5000:
        R$="("+STR$(DE)+")Y":R
        ETURN
MB 6040 IFZA$=")"THENH$="00"+M
        ID$(R$,2,2):GOSUB5000:
        R$="("+STR$(DE)+"X)":R
        ETURN
KF 6050 IFZA$="X"ORZA$="Y"THEN
        6070
KS 6060 H$=LEFT$(ZO$,4-LS)+R$:
        GOSUB5000:R$=STR$(DE):
        RETURN
KF 6070 IFLS=5THENH$=LEFT$(R$,
        4):GOTO6090
AC 6080 H$="00"+LEFT$(R$,2)
GE 6090 GOSUB5000:R$=STR$(DE)+
        ZA$:RETURN
FR 20000 DATAADC1097,AND1033,A
         SL3002,BCC8144,BCS817
         6,BEQ8240,BIT7036,BMI
         8048
BK 20010 DATABNE8208,BPL8016,B
         RK0000,BVC8080,BVS811
         2,CLC0024,CLD0216,CLI
         0088
ES 20020 DATACLV0184,CMP1193,C
         PX4224,CPY4192,DEC219
         8,DEX0202,DEY0136,EOR
         1065
KR 20030 DATAINC2230,INX0232,I
         NY0200,JMP6076,JSR903
         2,LDA1161,LDX5162,LDY
         5160
DP 20040 DATALSR3066,NOP0234,O
         RA1001,PHA0072,PHP000
         8,PLA0104,PLP0040,ROL
         3034
RD 20050 DATAROR3098,RTI0064,R
         TS0096,SBC1225,SEC005
         6,SED0248,SEI0120,STA
         1129
DR 20060 DATASTX2134,STY2132,T
         AX0170,TAY0168,TSX018
         6,TXA0138,TXS0154,TYA
         0152
```

Power BASIC: Line Count

(Article on page 80.)

Program 1: Line Count—64 Version

```
MS 10 FORA=49152TO49539:READB:
      POKEA,B:C=C+B:NEXT
PH 20 IFC<>51719THENPRINT"
      {CLR}DATA ERROR":STOP
MJ 30 PRINT "{CLR}{2 DOWN}
      {WHT}SYS 49155: TO INITI
      ALIZE":PRINT "{DOWN}SYS
      {SPACE}49152: FOR A REPO
      RT"
MM 40 DATA 76,13,193,32,41,192
      ,162,14,160,0,169,255,14
      5,251,136,208,251,230,25
      2
MS 50 DATA 202,208,246,169,50,
      141,8,3,169,192,141,9,3,
      169,255,141,129,193,141
JC 60 DATA 130,193,96,169,0,13
      3,251,169,194,133,252,96
      ,32,41,192,165,58,201,25
      5
PP 70 DATA 240,98,205,130,193,
      208,7,165,57,205,129,193
      ,240,86,160,1,185,57,0
MB 80 DATA 153,129,193,136,16,
      247,160,0,177,251,200,49
      ,251,201,255,208,17,165,
      58
KF 90 DATA 145,251,136,165,57,
      145,251,32,160,192,32,16
      0,192,208,47,160,0,177,2
      51
GP 100 DATA 217,57,0,208,32,20
       0,192,2,208,244,32,160,
       192,208,28,32,88,193,16
       9
JQ 110 DATA 99,160,193,32,30,1
       71,166,57,165,58,32,205
       ,189,162,128,108,0,3,32
EK 120 DATA 64,193,76,82,192,7
       6,228,167,160,2,177,251
       ,24,105,1,145,251,208,7
XH 130 DATA 200,177,251,105,0,
       145,251,96,169,0,141,13
       1,193,32,41,192,160,0,1
       77
CP 140 DATA 251,200,49,251,201
       ,255,240,65,160,1,177,2
       51,153,129,193,136,16,2
       48,160
HD 150 DATA 4,177,251,56,237,1
       29,193,200,177,251,237,
       130,193,176,33,169,1,14
       1,131
AF 160 DATA 193,160,7,177,251,
       72,136,16,250,160,4,104
       ,145,251,200,192,8,208,
       248
KA 170 DATA 160,0,104,145,251,
       200,192,4,208,248,32,64
       ,193,76,187,192,173,131
       ,193
RA 180 DATA 208,167,96,32,179,
       192,32,41,192,169,13,32
       ,210,255,160,0,177,251,
       170
JA 190 DATA 200,49,251,201,255
       ,240,25,177,251,32,78,1
       93,169,58,32,210,255,20
       0,177
FG 200 DATA 251,170,200,177,25
       1,32,78,193,32,64,193,2
       08,214,76,88,193,165,25
       1,24
AX 210 DATA 105,4,133,251,165,
       252,105,0,133,252,96,14
       0,131,193,32,205,189,17
       2,131
BB 220 DATA 193,96,169,228,141
       ,8,3,169,167,141,9,3,96
       ,13,84,79,79,32,77
EM 230 DATA 65,78,89,32,69,88,
       69,67,85,84,73,79,78,83
       ,32,79,70,32,76
AP 240 DATA 73,78,69,32,0,0,0,
       0
```

Program 2: Line Count—128 Version

```
KJ 10 FORA=3072TO3517:READB:PO
      KEA,B:C=C+B:NEXT
BH 20 IFC<>53196THENPRINT"
      {CLR}DATA ERROR":STOP
CS 30 PRINT "{CLR}{WHT}
      {2 DOWN}SYS 3075: TO INI
      TIALIZE":PRINT "{DOWN}SY
      S 3072: FOR A REPORT"
FG 40 DATA 76,63,13,169,48,141
      ,0,255,32,56,12,162,14,1
      60,0,169,255,145,251
RS 50 DATA 136,208,251,230,252
      ,202,208,246,169,65,141,
      8,3,169,12,141,9,3,169
FG 60 DATA 255,141,187,13,141,
      188,13,169,240,141,19,18
      ,169,0,141,0,255,96,169
SP 70 DATA 0,133,251,169,240,1
      33,252,96,169,48,141,0,2
      55,32,56,12,165,60,201
CQ 80 DATA 255,240,10,205,188,
      13,208,7,165,59,205,187,
      13,240,116,160,1,185,59
PK 90 DATA 0,153,187,13,136,16
      ,247,160,0,177,251,200,4
      9,251,201,255,208,17,165
QR 100 DATA 60,145,251,136,165
       ,59,145,251,32,210,12,3
       2,210,12,208,77,160,0,1
       77
QD 110 DATA 251,217,59,0,208,6
       2,200,192,2,208,244,32,
       210,12,208,58,32,155,13
GB 120 DATA 169,159,160,12,32,
       125,255,13,84,79,79,32,
       77,65,78,89,32,69,88
ES 130 DATA 69,67,85,84,73,79,
       78,83,32,79,70,32,76,73
       ,78,69,32,0,166
JX 140 DATA 59,165,60,32,133,1
       3,162,128,108,0,3,32,11
       9,13,76,102,12,76,162
SK 150 DATA 74,160,2,177,251,2
       4,105,1,145,251,208,7,2
       00,177,251,105,0,145,25
       1
XP 160 DATA 96,169,0,141,189,1
       3,32,56,12,160,0,177,25
       1,200,49,251,201,255,24
       0
SX 170 DATA 65,160,1,177,251,1
       53,187,13,136,16,248,16
       0,4,177,251,56,237,187,
       13
FK 180 DATA 200,177,251,237,18
       8,13,176,33,169,1,141,1
       89,13,160,7,177,251,72,
       136
RA 190 DATA 16,250,160,4,104,1
       45,251,200,192,8,208,24
       8,160,0,104,145,251,200
       ,192
RF 200 DATA 4,208,248,32,119,1
       3,76,237,12,173,189,13,
       208,167,96,169,48,141,0
SF 210 DATA 255,32,229,12,32,5
       6,12,169,13,32,171,13,1
       60,0,177,251,170,200,49
PF 220 DATA 251,201,255,240,25
       ,177,251,32,133,13,169,
       58,32,171,13,200,177,25
       1,170
PQ 230 DATA 200,177,251,32,133
       ,13,32,119,13,208,214,7
       6,155,13,165,251,24,105
       ,4
CB 240 DATA 133,251,165,252,10
       5,0,133,252,96,140,189,
       13,72,169,0,141,0,255,1
       04
BD 250 DATA 32,50,142,169,48,1
       41,0,255,172,189,13,96,
       169,162,141,8,3,169,74
CE 260 DATA 141,9,3,169,0,141,
       0,255,96,72,169,0,141,0
       ,255,104,32,210,255
```

```
FB 270 DATA 169,48,141,Ø,255,9
       6,Ø,Ø,Ø
```

Program 3: Line Count—Plus/4 and 16 Version

```
DQ 1Ø FORA=4Ø96TO4479:READB:PO
      KEA,B:C=C+B:NEXT
FG 2Ø IFC<>45377THENPRINT"
      {CLR}DATA ERROR":STOP
GP 3Ø PRINT "{CLR}{2 DOWN}SYS
      {SPACE}4Ø99: TO INITIALI
      ZE":PRINT "{DOWN}SYS 4Ø9
      6: FOR A REPORT"
SD 4Ø DATA 76,39,17,32,41,16,1
      62,14,16Ø,Ø,169,255,145,
      252,136,2Ø8,251,23Ø,253
HS 5Ø DATA 2Ø2,2Ø8,246,169,5Ø,
      141,8,3,169,16,141,9,3,1
      69,255,141,125,17,141
JC 6Ø DATA 126,17,96,169,Ø,133
      ,252,169,18,133,253,96,3
      2,41,16,165,58,2Ø1,255
GS 7Ø DATA 24Ø,124,2Ø5,126,17,
      2Ø8,7,165,57,2Ø5,125,17,
      24Ø,112,16Ø,1,185,57,Ø
GP 8Ø DATA 153,125,17,136,16,2
      47,16Ø,Ø,177,252,2ØØ,49,
      252,2Ø1,255,2Ø8,17,165,5
      8
MG 9Ø DATA 145,252,136,165,57,
      145,252,32,186,16,32,186
      ,16,2Ø8,73,16Ø,Ø,177,252
QD 1ØØ DATA 217,57,Ø,2Ø8,58,2Ø
       Ø,192,2,2Ø8,244,32,186,
       16,2Ø8,54,32,114,17,32
AK 11Ø DATA 79,255,13,84,79,79
       ,32,77,65,78,89,32,69,8
       8,69,67,85,84,73
QG 12Ø DATA 79,78,83,32,79,7Ø,
       32,76,73,78,69,32,Ø,166
       ,57,165,58,32,1Ø4
JD 13Ø DATA 17,162,128,1Ø8,Ø,3
       ,32,9Ø,17,76,82,16,76,2
       14,139,16Ø,2,177,252
EP 14Ø DATA 24,1Ø5,1,145,252,2
       Ø8,7,2ØØ,177,252,1Ø5,Ø,
       145,252,96,169,Ø,141,12
       7
DG 15Ø DATA 17,32,41,16,16Ø,Ø,
       177,252,2ØØ,49,252,2Ø1,
       255,24Ø,65,16Ø,1,177,25
       2
MQ 16Ø DATA 153,125,17,136,16,
       248,16Ø,4,177,252,56,23
       7,125,17,2ØØ,177,252,23
       7,126
EH 17Ø DATA 17,176,33,169,1,14
       1,127,17,16Ø,7,177,252,
       72,136,16,25Ø,16Ø,4,1Ø4
SD 18Ø DATA 145,252,2ØØ,192,8,
       2Ø8,248,16Ø,Ø,1Ø4,145,2
       52,2ØØ,192,4,2Ø8,248,32
       ,9Ø
EE 19Ø DATA 17,76,213,16,173,1
       27,17,2Ø8,167,96,32,2Ø5
       ,16,32,41,16,169,13,32
SG 2ØØ DATA 21Ø,255,16Ø,Ø,177,
       252,17Ø,2ØØ,49,252,2Ø1,
       255,24Ø,25,177,252,32,1
       Ø4,17
EE 21Ø DATA 169,58,32,21Ø,255,
       2ØØ,177,252,17Ø,2ØØ,177
       ,252,32,1Ø4,17,32,9Ø,17
       ,2Ø8
QM 22Ø DATA 214,76,114,17,165,
       252,24,1Ø5,4,133,252,16
       5,253,1Ø5,Ø,133,253,96,
       14Ø
QK 23Ø DATA 127,17,32,95,164,1
       72,127,17,96,169,214,14
       1,8,3,169,139,141,9,3
FE 24Ø DATA 96,Ø,Ø,Ø
```

ESCaping With 128

(Article on page 70.)

Program 1: ESC Sequence Demo

```
GF 1ØØ PRINTCHR$(14);
PA 11Ø MD=RGR(X):IF MD<>5 THEN
        PRINT"{CLR}{DOWN}{RVS}
       {2 SPACES}PLEASE USE 8Ø
        COLUMNS FOR THIS DEMO.
       {2 SPACES}{OFF}":END
AF 12Ø REM *******************
       **************
GP 13Ø REM * PRINT HEADER AND
       {SPACE}MAKE WINDOW *
JD 14Ø REM *******************
       **************
EJ 15Ø PRINT"{HOME}{RVS}
       {13 SPACES}USING ESCAPE
        CODES IN A BASIC PROGR
       AM - EXAMPLES
       {34 SPACES}":WINDOWØ,1,
       79,24,1
XF 16Ø PRINT"{CLR}";
MF 17Ø REM *******************
       *************
GS 18Ø REM * ESC D - DELETE LI
       NE EXAMPLE *
CD 19Ø REM *******************
       *************
FD 2ØØ PRINT"{3 DOWN}{3 RIGHT}
       THIS LINE WON'T BE ERAS
       ED..."
KD 21Ø PRINT"{3 RIGHT}THIS LIN
       E WILL BE ERASED USING
       {SPACE}ESC D .....{RVS}
       HIT ANY KEY{OFF}"
QE 22Ø PRINT"{3 RIGHT}THIS LIN
       E WON'T BE ERASED...BUT
        WILL MOVE UP ONE LINE.
       .";
CC 23Ø GETKEYA$:PRINT"{UP}"CHR
       $(27)"D"
PK 24Ø REM *******************
       ***********************
KX 25Ø REM * ESC Q - DELETE TO
        END-OF-LINE EXAMPLE *
DJ 26Ø REM *******************
       ***********************
MG 27Ø PRINT"{2 DOWN}{3 RIGHT}
       THIS LINE WILL BE HALF
       {SPACE}ERASED USING ESC
        Q...HIT ANY KEY
       {23 LEFT}";
MD 28Ø GETKEYA$:PRINTCHR$(27)"
       Q"
BJ 29Ø REM *******************
       ************************
       *
AQ 3ØØ REM * ESC P - DELETE TO
        START-OF-LINE EXAMPLE
       {SPACE}*
SK 31Ø REM *******************
       ************************
       *
EP 32Ø PRINT"{3 RIGHT}THIS LIN
       E WILL BE HALF ERASED U
       SING ESC P...HIT ANY KE
       Y{24 LEFT}";
JA 33Ø GETKEYA$:PRINTCHR$(27)"
       P"
HX 34Ø REM *******************
       ***************
DA 35Ø REM * ESC I - INSERT A
       {SPACE}LINE EXAMPLE *
XA 36Ø REM *******************
       ***************
KM 37Ø PRINT"{3 DOWN}{RVS}YOU
       {SPACE}COULD DEFINE THE
        FUNCTION KEYS AS WORD-
       PROCESSING TOOLS"
MF 38Ø KEY 1,CHR$(27)+"I" :PRI
       NT"{RVS}WITH: KEY 1, CH
       R$(27)+'I'{OFF}"
XF 39Ø PRINT"{DOWN}{3 RIGHT}TH
       IS LINE WILL STAY HERE.
       .."
RR 4ØØ PRINT"{3 RIGHT}THIS LIN
       E WILL MOVE DOWN ONE US
       ING ESC I...{RVS} HIT F
       1 {OFF}";
XD 41Ø GETKEYA$:GETKEYB$:C$=A$
       +B$:PRINTC$;
CX 42Ø PRINT"{RVS}...AND I WIL
       L INSERT THIS LINE...":
       PRINT"{2 DOWN}";
KF 43Ø REM *******************
       ***********************
AH 44Ø REM * ESC A - AUTOMATIC
        INSERT MODE EXAMPLE *
BE 45Ø REM *******************
       ***********************
QK 46Ø PRINT"{DOWN}WE'LL INSER
       T TEXT INTO THE MIDDLE
       {SPACE}OF THIS LINE USI
       NG ESC A ...HIT ANY KEY
       ..";
DX 47Ø GETKEY A$
RA 48Ø PRINT"{22 LEFT}";CHR$(2
       7)"A";"{RVS}THIS IS INS
       ERT MODE "
SK 49Ø REM *******************
       ************************
       **
HQ 5ØØ REM * REMEMBER TO CANCE
       L INSERT MODE FOR SPEED
        *
FJ 51Ø REM *******************
       ************************
       **
EM 52Ø PRINTCHR$(27)"C";
XS 53Ø REM *******************
       ************************
GS 54Ø REM * ESC @ - ERASE TO
       {SPACE}END-OF-SCREEN EX
       AMPLE *
HJ 55Ø REM *******************
       ************************
PH 56Ø PRINT"{2 DOWN}NOW WE'LL
        USE ESC @ TO ERASE HAL
       F OF THE SCREEN...HIT A
       NY KEY";
FF 57Ø GETKEYA$
BQ 58Ø PRINT"{HOME}{14 DOWN}";
       CHR$(27)"@";
BP 59Ø REM *******************
       ************************
       *
XC 6ØØ REM * ESC T & ESC B - S
       ET UP A WINDOW EXAMPLE
       {SPACE}*
QE 61Ø REM *******************
       ************************
       *
QE 62Ø PRINTCHR$(27)"T";"
       {3Ø RIGHT}{15 DOWN}";CH
       R$(27)"B"
XM 63Ø PRINT"{CLR}THIS IS A WI
       NDOW":PRINT"USING ESC T
        AND ESC B":SLEEP 3:CAT
       ALOG
HF 64Ø PRINT"{DOWN}YOU CLEAR T
       HE WINDOW":PRINT"BY PRI
       NTING 'HOME'":PRINT"TWI
       CE."
FB 65Ø PRINT"{DOWN}HIT ANY KEY
        TO END":GETKEYA$
KD 66Ø PRINT"{2 HOME}";:PRINT"
       {CLR}{3 DOWN}{5 RIGHT}
       {RVS}ALL DONE !!"
QA 67Ø END
RA 68Ø PRINT"HIT ANY KEY..":GE
       TKEYA$:RETURN
```

Program 2: ESC Sequence Function Keys

```
MD 90 REM DEFINE KEYS:
BX 100 REM
MX 110 KEY 1, CHR$(27)+"A" : R
       EM INSERT MODE
JD 120 KEY 2, CHR$(27)+"C" : R
       EM CANCEL INSERT MODE
ED 130 KEY 3, CHR$(27)+"I" : R
       EM INSERT A LINE
JD 140 KEY 4, CHR$(27)+"D" :RE
       M DELETE A LINE
PD 150 KEY 5, CHR$(27)+"J" :RE
       M MOVE TO START OF LINE
BQ 160 KEY 6, CHR$(27)+"K" :RE
       M MOVE TO END OF LINE
RX 170 KEY 7, CHR$(27)+"P" :RE
       M ERASE TO START OF LIN
       E
GF 180 KEY 8, CHR$(27)+"Q" :RE
       M ERASE TO END OF LINE
GG 190 REM
AM 200 REM{2 SPACES}SET UP MEN
       U AT BOTTOM OF SCREEN
DG 210 REM
XQ 220 PRINT"{HOME}{23 DOWN}";
       CHR$(27)"M";
MR 230 PRINT"{RVS}{3 SPACES}F1
       -INSERT{5 SPACES}F3-INS
       ERT LINE{5 SPACES}F5-MO
       VE TO START OF LINE
       {3 SPACES}F7-ERASE TO S
       TART"
JX 240 PRINT"{RVS}{3 SPACES}F2
       -NO INSERT{2 SPACES}F4-
       DELETE LINE{5 SPACES}F6
       -MOVE TO END OF LINE
       {5 SPACES}F8-ERASE TO E
       ND{2 SPACES}"
DK 250 REM
HG 260 REM PROTECT THE MENU WI
       TH A WINDOW
PM 270 REM
QP 280 WINDOW Ø,Ø,79,22,1
AR 290 PRINT"{CLR}";CHR$(27)"L
       ";
```

Directory Extension

(Article on page 75.)

```
AB 100 POKE 5328Ø,15:POKE 5328
       1,15:PRINT"{CLR}{WHT}
       {6 DOWN}{1Ø SPACES}DIRE
       CTORY EXTENSION"
XS 110 PRINT"{1Ø SPACES}⁅19 T⁆
       {5 DOWN}"
KR 120 FOR I=83Ø TO 9Ø2:READ D
       :POKE I,D:NEXT
KC 130 DN$="{HOME}{21 DOWN}":P
       $="⁅P⁆{DOWN}{2 LEFT}⁅N⁆
       {RVS} {OFF}⁅H⁆{DOWN}
       {2 LEFT}⁅Y⁆":XX=1
HE 140 MT$="{RVS}{WHT}
       {19 SPACES}{OFF}
       {4 SPACES}{RVS}
       {36 SPACES}{BLK}"
PD 150 DM=144:DIM F$(DM),S$(DM
       ),T$(DM),S2$(DM)
RD 160 FOR I=Ø TO DM:F$(I)=CHR
       $(32)+"{18 SPACES}":NEX
       T
BS 170 PRINT"{UP}{11 SPACES}RE
       ADING DIRECTORY
       {2 SPACES}":OPEN1,8,Ø,"
       $":SYS 83Ø:CLOSE 1
FB 180 GOSUB 95Ø
QR 190 POKE 198,Ø:PRINT"{CLR}
       {DOWN}{WHT}"SPC(9)P$"
       {UP}{LEFT}{RVS}1{OFF}"S
       PC(3)"VIEW DIRECTORY FI
       LE"
CE 200 PRINT"{2 DOWN}"SPC(9)P$
       "{UP}{LEFT}{RVS}2{OFF}"
       SPC(3)"SEND DISK COMMAN
       DS"
DH 210 PRINT"{2 DOWN}"SPC(9)P$
       "{UP}{LEFT}{RVS}3{OFF}"
       SPC(3)"SAVE"
HK 220 PRINT"{2 DOWN}"SPC(9)P$
       "{UP}{LEFT}{RVS}4{OFF}"
       SPC(3)"VIEW DIFFERENT D
       ISK"
PB 230 PRINT"{2 DOWN}"SPC(9)P$
       "{UP}{LEFT}{RVS}5{OFF}"
       SPC(3)"QUIT"
JB 240 PRINT"{2 DOWN}"SPC(28)P
       $"{UP}{LEFT}{RVS} {OFF}
       ":PRINT"{UP}"SPC(9)"ENT
       ER YOUR CHOICE ⁅N⁆⁅B⁆
       {LEFT}";
GQ 250 POKE 2Ø4,Ø:GET CG$:IF C
       G$<"1" OR CG$>"5" THEN
       {SPACE}25Ø
BR 260 POKE 2Ø4,1:POKE 2Ø7,Ø:P
       RINT"{RVS}"CG$:CG=VAL(C
       G$)
EF 270 ON CG GOTO 29Ø,1Ø9Ø,52Ø
       ,124Ø,134Ø
ER 280 REM *** VIEW FILES ***
XP 290 C=PEEK(Ø)-1:A=1
EJ 300 IF A=> C THEN 33Ø
AH 310 IF A<Ø AND C/6=INT(C/6)
        THEN A=C-5:GOTO 33Ø
SB 320 IF A<Ø THEN DN=INT(C/6)
       :A=DN*6+1
HQ 330 PRINT"{CLR}{DOWN}{BLK}
       {OFF}{3 SPACES}DISK TIT
       LE: {RVS}"F$(Ø)" ":PRIN
       T"{DOWN}":PQ=5
HG 340 FOR I=A TO A+PQ
CX 350 IF F$(I)=CHR$(32)+"
       {18 SPACES}" OR F$(I)="
       " THEN F$(I)="":PQ=PQ-1
RJ 360 NEXT
PE 370 FOR I= A TO A+PQ
DE 380 IF I>9 THEN W$=" {LEFT}
       {RVS}":GOTO 4ØØ
GR 390 W$="{2 SPACES}{LEFT}
       {RVS}"
FE 400 PRINT"{LEFT}"I;W$;LEFT$
       (F$(I),16)":"MT$:PRINT"
       {2 UP}"SPC(2Ø);
FA 410 PRINT"{RVS}{WHT}"S$(I)"
       {2 DOWN}{BLK}":NEXT
AJ 420 PRINT DN$"{3 DOWN} F1-M
       ENU F3-ADD F5-FWD F7-BA
       CK ↑-DELETE{HOME}":POKE
        198,Ø:WAIT198,1
AA 430 IF PEEK(197)=54 THEN 12
       8Ø
SH 440 IF PEEK(197)=4 THEN 19Ø
BF 450 IF PEEK(197)=5 THEN 61Ø
MS 460 IF PEEK(197)=6 AND A=C
       {SPACE}THEN A=1:GOTO 3Ø
       Ø
RM 470 IF PEEK(197)=6 AND A+6>
       C THEN A=1:GOTO 3ØØ
JA 480 IF PEEK(197)=6 AND A<C
       {SPACE}THEN A=A+6:GOTO
       {SPACE}3ØØ
FP 490 IF PEEK(197)=3 THEN A=A
       -6:GOTO 3ØØ
DC 500 GOTO 43Ø
XJ 510 REM *** SAVE DESCRIPTIO
       NS ***
RA 520 PRINT"{2 UP}"SPC(28)"
       {LEFT}{2 DOWN} ":PRINT"
       {2 UP}"SPC(9)"
       {2 SPACES}SAVING "CHR$(
       34)"DIR FILE";
KB 530 PRINT CHR$(34)"...":FOR
        R=1 TO C: F S$(R)="" T
       HEN S$(R)="{RVS} "
MQ 540 NEXT R
XG 550 CLOSE1:CLOSE 15:OPEN 1,
       8,15,"SØ:DIR FILE":CLOS
       E 1:OPEN 15,8,15,"I"
CH 560 OPEN 1,8,8,"Ø:DIR FILE,
       S,W":GOSUB 1Ø5Ø
JX 570 PRINT#1,C
JA 580 FOR I=1 TO C:PRINT#1,LE
       FT$(F$(I),16):PRINT#1,S
       $(I):NEXT:CLOSE 1
CD 590 PRINT"{UP}{31 SPACES}":
       GOTO 19Ø
JD 600 REM *** ADD STATEMENTS
       {SPACE}***
HX 610 PRINTDN$"{3 DOWN}"SPC(9
       )"{RVS}F3-ADD{HOME}"
XA 620 PRINT DN$"{DOWN}
       {3 SPACES}ENTER FILE #"
       ;:POKE 198,Ø:INPUT FI$:
       FI=VAL(FI$)
CX 630 IF FI$="Ø" THEN PRINTDN
       $"{DOWN}{18 SPACES}":GO
       TO 42Ø
KA 640 IF FI<A OR FI>A+PQ THEN
        PRINT DN$ SPC(17)"
       {DOWN}{18 SPACES}":GOTO
        61Ø
XM 650 PRINT DN$"{DOWN}
       {26 SPACES}":IF FI<7 TH
       EN Z=FI:GOTO 68Ø
GA 660 Z=INT(FI/6)+1:DD=Z*6-FI
       :Z=6-DD
DD 670 IF Z=Ø THEN Z=6
EX 680 PRINT"{HOME}{DOWN}":FOR
       I=1 TO Z:PRINT"{DOWN}":
       NEXT
HE 690 F$(FI)=LEFT$(F$(FI),16)
SC 700 I$=""
GQ 710 PRINT"{UP}"SPC(19)"
       {RVS}>{WHT}"SPC(LEN(I$)
       );:B$="<":CC=LEN(B$):L=
       1
PQ 720 PRINT"{RVS}"MID$(B$,L,2
       );"{LEFT}";:L=-L*(L<CC)
       +1
FG 730 GET A$:IF A$="" OR A$=C
       HR$(34) OR A$=CHR$(44)
       {SPACE}OR A$=CHR$(58) T
       HEN 73Ø
GA 740 IF A$=CHR$(59) THEN 73Ø
GQ 750 IF A$=CHR$(13) AND LEN(
       I$)=18 THEN 91Ø
PF 760 IF LEN(I$)=18 AND A$<>C
       HR$(2Ø) THEN I$=I$+A$:P
       RINT A$"{OFF}{4 SPACES}
       {RVS}<{LEFT}";:GOTO 72Ø
BJ 770 IF LEN(I$)=55 THEN PP=1
JC 780 IF LEN(I$)=55 AND A$<>C
       HR$(2Ø) AND A$<>CHR$(13
       ) THEN 73Ø
QF 790 IF A$=CHR$(13) AND LEN(
       I$)=Ø THEN 33Ø
EM 800 IF A$=CHR$(13) THEN 91Ø
XX 810 IF A$=CHR$(2Ø) AND LEN(
       I$)=19 THEN 88Ø
HF 820 IF A$=CHR$(2Ø) AND LEN(
       I$)>Ø THEN 85Ø
EQ 830 IF ASC(A$)<32 OR ASC(A$
       )>95 THEN 72Ø
AK 840 I$=I$+A$:PRINT A$;:GOTO
        72Ø
EE 850 I$=LEFT$(I$,LEN(I$)-1)
FQ 860 IF LEN(I$)=54 THEN PRIN
       T"{OFF} {RVS}{2 LEFT}";
       :GOTO 72Ø
PA 870 PRINT"{RVS} {2 LEFT}";:
       GOTO 72Ø
HE 880 I$=LEFT$(I$,LEN(I$)-1):
       PRINT"{RVS} ":D=19
DQ 890 IF PP=1 THEN PRINT"
```