**Advice To PET Owners: How To Be A VIC Expert**

The 6502 Resource Magazine
PET • Apple • Atari • OSI • KIM • SYM • AIM

**Unleash The Power of Your Atari CPU**

# COMPUTE!

$2.50
April, 1981
Issue 11
Vol. 3, No. 4
63379

## The Journal For Progressive Computing™

**Product Preview: Commodore's $299 VIC-20 Computer**

**Super Cube: Artistry On Your Atari**

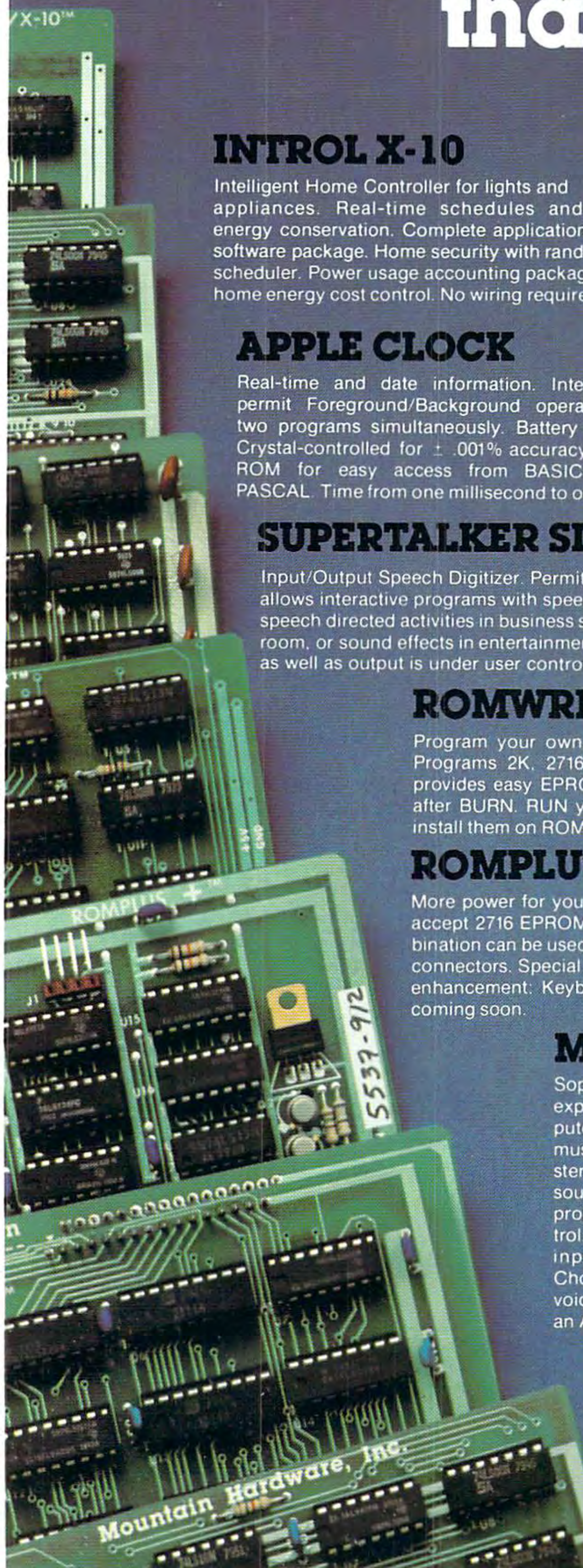
**Basically Useful BASIC: An Ascending-Descending Sort**

**Resolving Applesoft And Hires Graphics Memory Conflicts**


04
74470 63379

# Mountain Computer makes more peripherals for the Apple Computer than Anybody.

and . . . a place to put them




## INTROL X-10

Inteligent Home Controller for lights and appliances. Real-time schedules and energy conservation. Complete applications software package. Home security with random scheduler. Power usage accounting package for home energy cost control. No wiring required.

## APPLE CLOCK

Real-time and date information. Interrupts permit Foreground/Background operation of two programs simultaneously. Battery back-up. Crystal-controlled for ± .001% accuracy. Onboard ROM for easy access from BASICs. Supports PASCAL. Time from one millisecond to one year.

## SUPERTALKER SD200

Input/Output Speech Digitizer. Permits talking programs. I/O capability allows interactive programs with speech-prompted inputs. Use output for speech directed activities in business systems, announcements in a control-room, or sound effects in entertainment programs. Easy to use because input as well as output is under user control with special software operating system.

## ROMWRITER

Program your own EPROMs. Create your own firmware. Programs 2K, 2716 5V EPROMs. Disk software package provides easy EPROM programming. EPROMs are verified after BURN. RUN your programs from on-board socket or install them on ROMPLUS+.

## ROMPLUS+

More power for your system through firmware. Six sockets accept 2716 EPROMs or ROM equivalents. Six or any combination can be used at once. Scratch-pad RAM and two TTL connectors. Special 2K ROMs available for powerful system enhancement: Keyboard Filter ROM—COPYROM—Others coming soon.

## MusicSystem

Sophistication previously available only on experimental mini and mainframe computer synthesizers. Digital instrumental music synthesizer system. 16 voices in stereo. Instrument definitions simulate the sound of real instruments—and more. Fully programmable waveforms. Envelope Control. Composition system—sheet music input using standard music notation. Chords and multi-part scoring up to 16 voices. A true instrument that anyone with an Apple can play.

## A/D+D/A

16 channels analog to digital input. 16 channels digital to analog output. Eight bit resolution. Super-fast 9$\mu$ sec. conversion time. Monitor and output to the real world. All on one card.


## EXPANSION CHASSIS

By popular demand! Eight more slots for your Apple. Attractive sturdy enclosure. Its own heavy duty power supply. Easy to use. Address cards in Expansion Chassis the same way as in your Apple. Only one additional command to specify in Apple or in Expansion Chassis. Compatible with all Apple peripherals.

## CardReader

At last! An intelligent, high-quality device for data entry from user-marked cards. Implement BASIC programming, examination scoring, inventory maintenance and other applications requiring off-line data preparation for batch entry later. Connects to any computer having RS-232 interface. Software and cards are available for jobs in business, science and education.

MOUNTAIN COMPUTER has the most comprehensive line of Apple peripherals available. Anywere. From anybody. We know the Apple inside and out and are committed to providing the most innovative and unique products to expand and enhance its capabilities and use. After all, we were the first company to make an Apple peripheral—except Apple Computer.

Available at Apple Dealers worldwide.

Mountain Computer INCORPORATED
300 Harvey West Blvd., Santa Cruz, CA 95060
(408) 429-8600 TWX 910 598-4504

*Based on Manufacturer's published catalogs—Apr. 1980


# Turn your Apple into the world's most versatile personal computer.

**The SoftCard™ Solution.** SoftCard turns your Apple into two computers. A Z-80 and a 6502. By adding a Z-80 microprocessor and CP/M to your Apple, SoftCard turns your Apple into a CP/M based machine. That means you can access the single largest body of microcomputer software in existence. Two computers in one. And, the advantages of both.

**Plug and go.** The SoftCard system starts with a Z-80 based circuit card. Just plug it into any slot (except 0) of your Apple. No modifications required. SoftCard supports most of your Apple peripherals, and, in 6502-mode, your Apple is still your Apple.

**CP/M for your Apple.** You get CP/M on disk with the SoftCard package. It's a powerful and simple-to-use operating system. It supports more software than any other microcomputer operating system. And that's the key to the versatility of the SoftCard/Apple.

CIRCLE READER SERVICE NUMBER

**BASIC included.** A powerful tool, BASIC-80 is included in the SoftCard package. Running under CP/M, ANSI Standard BASIC-80 is the most powerful microcomputer BASIC available. It includes extensive disk I/O statements, error trapping, integer variables, 16-digit precision, extensive EDIT commands and string functions, high and low-res Apple graphics, PRINT USING, CHAIN and COMMON, plus many additional commands. And, it's a BASIC you can compile with Microsoft's BASIC Compiler.

**More languages.** With SoftCard and CP/M, you can add Microsoft's ANSI Standard COBOL, and FORTRAN, or Basic Compiler and Assembly Language Development System. All, more powerful tools for your Apple.

**Seeing is believing.** See the SoftCard in operation at your Microsoft or Apple dealer. We think you'll agree that the SoftCard turns your Apple into the world's most versatile personal computer.

**Complete information?** It's at your dealer's now. Or, we'll send it to you and include a dealer list. Write us. Call us. Or, circle the reader service card number below.



MICROSOFT
CONSUMER PRODUCTS

Microsoft Consumer Products, 400 108th Ave. N.E., Bellevue, WA 98004. (206) 454-1315

## Table of Contents

**April, 1981, Vol. 3, No. 4**


The Editor's Notes ........ Robert Lock, 4
An Interview with Dr. Chip ........ Robert Lock, 8
Computers and Society ........ David D. Thornburg, 10
The Beginner's Page ........ Robert Lock, 14
An Applications Commentary: Stimulating Simulations ........ Gregory R. Glau, 18
The Commodore VIC-20: A First Look ........ David D. Thornburg, 26
Advice to PET Owners: How to be a VIC Expert ........ Jim Butterfield, 34
Basically Useful BASIC: Ascending/Descending Sort ........ Rick Keck, 36
Program Compactor ........ Edward H. Carlson, 40
The Mysterious and Unpredictable RND: Part 4,
Create an Adventure ........ Bob Albrecht and George Firedrake, 46
Micros with the Handicapped ........ Susan Semancik, 52
Matrix Row Operations:
A Mathematics Classroom Teaching Aid ........ William L. Hinrichs, 56
A Floating-Point Binary to BCD Routine ........ Marvin L. DeJong, 66

**The Apple Gazette ........ 76**

Resolving the Applesoft and Hires Graphics
Memory Conflicts ........ Jeff Schmoyer, 76
Fill the Screen with your Message: Generating Large Multi-colored
Characters using Apple Low-resolution Graphics ........ Francis A. Harvey
Rosann W. Collins and Theodore C. Hines, 80
Decrementing the For...Next and Endless Loops ........ Derek Kelly, 88

**The Atari Gazette ........ 92**

Super Cube ........ Steve Steinberg, 92
Printing Characters in Mixed Atari Graphics Modes ........ Craig Patchett, 95
Invaders from Outer Space:
An Atari Program Using One Joystick ........ David H. Markley, 97
Unleash the Power of Your Atari CPU ........ Ed Stewart, 102
String Arrays in Atari BASIC ........ Charles Brannon, 103
Reviews: Atari Super Breakout and Others ........ Robert Baker, 105
Speeding up the Player-Missile Demo ........ Larry Isaacs, 107

**The OSI Gazette ........ 110**

OSI C1P Control Functions ........ C. A. Stewart, 110
Double the Cassette Baud Rate of your
OSI Superboard II/1P ........ James Mason, 114
Review: HEXDOS 2.3: A Disk Operating System for the
OSI C1P or Superboard II ........ Ronald C. Whitaker, 115

**The PET Gazette ........ 117**

Partition and Load ........ R. D. Young, 117
Relative File Mechanics ........ Jim Butterfield, 122
COPLOT ........ Anselm Wachtel, 126
ROM Expansion for the Commodore PET ........ F. Arthur Cochrane, 130
Working with Basic 4.0 ........ Jim Butterfield, 136
Review: Papermate Word Processor ........ Paul W. Sparks, 138
Dissecting C. W. Moser's ASSM/TED 1.0 ........ Francis Turco, 142
PET File I/O and Machine Language ........ Raymond A. Diedrichs, 144
How to Get Started in Machine Code and Not Go Crazy–
With a Routine for Two Joysticks ........ Elizabeth Deal, 146
Machine Language: The Wonderful Wedge ........ Jim Butterfield, 152

**The Single Board Computer Gazette ........ 154**

The Single Board 6502 ........ Eric Rehnke, 154
KIM Tidbits: Expanding Your System ........ Harvey B. Herman, 163
Printing a Symbol Table for the AIM-65 Assembler ........ Richard F. Olivo, 165

**New Products ........ 166**

CAPUTE!: Corrections/Clarifications ........ 174
PET Program Listing Conventions ........ 174

**Advertisers Index ........ 176**


Commodore's new Home Computer Reviewed.

Easy to figure out.


You can't have too much ROM.

Kim Either!

**COMPUTE!** The Journal for Progressive Computing (USPS: 537250) is published 12 times each year by Small System Services, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 200 East Bessemer Ave., Greensboro, NC 27401.

Domestic Subscriptions: 12 issues, $20.00. Send subscription orders or change of address (P.O. Form 3579) to Circulation Dept., **COMPUTE!** Magazine, P.O. Box 5406, Greensboro, NC 27403. Controlled circulation postage paid at Greensboro, NC 27403. Application to mail at controlled circulation rates pending at Hickory, NC 28601.
 ISSN 0194-357X.

A Small System Services, Inc. Publication

COMPUTE!

# The Editor's notes...

Robert Lock, Editor/Publisher

Beginning in the May issue, you'll see a new and expanded **Reader's Feedback** column. One part will be as we've done in the past, with reader input regarding the Editor's Feedback card. (For those of you new to **COMPUTE!,** The Editor's Feedback card is your input "hotline"... I read every single one that comes in, and use them to help with planning, problems, and so on.)

The second, and new, part of the column is called **Ask The Readers.** When you're trying to solve a particular programming or technical problem, and can't get it solved, drop a short note to Ask The Readers, c/o **COMPUTE!**, P.O. Box 5406, Greensboro, NC 27403 USA. If we think it's a shared problem, we'll run it, and in later issues run responses from our panel of experts (other readers who respond with answers). Don't be intimidated if you think it's a simple problem; conversely, don't be untimidated if you think the solution is simple. For beginner's, those are frequently the toughest kind.

**Note To Our Authors— No Back Issues!**
Recently, we've been running more and more into the problem of **COMPUTE!** authors referring back to various and sundry early issues of **COMPUTE!** One problem this causes is that many of our readers don't have access to back issues. Please remember this in your articles. It's fine to refer back to an earlier issue, but please take the additional time to incorporate the information from that article that's necessary to your own point. Here's our checklist of available back issues:

**Issue 1, 2, 3, 4, 5, 6, 7 SOLD OUT**
**Issue 8, 9 and 10 Still available**

If you're interested in ordering back issues, we've an ad in here someplace called **COMPUTE's Book Corner.** You'll also be pleased to know that we're putting together our first three books:

**COMPUTE!'s First Book of Atari**
**COMPUTE!'s First Book of PET/CBM**
**COMPUTE!'s First Book of Educational Computing**

We'll tell you more about these next time, but we expect to be releasing them in the June/July time frame. We'll send your local dealer information as soon as it's available. ©

# February's Mailing Problems And Other Imprecisions. Ouch! Groan!

We have been aware that your subscription copies are reaching you later than store copies reach dealers. We are actively working on resolving the disparity, so no group of readers, whether newsstand or subscriber, is discriminated against. In this effort to promote timeliness and maintain quality, you subscribers will notice that we've returned to mailing the magazine in an envelope. This is done entirely to protect the magazine.

**February's Problem**

As far as we can tell, everyone received their magazine, albeit late. Murphey struck hard, but hopefully not again. As part of our effort to improve your speed of home delivery, we changed our mailing services to be geographically closer to our printer. Unfortunately, the local post office had not dealt with a magazine with a volume such as ours before and told our mailing personnel that the magazines needed to be bundled (e.g. by zone) only, and did not have to be bagged. Also unfortunately, the mailing service personnel believed them. The result was that the magazines went out promptly. They did not reach you promptly because, as far as we can establish, they sat in a regional bulk mail center for a week or two, waiting to be bagged. In effect, although they were in the hands of the Post Office, they were trickling out to their destinations. We apologize for the delay and concern it caused many of you. We are gradually speeding up the subscription delivery, and expect to reach par with newsstand/dealer delivery over the next few issues.

**Subscription Price Increases**

This probably isn't the optimal place to mention it, but I thought I'd take a moment to explain the new prices, and in particular the disparity between US and Canadian subscriptions. As of last issue, a twelve-issue subscription to **COMPUTE!** is $20.00 in the US, $25.00 in US funds in Canada, and for surface delivery elsewhere in the world. You're all aware of the rising costs of production, postage, etc., and the price increase, in part due to the tremendous growth in physical size of **COMPUTE!** is quite necessary.

As of January 1, our postage cost for sending the magazine to Canada increased by 93%. We found out about this increase when we went to the post office to mail some individual magazines on January 2. We are actively looking for alternative methods of reducing these costs. When we find them, we'll pass the savings along. ©

# ATARI® PERSONAL COMPUTER SYSTEMS THAT GROW WITH YOU

**Start with a better computer.** Atari computers have built-in capabilities you can't even add onto many other personal computers. Three programming formats (ROM cartridge, disk and cassette). A 57 key upper/lower case ASCII keyboard with 29 keystroke graphics symbols. 128 colors and hues. Four separate sound channels and a built-in speaker. Four controller ports. A built-in RF modulator and FCC approval for connection to any TV. Plus, nationwide Atari Authorized Service Centers. And more.

**Add memory.** The ATARI 800™ is supplied with 16K of memory. You can expand up to a full 48K of RAM with 8K or 16K Memory Modules™ you install yourself. In less than a minute. The ATARI 400's™ 8K of RAM may be expanded to 16K at Authorized Service Centers. Both may be expanded to 26K of ROM with slip-in ROM cartridge programs.

**Add peripherals.** The ATARI 410™ audio-digital program recorder. Single or dual density* disk drives. The ATARI 800 individually addresses up to four drives. Add the ATARI 850™ RS232 Interface Module. Add high speed 40 or 80-column printers. Add an acoustic modem for remote data access. Add a light pen.* And there are more Atari peripherals on the way.

**Add programs.** Choose among dozens of programs in Atari's rapidly expanding software library. Programs categories include:

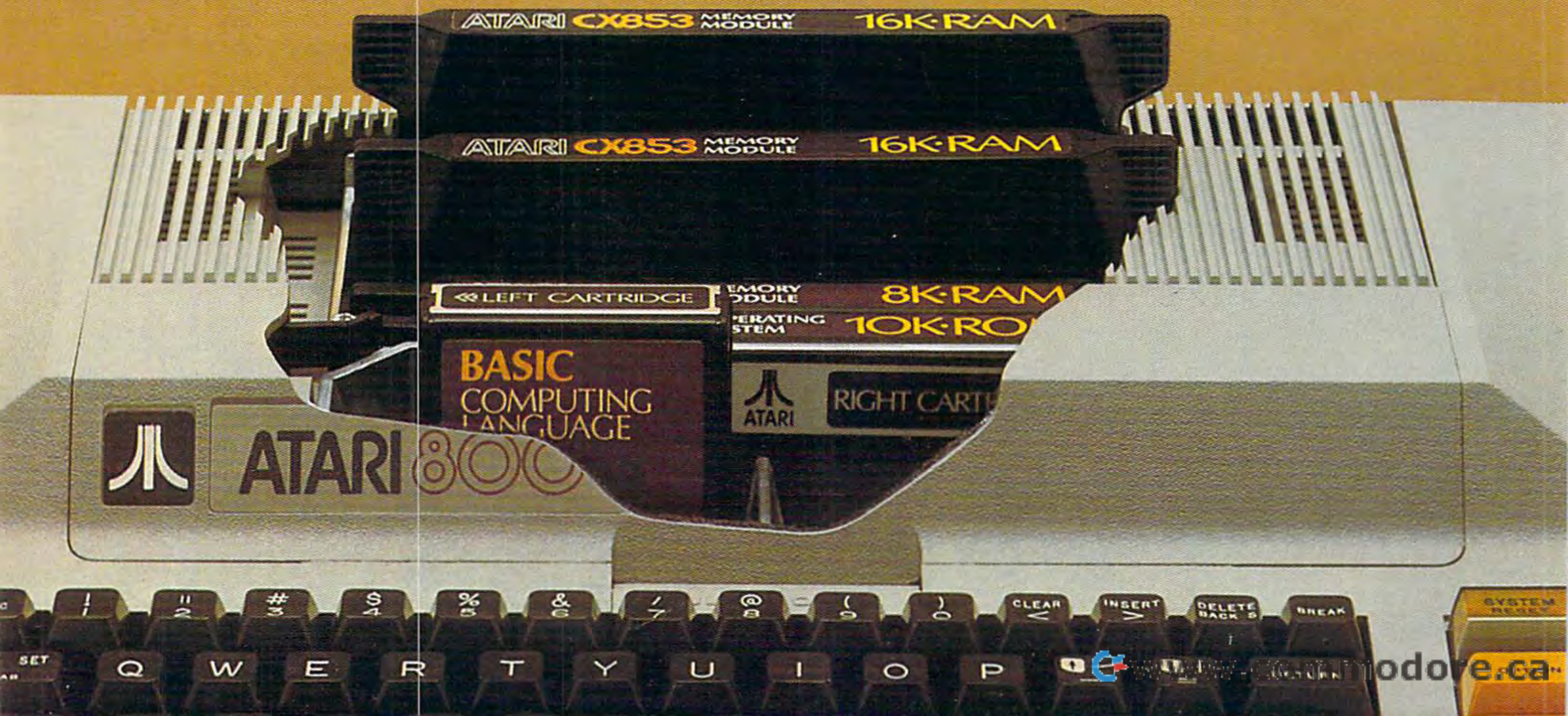
- **Personal Finance & Record Keeping**
- **Personal Interest & Development**
- **Professional Applications**
- **Education**
- **Information & Communication**
- **Entertainment**
- **Programming Languages**
- **Small Business Accounting**

**Add it up.** With Atari, you start with more. And you can build to more. Because Atari offers you personal computer systems that grow with you. Ask your Atari retailer to give you a full demonstration of Atari computers, peripherals and programs. Complete systems. Because when other people were thinking hardware and software, Atari was thinking systems.

*Available Fall, 1980



ATARI
**PERSONAL COMPUTERS**
1265 Borregas Avenue, Sunnyvale, CA 94086
Call toll-free (800) 538-8547 (Except Alaska and Hawaii)
(In California: (800) 672-1404) for the name of your nearest Atari retailer.



## Staff

Robert C. Lock, Publisher/Editor
Kathleen Martinek, Publication Assistant
Georgia Papadopoulos, Art Direction/ Production Assistance
Joretta Klepfer, Manager, Dealer/ Distributor Marketing
Carol Holmquist Lock, Circulation Manager
Dai Rees, Shipping Department
Charles Brannon, Software Lab Assistant
Terry Cash, Typesetting/Production
Harry Blair, Advertising/ Promotion Manager

## Associate Editors

Jim Butterfield, Toronto Canada
Harvey Herman, Greensboro, NC

## Contributing Editors

Robert Baker, 15 Windsor Drive, Atco, New Jersey 08004
Gene Beals, 115 E. Stump Road, Montgomeryville, PA 18936
Len Lindsay, 5501 Groveland Terrace, Madison, WI 53716
Craig Patchett, 2 Swan Terrace, Greenwich, CT 06830

## Subscription Information

**(12 Issue Year):**
**COMPUTE! Circulation Dept.**
**P.O. Box 5406**
**Greensboro, NC 27403 USA**
U.S. $20.00
Canada $25.00 (U.S. funds)
Europe: Surface Subscription, $25.00 (U.S. funds) if ordered direct, or available in local currency from the following distributors:

United Kingdom
Contact L. P. Enterprises,
8-11 Cambridge House
Cambridge Road
Barking, Essex
England IG1 18NT

Germany,
Switzerland,
Austria,
Contact Ing. W. Hofacker GMBH
8 Munchen 75
Postfach 437
West Germany

Canadian Retail Dealers should contact:
Micron Distributing
409 Queen Street West
Toronto, Ontario M5V 2A5
(416) 361-0609

## Advertising Sales

If you're in **Oklahoma, Texas or the Western States, we're now represented by Jules E. Thompson, Inc.** Give them a call for space reservations, contract/ insertion information, or questions. You can reach them through the following offices:

Southern California,
Arizona, New Mexico
**Jules E. Thompson, Inc.**
2560 Via Tejon
Palos Verdes Estates, CA 90274
213 378-8361
**Jo Ann Sullivan**

Northern California,
Pacific Northwest, Rocky Mountain States,
Texas, Oklahoma
**Jules E. Thompson, Inc.**
1290 Howard Avenue, #303
Burlingame, CA 94010
415 348-8222
**Phoebe Thompson**

If you're in the East, we're now represented by **The Gittelman Company.** You can reach them through the following offices:

New England,
New York State
**The Gittelman Company**
Statler Office Building
Suite 582
20 Providence Street
Boston, MA 02110
617 451-0822
**Joan Donahue**

New York City Metro Area,
Mid-Atlantic and Southeastern States:
Local numbers:
New York 212 567-6717
Atlanta 404 523-1252
**The Gittleman Company**
Summit Office Centre
7266 Summit Avenue
Fort Washington, PA 19034
215 646-5700
**Doug Johnson**

If you're in the **Midwest, we're now represented by GB & Associates.** You can reach them through the office of:

**GB & Associates**
P.O. Box 335
Libertyville, IL 60048
312 362-1821
**Gordon Benson**

**Address all advertising materials to:**
**COMPUTE**
200 East Bessemer Avenue
Greensboro, NC 27401 USA

**Mailing address: COMPUTE!**
Post Office Box 5406
Greensboro, NC 27403 USA

**Telephone: (919) 275-9809**

# An Interview With Dr. Chip

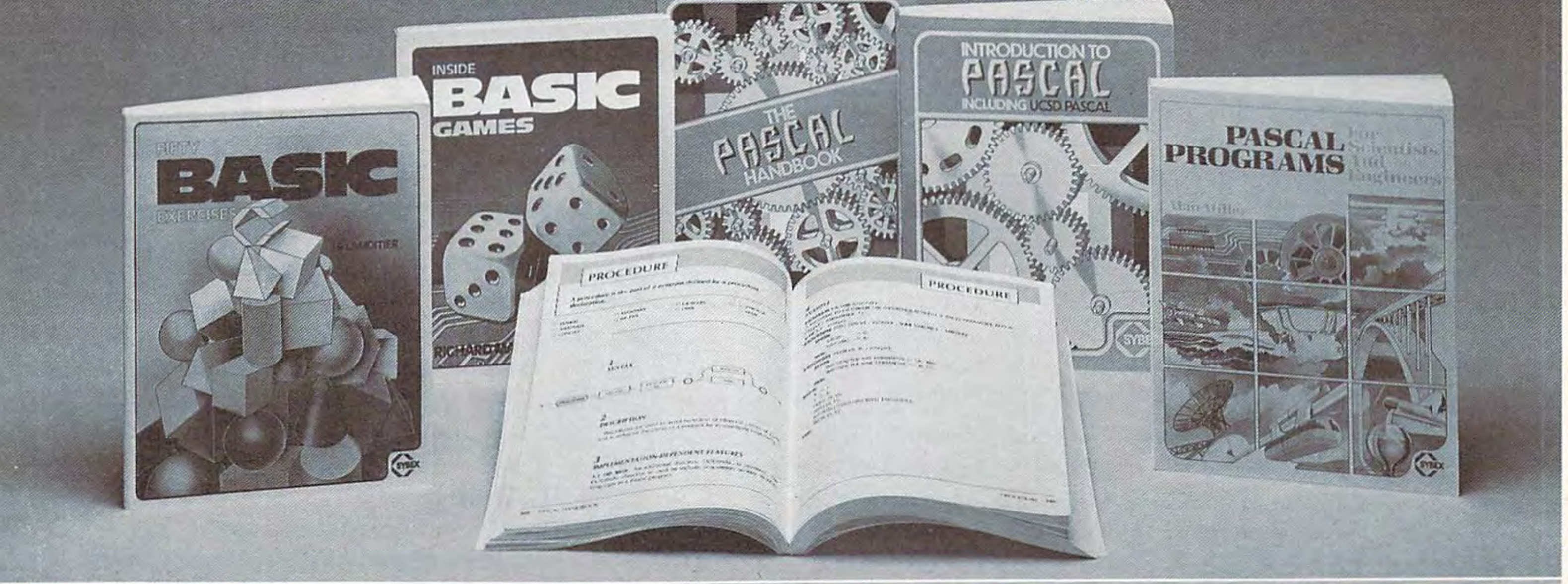
Robert Lock, Editor/Publisher

*For those readers new to* **COMPUTE!**, *Dr. Chip is Professor of 6502 Science at Figment U. He's also head of the Figment U. 6502 User's Group, a collection of 6502 users located at various Figment U. branches around the world. From time to time he consents to these interviews.*

**RCL:** Dr. Chip, it's good to see you again. I understand you've been quite busy.
**Chip:** I've been trying to piece together some late happenings in this industry, that is, when I can get any work done. That character who's trying to adopt me, the Silver Streak, has taken to calling me up on the phone in the middle of the night trying to sell me stock in his new T-shirt factory.
**RCL:** I'll have to admit, Chip, that The Silver Streak got some appreciative mail the first time he wrote you.
**Chip:** Harumph! Can you imagine me on a T-shirt...? Never mind. Back to business. First things first. Commodore has taken their dealer relations problem squarely on the chin. A recent business/financial article raked them over the coals. I'll tell you this; if they don't resolve some of their communications and customer relations problems, they'll be in a 6502 pickle. Finke (the new President of Commodore) has apparently taken direct responsibility for getting the Northeast distribution region ship-shape. That's one of Commodore's seven US regions, and I think he'll set up a model for the rest of the country.
**RCL:** I've received a good bit of mail lately from readers complaining about never receiving their Commodore US PET Users Group Newsletter. Commodore US told me last fall it was all being taken care of.
**Chip:** I suspect they'll get it together. Just as an example of how other Commodore operations treat their customers, I'll point out that Commodore Canada has been making refunds direct to their Canadian customers who sent orders to the US and never received anything. They seem to have a good handle on customer relations up there.
**RCL:** I can tell. We have extensive Canadian circulation, and I never get customer relations letters from Canada like those I get from the US readers.
**Chip:** Well, my money's on Finke.
**RCL:** I feel as though I've said this before, but we'll have to wait and see. By the way, have you found out what's happening to the KIM? (The KIM, for you uninitiated readers, is the "single-board" 6502 computer from Commodore that started the 6502 family several years ago).
**Chip:** It looks as though they havé stopped producing it. We can't find out what's going on.
**RCL:** I tried to check on it and they said there were plenty in supply. Turns out, at least from the information that I get, that the plenty in supply aren't necessarily new units. Dealers we've heard from are completely out and can't get any. One industrial client called (he has an installed base of 175 KIMs running in an industrial environment), and all of a sudden he can't get any more. No warning, no comment, no answers. I'd like to hear from anybody caught up in the midst of this.
**Chip:** Sometimes I can't figure those guys out.
**RCL:** Any other news Chip?
**Chip:** Bits and pieces. There's a lot going on in the language area. Atari's new PILOT, (previewed by David Thornburg in last issue's "Computers and Society" Column), looks quite exciting, especially for beginners at any level. There's a new language coming along from Commodore called COMAL, "Commodore Algorithmic Language". Reports have it that it's a combination of the structured preciseness of PASCAL and the simplicity of Basic. It was developed by a Danish educator and is being used extensively in the Danish educational system. We hear it may be given away as public domain material.

University of Waterloo also has some interesting projects underway at the moment, but I can't provide details until later.
**RCL:** That sounds interesting. Waterloo is where they've done extensive research and development on the advanced languages for the larger machines, isn't it?
**Chip:** That's the place. I'll fill you in on their latest projects next time.
**RCL:** What's happening with VIC, Chip?
**Chip:** Commodore's new color computer seems to have been pushed back in US introduction time to a May-June time frame. We do know they're in the process of final redesign to meet the new FCC regulations. There are mixed reports on the Apple II and compliance with the new regs as well, but we can't yet tell what Apple's doing about them. The Atari units appear to already meet the new specs from the FCC, a point consistent with their methodical approach to this marketplace from the beginning. ©

# SYBEX BREAKS THE LANGUAGE BARRIER




*Let the chips fall where they may. These two books on BASIC assure comprehension and competence.*

**■INSIDE BASIC GAMES**
*by Richard Mateosian* teaches interactive game design and BASIC programming through thorough analysis of eight different kinds of computer games. Programs are presented in Microsoft BASIC with versions for PET/CBM, TRS-80 and APPLE II.
350 pp., 120 illustr., Ref. B245, $13.95

**■FIFTY BASIC EXERCISES**
*by J.P. Lamoitier* provides the surest way of learning BASIC—actual practice. Graduated exercises, each containing a detailed explanation, flowchart and sample run, develop skill and competence rapidly. Applications include mathematics, business, operations research, statistics and more.
256 pp., 194 illustr., Ref. B250, $12.95

*Get in gear and accelerate your programming productivity with Pascal's power.*

**■INTRODUCTION TO PASCAL**
*by Rodnay Zaks* is a simple yet comprehensive guide to standard and UCSD Pascals: step-by-step presentation with exercises for beginners, complex concepts and extensive appendices for experienced programmers. An indispensable book for everyone who wants to learn Pascal programming.
320 pp., 100 illustr., Ref. P310, $14.95

**■THE PASCAL HANDBOOK**
*by Jacques Tiberghien* is an easy-to-read, easy-to-use dictionary containing all the features for most existing versions of Pascal (Standard, Jensen-Wirth, OMSI, UCSD, HP1000, Pascal/Z). Over 180 entries, arranged alphabetically; each includes definition, description, syntax diagram, details of implementation, variations and examples. The perfect reference tool for any Pascal user.
500 pp., 150 illustr., Ref. P320, $14.95

*Scientists and engineers involved in significant work have been delayed by having to reinvent algorithms for a new computer language. No more.*

AVAILABLE MAY 1981

**■PASCAL PROGRAMS FOR SCIENTISTS AND ENGINEERS**
*by Alan Miller* is a comprehensive collection of frequently used algorithms for scientific and technical applications programmed in PASCAL. This time saving book includes programs for curve fitting, fast Fourier transform, approximations, random number generation, integrals, statistical techniques and more.
250 pp., 80 illustr., Ref. P340 ☐$16.95 paper, ☐$25.00 cloth

## MORE SYBEX BOOKS

**■YOUR FIRST COMPUTER** *by Rodnay Zaks*
280 pp., 150 illustr., 2nd Edition, Ref. C200A, $7.95

**■MICROPROCESSORS: FROM CHIPS TO SYSTEMS** *by Rodnay Zaks*
420 pp., 250 illustr., 3rd Edition, Ref. C201, $12.95

**■MICROPROCESSOR INTERFACING TECHNIQUES** *by Rodnay Zaks & Austin Lesea*
464 pp., 400 illustr., 3rd Edition, Ref. C207, $15.95

**■PROGRAMMING THE 6502** *by Rodnay Zaks*
392 pp., 160 illustr., 3rd Edition, Ref. C202, $12.95

**■6502 APPLICATIONS** *by Rodnay Zaks*
288 pp., 207 illustr., Ref. D302, $12.95

**■6502 GAMES** *by Rodnay Zaks*
304 pp., 140 illustr., Ref. G402, $12.95

**■PROGRAMMING THE Z80** *by Rodnay Zaks*
620 pp., 200 illustr., 2nd Edition, Ref. C280, $14.95

**■PROGRAMMING THE Z8000**
*by Richard Mateosian*
312 pp., 124 illustr., Ref. C281, $15.95

**■THE CP/M HANDBOOK (With MP/M)**
*by Rodnay Zaks*
336 pp., 100 illustr., Ref. C300, $14.95

MAIL TO:
SYBEX DEPT. C 41
2344 SIXTH STREET
BERKELEY, CA 94710
PHONE ORDERS:
INSIDE CA 415/848-8233
TOLL FREE OUTSIDE CA 800-227-2346

*or at book and computer stores everywhere*

NAME ______________ ☐ SEND ME YOUR FREE CATALOG
ADDRESS ______________
CITY ______________ STATE ________ ZIP ________
ADD ☐$1.50/book UPS or ☐75¢/book 4th class mail or ☐$8/book overseas airmail
(CA add tax) Total Amt. Enclosed ________ OR CHARGE MY ☐VISA ☐MC ☐AM EX.
CARD NO. ______________ EXP. DATE ______________
SIGNATURE ______________

# Computers and Society

David D. Thornburg
Innovision
P.O. Box 1317
Los Altos, CA 94022

My last column concentrated on the software interface between people and computers. As I said at that time, the mechanical devices through which we communicate with our computers are no less deserving of attention. I find it both sad and amusing to see that the principal method by which humans interact with computers is through a keyboard whose 100 year old design is based on the limitations of antiquated mechanical devices.

I know that I touched on this topic in a previous column, but a phone call I received a few weeks ago has rekindled this issue for me, and perhaps it is time to talk about keyboards again. It was my pleasure to receive a phone call from Dr. Mary Humphrey in Canada. She told me of her experiences in working with children who use computers extensively in an educational environment. As I recall, these children range from the educationally handicapped to the gifted, and cover a wide range of ages. After working with the computers for a while, the children are asked to describe what things they like and what things they dislike. She told me that the major problem that children encounter with the computer is the arrangement of keys on the keyboard. "The keys are all mixed up." Some children even propose alternative keyboard layouts — all of which are alphabetical.

I know that this is a tender topic, and one which has been with us for a long time, but I persist in thinking that there is a golden opportunity, right now, for us to improve this interface between people and computers.

This month we will explore the development of the commonly used Sholes keyboard, the evolution of alternatives, and the promise of a new keyboard environment for the many millions of new computer users who have no training in typing.

Did you know that the 100 year old layout of the keyboard used on most personal computers was intentionally designed to be hard to use? This arrangement was developed in 1872 by typewriter inventor C. Latham Sholes (1) and his attorney, James Densmore, to overcome a major problem in the design of Sholes' original typewriter. Originally the keys were arranged in alphabetical order. Unfortunately, this arrangement made it very easy for certain commonly used keys to be typed in such quick succession that adjacent type elements would jam together before hitting the ribbon. To overcome this problem, Sholes and Densmore placed the most commonly typed letters as far apart in the type basket as possible, and the result was the QWERTY keyboard we have today (see Figure 1). The name QWERTY is derived from the first five keys in the top alphabet row of this keyboard, and is a folksy name for the Sholes arrangement.

Most of the effort applied to improvements on the Sholes keyboard has been geared towards improving typing speed and reducing operator fatigue.

**...the 100 year old layout of the keyboard used on most personal computers was intentionally designed to be hard to use...**

In 1932, after many years of work, August Dvorak (2) suggested a keyboard arrangement similar to that shown in Figure 2. As with the Sholes keyboard, the DSK (for Dvorak Simplified Keyboard) requires a lot of training to use effectively. Its principal advantage for touch typists is that skilled DSK users can type at up to twice their previous typing speed with less fatigue.

If DSK is so much better than Sholes, it is logical to ask why the improved keyboard has not displaced the older inefficient model, especially since the original mechanical limitations leading to QWERTY have been overcome for many years. There seem to be two causes for this failure. The first is the inertia associated with displacing the many millions of Sholes keyboards in use today. Second, there is the understandable resistance on the part of typists trained on the Sholes arrangement, each of whom would have to spend about a month making the transition to a new and (presently) hard to obtain keyboard.

Neither of these problems has deterred other researchers however, and many other alternatives to QWERTY have been proposed in recent years. Among the more interesting concepts that have been studied is the "chord" keyboard. This idea, pioneered by E. T. Klemmer at IBM (3) in 1958, entails the use of a keyboard with only ten keys, one for each finger. Letters are typed by pressing the correct sequence of keys at the same time, much as one would play a chord on a keyboard instrument. With ten keys, 1023 different patterns can be generated. To make it easier on the user, Klemmer didn't ask the user to press more than two keys at once. Users of this system were able to type at more than 40 words per minute after intensive training. Klemmer felt that the real power of his keyboard would come from using additional finger chords for the entire words.

While Klemmer's keyboard was designed from the human's point of view (commonly used letters used single keystrokes and favored the stronger fingers), other keyboard designers have decided that people should change their behavior to match that of the machine. Several designers have introduced five-key keyboards on which the user is expected to type the alphabet in raw ASCII-like code. While having certain appeal to some high-tech acquaintances of mine, most of these efforts have, thankfully, died a quiet death.

My concern is that neither the Sholes, the DSK arrangement, nor any of the chorded keysets makes any sense to the novice user. As the personal computer market continues to expand, an increasing number of people are being asked to type on a keyboard which makes no sense whatsoever. Anyone who has watched a child use a computer has seen the intense concentration with which he or she scans the keyboard looking for the right key. These novice "hunt and peck" typists typically use the index finger of one hand to do their typing, with "advanced" novices using the index fingers of both hands. When one considers the myraid applications for the computers used by novice typists, it seems almost criminal that a powerful modern tool like the personal computer should be constrained to use a keyboard designed as an apology to the limitations of nineteenth century mechanical skill.

Since you, most likely, have had some exposure to the Sholes keyboard, you might think I am overstating my case. You should perform the following experiment: Look at the DSK keyboard shown in Figure 2 and type: The quick brown fox jumped over the lazy dog.

*Now imagine how a child feels when presented with QWERTY for the first time!*

I feel that a solution to this problem exists. It is logical to ask why I think the time is ripe for change when Dvorak had so much trouble forty years ago, and since none of the other systems has moved far from the research laboratory.

The answer to this question is that, for the first time since 1873, a major keyboard market has opened for which the purchasers and users of these keyboards are not already skilled typists. The personal computer market in the United States jumped from almost nothing in 1977 to 150,000 machines in 1979. The annual sales figure appears to be doubling every year, and sales this year might reach a rate of over one million computers per year. While there is no reliable figure on the saturation level of this market, conservative estimates of 50 million computers represent a probable lower bound for this marketplace. This massive market, coupled with the fact that the overwhelming majority of new personal computer users are not already "touch typists" is what gives encouragement to the concept of a new keyboard arrangement.

In thinking about new keyboard arrangements useful to novices, it is fairly obvious that the keys should be arranged in alphabetical order. Consider the environmental forces which lead to this conclusion. Children are taught their ABC's from the moment they can talk. One often hears children singing the alphabet song:

*"Ay bee cee dee ee eff gee,*
*aitch eye jay kay,*
*ell em en oh pee . . ."*

On the other hand, I have never heard a child sing:

*"Kew doubleyou ee are tee wy,*

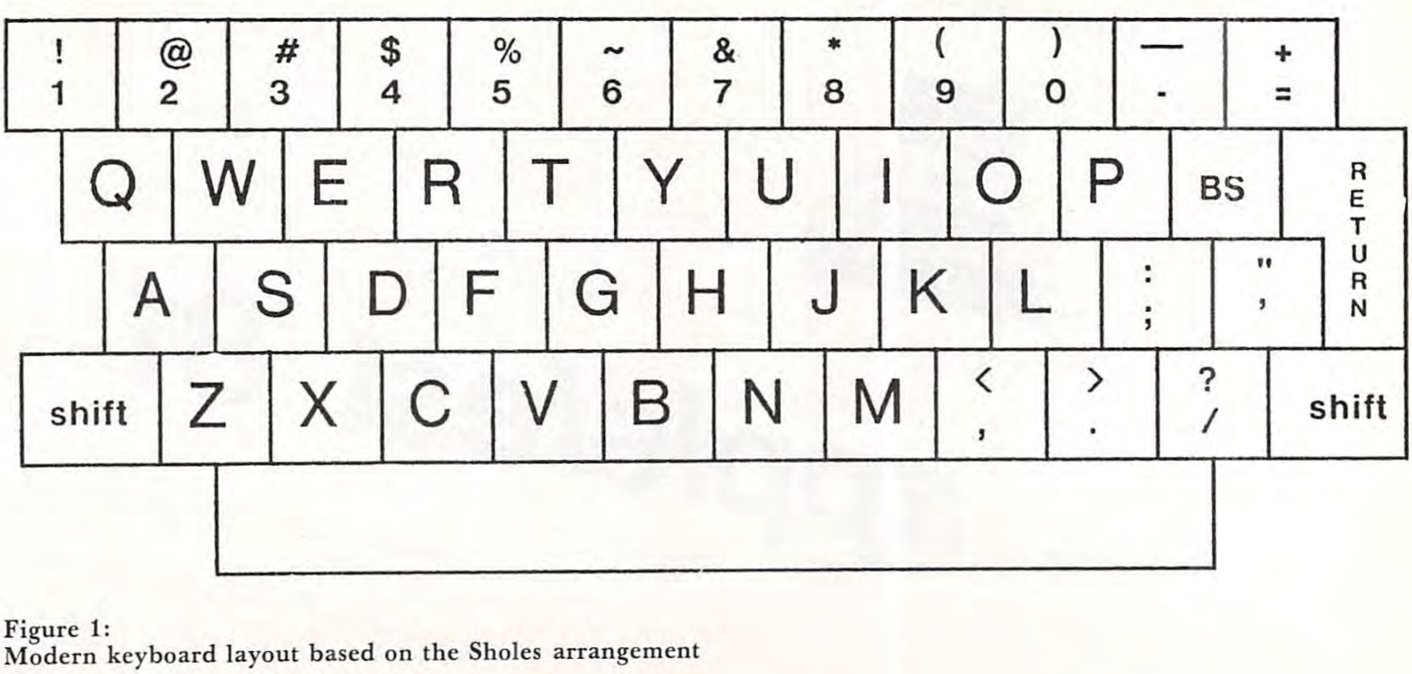



**Figure 1:**
**Modern keyboard layout based on the Sholes arrangement**

*you eye oh pee,*
*ay ess dee."*

Later on, as reading skills are better developed, we teach our children how to use the dictionary. Once again, they encounter the letters in alphabetical order. The reason that words in dictionaries are arranged in alphabetical order is very simple. The user's pre-existent knowledge of the letter sequence allows rapid location of a chosen word among thousands of other words. Those of you who have learned a language which uses a different alphabet (such as Russian) probably experienced some frustration in learning to use a dictionary in that language until the "new" alphabetical order became entrenched in your mind.

If the QWERTY arrangement is so good, then one must ask why office workers who use it for typing don't also use it for filing documents.

As it turns out, some enlightened vendors of consumer products have realized the value of our early childhood education, and offer alphabetic keyboards on their products. The Texas Instruments' Speak & Spell, Mattel's Brain Baffler and the Craig "pocket translator" immediately come to mind. In one of the more ambitious projects of considerable relevance to computer using educators, Children's Television Workshop used alphabetical keyboard arrangements in the seventy-odd Apple computers located at Sesame Place in Bucks County Pennsylvania.

If these domestic projects aren't enough, consider the fact that the government of France will be performing a test of their electronic phone directory system by installing 250,000 computer terminals in people's homes this year. Within a few years three million of these terminals will be installed. A few years after that, perhaps 37 million such terminals will be in daily use - each of them with an alphabetic keyboard layout. The point is not just that alphabetical keyboard arrangements are possible, but that some companies are actually finding that these arrangements are commercially successful.

There are many merits to using an alphabetical keyboard arrangement. Unlike either of the keyboards shown in Figures 1 and 2, the user does not have to scan the whole keyboard to find a given key. This is important since the foveal regions of a user's eyes can only be focused on one or two keys at a time. Once a key is perceived (assuming it is the wrong key) the logic behind the alphabetic layout helps to reduce the time required to find the desired letter or symbol. The use of color coded keytops (with vowels having a different color than consonants, for example) may also be of benefit to some users.

The development costs associated with alphabetic keyboards are no different from those associated with the Sholes arrangement, so there is no particular reason for this new keyboard to be more expensive than the more traditional model.

Can a logical keyboard designed for novices find its way to the marketplace? The answer is yes if those millions of us who are buying computers make it happen. The tyranny of QWERTY can be stopped at last!


**References:**

1. C. L. Sholes, U. S. Patent 207,559 (1878).
2. A. Dvorak, et al., "**Typewriting Behavior: Psychology Applied to Teaching and Learning Typewriting**", American Book Co., New York (1936).
3. E. T. Klemmer, "**A Ten-Key Typewriter**", IBM Research Memo #RC-65 (1958).

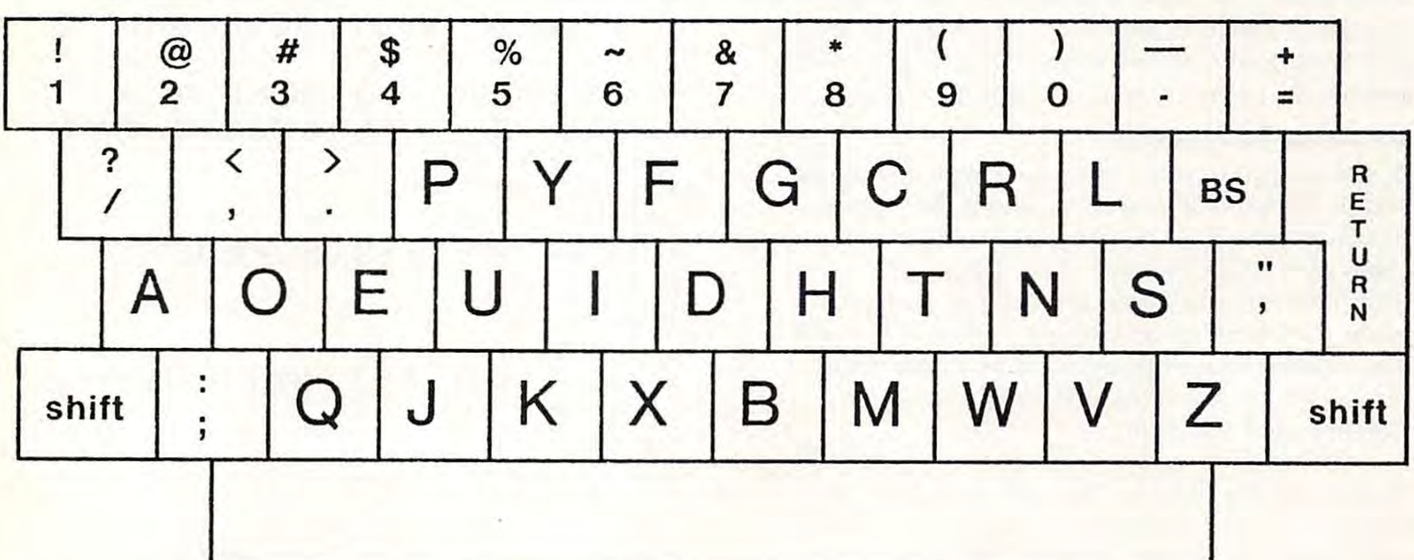




**Figure 2:**
**Modern keyboard layout based on Dvorak Simplified arrangement (except for numerals)** ©

# The Beginner's Page

Robert Lock
Editor/Publisher

*This page is a continuing, "re-cycling" feature in* **COMPUTE!**. *It consists of a set of articles that repeat, in sequence, across issues. Thus, if you're a beginner to computing, you can pick up the series whenever you start with* **COMPUTE!**, *and within four or five issues, have the set. By then, you'll find you've advanced far beyond where you are now, especially if you have your hands on a computer.*

## Part Two

### Access to Resources

If you're just getting started, you'll find several important sources of information are available to you. Beyond the obvious channels, such as magazines and books, you'll quickly discover a community of users. Your local computer store can help there. They can frequently specialize in, or at least cover, your particular computer. Depending on the size of your user community, you may even find seminars for beginners, a lending library of back issues of magazines, and so on.

If you're in an area where activity hasn't yet grown to the point of established clubs, or there's not a computer store around to provide such information, drop a note to your machine's manufacturer or give a call to the district office. They may be able to provide the names of some clubs in your region.

### Learning To Program

Assuming you have no experience with computers, and no established local users group for support, where do you start? Well, you have the manuals that came with your computer. And depending on the manufacturer, you'll find there are several good books on BASIC programming around that will help. One sure method of plunging in is to take some of the simple programs that we present here, for example, and use them.

Once you've entered a program, and have it working as described by the author, go back and figure out how it works, and why it works. You'll soon find you can start to make additions to programs from books or magazines that help "customize" them for your own use. This is an ideal way to learn.

My advice is to start at the very beginning, and use some feature of BASIC until you understand its usefulness and purpose. Continue to add on features as you need them or want to understand them. Above all, don't get frustrated. The best way to learn to program is to program.

Here's a sample of what I mean. Type this program into your computer (press return after each line):

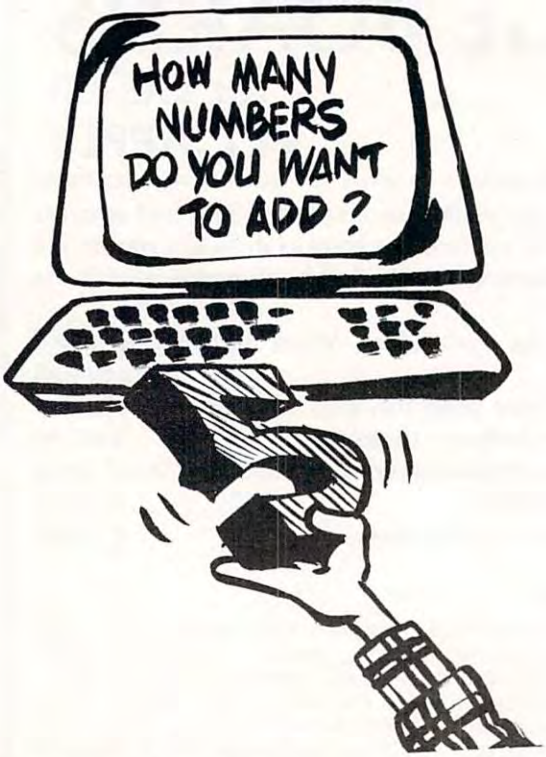
```
10 REM PROGRAM #1
20 PRINT "HELLO"
30 END
```


**NOW TYPE RUN, AND PRESS RETURN.**

Your computer should print HELLO on the screen, followed by READY. Ta Da! A working program. Surely, you say, I bought this machine to do more than this. Of course you did. Let's turn our sample into a more useful program, adding a few more features common to all our BASIC languages.

```
10 REM PROGRAM TO ADD NUMBERS
20 PRINT "HOW MANY NUMBERS DO YOU
   WANT TO ADD?"
30 INPUT N
40 FOR I = 1 TO N
50 PRINT "ENTER THE NUMBER."
60 INPUT J
70 K = K + J
80 NEXT I
90 PRINT "THE SUM OF THE NUMBERS IS
   ";K'
100 END
```

When your computer asks how many numbers you want to add, type in some small number like **5.** It will then ask you, 5 times, to **"Enter the number."** Each time, type in one number that you want to add to the sum.

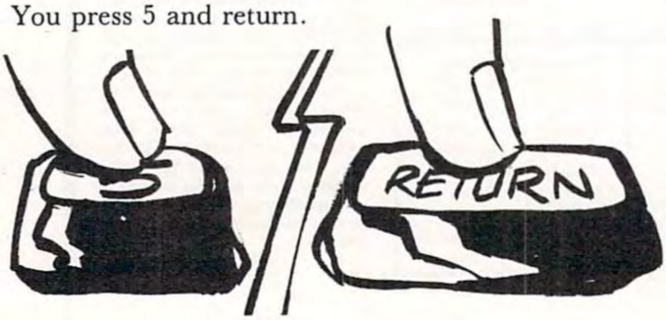



Here's what you should see on your screen after typing run.

**HOW MANY NUMBERS DO YOU WANT TO ADD?**

?■

You press 5 and return.




Now you should see:

ENTER THE NUMBER.

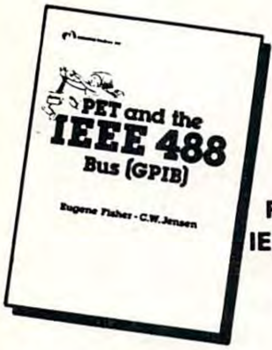
?■

Here you should type in the first number of your group of 5, and so on (5 times) until the computer says:

**THE SUM OF THE NUMBERS IS:__________**

This will be the sum.

Try this one out, and next time we'll expand it further, explaining how it works, and some nice ways to make it work more usefully. ©




**PET and the IEEE 488 Bus (GPIB)**
by E. Fisher and C. W. Jensen

This is the only complete guide available on interfacing PET to GPIB. Learn how to program the PET interface to control power supplies, signal sources, signal analyzers and other instruments. It's full of practical information, as one of its authors assisted in the original design of the PET GPIB interface.

#31-4 $15.99


NEW PET/CBM edition

**Some Common BASIC Programs**
by L. Poole, M. Borchers, C. Donahue

**76 Programs** you can use even if you don't know BASIC. This book gives you a variety of math power including personal finance, taxes and statistics as well as other programs you'll want like Recipe Cost and Check Writer. All programs can be run on a PET or CBM with 8K or more.

#40-3 $14.99


PET owners can purchase the programs ready-to-run on cassette or disk. Use the book as a manual for operating instructions and programming options.

Disk #33-0 $22.50
Cassette #25-X $15.00

**Practical BASIC Programs**
ed. Lon Poole

These are 40 easy to use programs that each do something useful.

Income averaging, checkbook reconciliation, statistics, factorials, temperature conversion and musical transposition are just a few. It offers a wealth of practical computing power. Includes write-ups, program notes and instructional examples to help you realize the potential uses of each program.

#38-1 $15.99

**6502 Assembly Language Programming**
by L. Leventhal

Increase the capabilities and performance of PET (and other 6502-based computers) by learning to program in assembly language.

#27-6 $16.99

# New for your PET

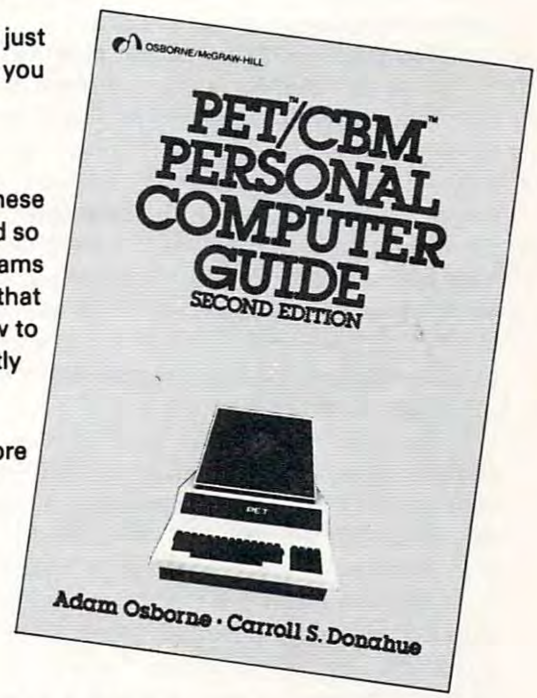
## from OSBORNE/McGraw-Hill


## PET™/CBM™ Personal Computer Guide Second Edition

by Adam Osborne and Carroll S. Donahue

The PET/CBM Personal Computer Guide is a step-by-step guide that assumes no prior knowledge of computers. If you can read English, you can use this book.

This book provides the important information and documentation that PET/CBM users have sought for so long. After reading this book you will have a good understanding of what a computer — especially the PET/CBM computer — can do for you. If you've just bought a PET or CBM this is the book you must have to really understand your computer. By using the examples found in this book, you will quickly get your PET/CBM up and running. These examples are thoroughly documented so you can learn how and why the programs work. It is the "how" and the "why" that are important if you want to learn how to make your PET or CBM work efficiently for you.




This second edition contains even more useful information than the first edition of this book. The guide contains a wealth of information on everything from keyboard variations to a detailed description of PET and CBM memory.

Included are:

Complete operating instructions for

- keyboard
- tape cassette
- disk

Description of all CBM BASIC statements
Optimal programming techniques including

- input/output programming
- file handling
- screen editing

Solutions to programming problems
CBM capabilities and limitations

#55-1 $15.00

| Book/Cassette/Disk | Price | Quantity | Amount |
|---|---|---|---|
| 27-6 6502 Assembly Language Programming | $16.99 | | |
| 30-6 PET Personal Computer Guide | $15.00 | | |
| 31-4 PET and the IEEE 488 Bus (GPIB) | $15.99 | | |
| 40-3 Some Common BASIC Programs PET/CBM ed. (book) | $14.99 | | |
| 25-X Some Common BASIC Programs PET Cassette | $15.00 | | |
| 33-0 Some Common BASIC Programs PET Disk | $22.50 | | |
| 38-1 Practical BASIC Programs | $15.99 | | |
| | | California resident tax | |
| | | Shipping | |
| | | Total Amount Enclosed | |

California residents add 6% sales tax.
S.F. BART residents add 6½% sales tax.

Shipping: (Shipping for large orders to be arranged)

- All foreign orders $4.00 per book for airmail
- $0.45 per book 4th class in the U.S. (allow 3-4 weeks)
- $0.75 per book UPS in the U.S. (allow 10 days)
- $1.50 per book special rush shipment by air in the U.S.

Cassettes and Disk:

- No additional charge in the U.S.
- $1.50 each foreign airmail

**OSBORNE/McGraw-Hill**
**630 Bancroft Way, Dept.G5**
**Berkeley, California 94710**
**(415) 548-2805 · TWX 910-366-7277**

Name: ______

Address: ______

City: ______

State: ______ ZIP: ______

Phone: ______

☐ VISA ☐ Master Card

Credit Card No.: ______

# An Applications Commentary
# Stimulating Simulations

Gregory R. Glau
Prescott, AZ

Well, there it sits: *your computer.* You've spent the past few months learning BASIC, writing all kinds of programs and learning how to use and interact with the computer and its tape or disk storage system. And, if you do say so yourself, you've turned into a pretty darned good programmer!

But gosh — after all those months — *there it sits.*

Sure, it's still fun to demonstrate a game or two if a friend stops by, or perhaps you've invested in a modem and can access a Network... but by and large, if you made a list of the reasons why you bought your computer in the first place, a list of 'all the things I'll be able to do,'... well, *you've done them.* You find yourself spending a half hour or so a day working on the computer, perhaps keeping your checkbook up-to-date, or making a budget listing, or keeping track of the amortization for the new car... but it mostly just sits there!

Suddenly it dawns on most of us that finding the answers is not the problem — we understand BASIC enough and disk data files and tape loading that we can figure out an answer to a problem. The difficulty is in finding the problem itself, in *asking the questions,* in figuring out things we can have the computer do. And not just ideas that take ten minutes and display a cute drawing on the screen, or a program to print all odd numbers between ten and a thousand. The whole purpose of any computer is to save time and make us more efficient in our work and/or home affairs.

But where do you get ideas to — as the ad says — simplify your life?

The first place to look is to examine any and all paperwork you handle, whether you use your computer in your home or business.

The businessman has some obvious needs — invoicing, monthly statements, payroll. We've found that our APPLE II saves an hour or so every week by figuring and printing payroll checks. So-so. But it also automatically balances all the figures and keeps them on a disk for all employee's year-to-date totals. The old way, balancing those figures by hand every quarter, literally took hours and hours. Here's a case where the initial time-saving didn't seem too terrific, but since everything is always 'in-balance' and up-to-date, over a period of a year it'll save hundreds of dollars in labor costs.

Accounts receivable and accounts payable are obvious savings, compared to the way we used to do things (and many small businesses still do) — by hand posting. Right out of the Middle Ages! Sending statements used to take a day... now it takes two hours.

But the businessman has to take a closer look at the other paperwork he's involved with.

How about keeping mileage and cost-per-mile records for any vehicles you own (the homeowner can do the same)?

While the businessman is making sales projections on his computer, the homeowner can project a budget/expense program on his.

**The difficulty is in finding the problem itself, in asking the questions, in figuring out things we can have the computer do.**

The businessman can keep a running record of each employee's job efficiency (is he making or losing money for you?)... and the homeowner can keep a record of what his wife (or husband) spends!

The businessman is able to *project* what a major investment will do to his cash flow and net profit, and the homeowner, with the right program, can readily tell if he can afford that Coleman tent-trailer.

The businessman can see exactly what will happen to his profits *if* sales drop ten percent. *If* the housing industry stays in its slump. *If* the Summer gets hot and his air conditioning units sell like crazy. And the homeowner can predict his cash situation nine months from now when the new arrival is due (they don't let you take babies home unless they're paid for!).

Could you forecast the weather based on past trends and current data?

Would you like to know who's got the best chance of winning the second race tomorrow?

Are you interested in your youngster's projected SAT test score?

Could you plan what you'd do if your health insurance costs increased 15% next month?

Would it help to know that if you spent X dollars for insulation that you'd save Y dollars on air conditioning costs? Or if you install that solar water heater what effect it'll have on your tax return?

Is it useful to understand that if you could somehow *save* so many dollars per month over the next so many months that you could save X dollars in interest for that awning/cooler/television/exercycle you wanted, by paying cash instead of financing it?

Wouldn't it be interesting to program your computer to analyze the miles you and your next-door neighbor and Joe (he lives down the block) and your brother-in-Law around the corner each drive to work every day? To perhaps route a carpool for the four of you? To get an exact projection of the gas you'd save? To perhaps do the same thing with you wife and her friends...who all drive their kids to school? To program your computer with the basic items your family uses (Cheerios, ketchup, kleenex, soup,

**Anything that you now do with data can be adapted into a simulation, a projection of the future and its results on you.**

whatever), along with the average rate of usage... and then printout a grocery list *before* you ran out of anything?

Anything that you now do with *data* can be adapted into a simulation, a projection of the future and its results on you.

This — the area of simulation — is perhaps the most powerful thing a computer can do for any of us. Yes, it's wonderful to have the computer keep track of our monthly bills, to have it do the mundane record-keeping chores, but how much more invaluable it becomes when we project the future with our ideas!

Let's see...if we're making X number of dollars per year now, and inflation will average Y percent over the next so many years and my raises will give me a total income of X dollars at that time...will the kids be able to go to College? Will the wife and I be able to buy that motorhome?

The whole idea is to project — and thus *predict* — the future.

Now what if I get laid off work for ten days over the next year? What if the wife could find a part-time job (perhaps running ammortization schedules for banks in your area on your Computer?) — what would her income do to our net total? How about that duplex we've been wanting to buy and fix up? If we pay so much for it and it costs us this much to fix it up...we should be able to rent it for... And what will it do to our income tax situation to get all that depreciation from it as a rental? How much would I have to save every month to have enough to pay cash for a new car, say, in twenty months?

Well, you get the idea — *simulation* on your computer can help you find out what the future will bring, and perhaps in time to allow you to change things, if you don't like what it displays for you on your CRT!

**The whole idea is to project — and thus predict — the future.**

Most simulations, by the way, can be generated from past data. The businessman can project labor costs based on the jobs he sold last year. The homeowner can predict what his salary needs are by basing his estimates on last year's budget printout (My God, Helen, did we spend *that much* on shoes?)

Once you've exhausted all the record-keeping and paper-work handling and forms-filling-out things that you once did by hand (but now your APPLE II or TRS-80 or OHIO SCIENTIFIC or PET does better and faster), the logical place to turn to is this area of *simulation.* And after all, there are only so many record-keeping chores we have to take care of, and once they're accomplished — and the computer is being used only an hour or so a day — the ideal place for one's creativity is in simulation.

And the best part is that you'll quickly discover that one idea leads to another — you might start projecting your net income and end up looking at life insurance values in relation to education costs ten years down the road. This in turn might lead to new record-keeping ideas, which will give you more simulation directions...

So, here are a baker's dozen ideas for **simulation** in areas the average programmer should have an interest in (after all, it's your money) in addition to the ideas already mentioned, so that perhaps a few will sound good to you and be of some help in your own financial planning:

**1.** Design a program to show you how much money you'd have to save weekly/monthly/yearly to end up with X dollars Y number of years from now. By being able to change every combination, you'll soon find a plan you can afford that'll give you the cash you need...when you need it.

**2.** Project the cash savings by replacing your present air conditioning system/furnace/water heater/cookstove with a new energy-savings one. How long will it take to pay for itself at the present gas or electric rates? What if the rates increase X percent?

**4.** Printout the values of your stock portfolio if inflation goes up so much percent while the market goes down X percent.

**3.** Compare the overall costs of remodeling the basement vs. the cost of a new home.

**5.** Figure out what salary you'll need with an annual raise of X dollars, an inflation rate of Y percent...when you son starts college in 19—.

**6.** Estimate the cost savings of adding an evaporative cooler to your air conditioning system.

**7.** Find out what's best for your own situation by comparing the net cost of life insurance: whole life gets cash value plus dividends while term insurance doesn't. Which costs less?

**8.** Find out how much of a raise you'll have to ask for based on X percent inflation this year.

**9.** Get a budget projection comparing the number of movies you go see (the cost of tickets & popcorn) to what that new cable TV deal costs.

**10.** Discover what your annual car expenses are over the next five years if you either (A) keep your present car or (B) buy that new gas-saving model. Factor-in different per-gallon costs, and don't forget the new one costs more for insurance and license plates.

**11.** Figure out exactly how much life insurance you need right now — determine how much income your wife will need for how many years.

**12.** If you can save so many dollars per month, display or printout the various options you might have (money market funds, regular savings, certificates of deposit, mutual fund programs); which is best for you?

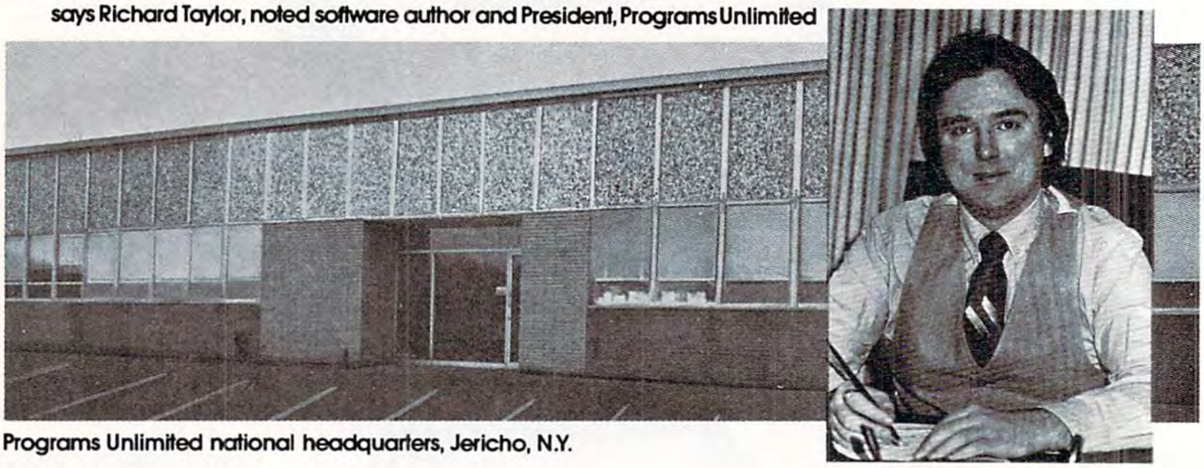
**13.** Find out exactly what happens if you buy a rental unit, by using forced inflation. The duplex might cost you X dollars, but if you spend Y dollars to fix it up, how much will it increase in value? Compare this duplex with that triplex. Which is best? If you fix a place up and then keep it, how much can you raise the rents? What does it do to your tax return?

Somewhere in all these ideas is a problem that *you need a solution to.* So...get your programming pad out and get to work...and all of a sudden you'll find your computer in almost constant use...and not just sitting there any more!

*Editor's Note: Once you have one of these programs up and running, write a tutorially descriptive article to go with it, and send it in to me at* **COMPUTE!** *We'll look forward to it, and so will all our readers who aren't quite programmers yet. RCL.* ©

# "The ideal software source...
# "We couldn't find it.

says Richard Taylor, noted software author and President, Programs Unlimited



Programs Unlimited national headquarters, Jericho, N.Y.

We know what it's like out there. Because that's where we came from.

Before we decided to become software entrepreneurs, we were just like you—micro-computer enthusiasts searching and sifting through myriad magazines for the ideal mail order software source.

What we hoped to find was a single entity that offered an ultra-wide selection of software (and peripherals) for our particular micro-computers. That pre-selected only the best of the many similar-sounding programs that reached the market each month. That could give us personal assistance with the purchase-decision process. And that stood behind its products with a no-holds-barred guarantee.

Isn't that what you've been looking for too?

When we couldn't find it, we decided to become it. And PROGRAMS UNLIMITED was born.

**An Information Services Department: To Solve, Not Sell**

We're professionally staffed and solidly financed. We set high standards for the programs we carry, whether developed by ourselves or others; if they don't meet our standards, we know they won't meet yours.

We have a computer Bulletin Board to keep you apprised of our latest offerings, and a toll-free number for telephone orders. In addition, we've established an Information Services Department; its primary function is not to sell merchandise, but to provide expert answers—from Richard Taylor and others—to questions you may have about your system.

**Our Guarantee is *Really* a Guarantee**

We know who we are and what we're up to, but **you** may not—yet. That's why we give you the best guarantee we can think of: if for any reason you're not satisfied with any PROGRAMS UNLIMITED purchase, return it within 14 days for a full, no-hassle refund.

Think we're going overboard? You're right.

We know that the future of our business is **repeat** business. We want you to be satisfied first time, second time, **every** time you buy.

We want to earn the right to be **your** *software source.*

**the software source™**

**PROGRAMS**

# The Commodore VIC-20: A First Look

David D. Thornburg
Innovision
P.O. Box 1317, Los Altos, CA 94022

It seems surprising to realize that it was only three years ago when the Commodore PET started to show up in dealer's showrooms. At a price of $799, this astounding machine (along with its temporal counterparts from Apple, Radio Shack, and others) broke down many price/performance barriers in the computer field.

It wasn't much later that people started wondering if the small computer industry was going to copy the calculator industry, with ever more sophisticated machines being made available at the same, or lower, cost than that of earlier models. Mythical price barriers were erected, only to be smashed by new product announcements.

It was the breaking of the $400 barrier last fall by Radio Shack's TRS-80 Color Computer that caused some industry observers to predict the final arrival of the true "consumer" computer (see the review on this machine which appeared in the November-December 1980 issue of **COMPUTE!**). But almost before the TRS-80 Color Computer (which I will refer to as the TRS-80 from now on) was in full production, Commodore announced the collapse of yet another barrier with the introduction of the VIC-20, priced at only $299.

For those of you who have yet to see a photograph of the VIC, it is about the smallest size a computer could be and still have a full-sized keyboard. This compact size makes the VIC fit easily into almost any imaginable home location — an important feature which other manufacturers have yet to understand.

Before going into details, I want to mention a little about the "ambience" of the VIC. Those of you who are familiar with the PET will find many of the good PET features on the VIC. Running a program from tape, for example, requires merely pressing SHIFT RUN and the cassette PLAY button. The full screen editor (using cursor control keys) is supported by the VIC, as is the PET graphics character set.

In a move which is certain to guarantee much support from the indigenous PET software community, Commodore even kept the tape formats identical so that PET programs could be loaded directly into the VIC. Most of the PET programs I have run on the VIC required only a few lines of revision to work perfectly. This suggests that outside software support for the VIC will appear instantaneously upon its arrival in the marketplace.

As a sign of Commodore's attention to detail in this area, both the user port and the cassette connector are identical to their PET counterparts, so many plug-in peripherals for the PET will plug into the VIC as well.

An IEEE-488 interface (standard with the PET) is available as an add-on for the VIC. Except for this interface, the PET and VIC interface environments are quite similar.

That the VIC-20 is an astounding machine for

**...it is about the smallest size a computer could be and still have a full-sized keyboard.**

the price is unquestioned. What we will try to do in this review is describe the VIC's capabilities and features in comparison with the machines with which it is likely to share the limelight — the TRS-80 (Color Computer) and the Atari 400.

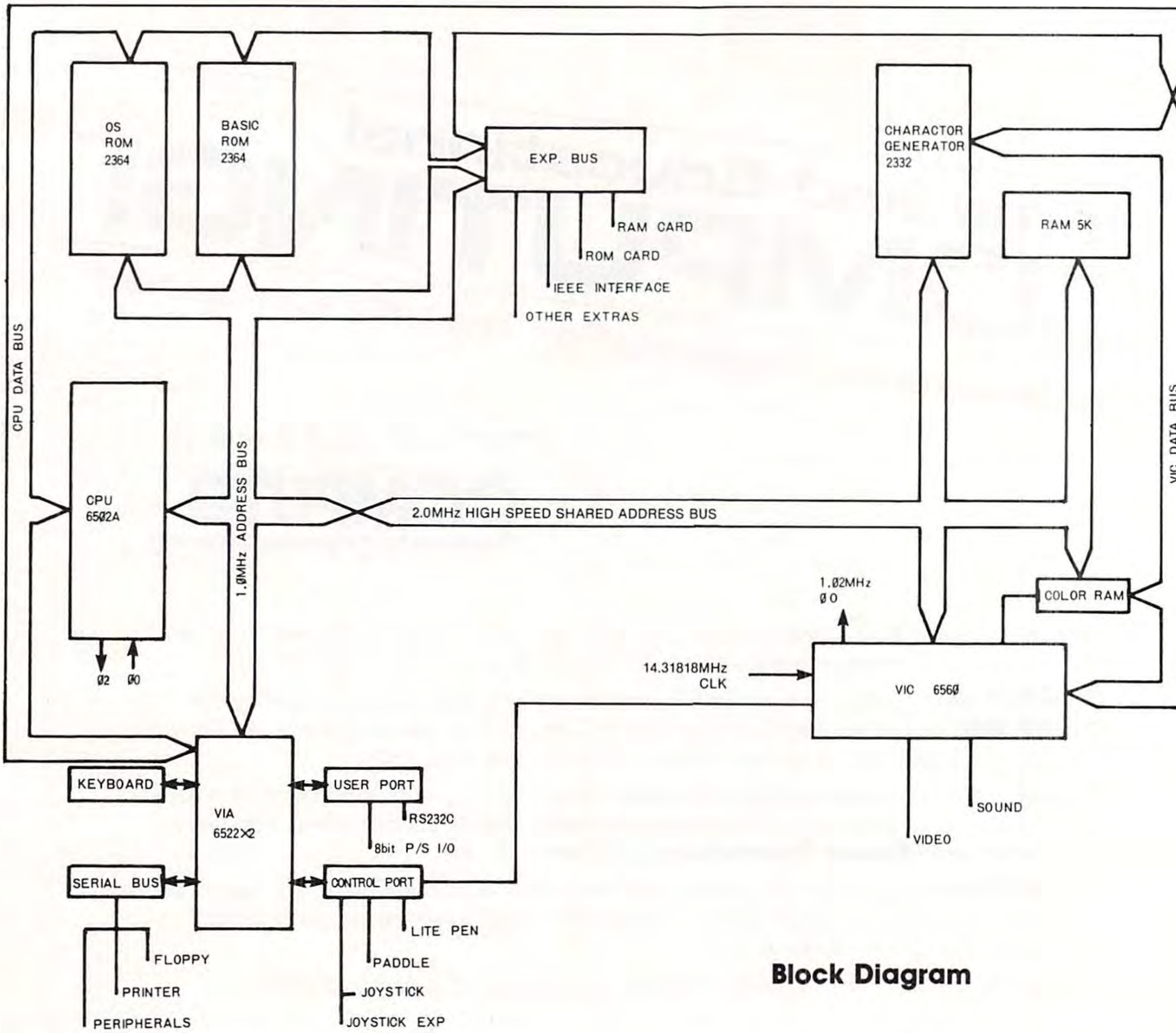
In order to provide some structure to this review, I have prepared a table which shows the salient features of each machine. This table is divided into four categories: OUTPUT, INPUT, EXTERNALS, and INTERNALS. We will discuss each of these in turn.

### OUTPUT...

Communication from the computer to the user requires (for all three machines) a television set. All three computers support color and generate user programmable sounds which are heard through the TV loudspeaker. All three computers come with RF modulators, thus making the connection to the home television a moderately trouble-free task.

These similarities between machines should in no way be taken to imply that there are no substantiative differences between these computers, however; the differences are *most* important.

In the area of alphanumeric display, for example, the VIC displays a maximum of 23 lines of 22 characters, compared to 16 lines of 32 characters for the TRS-80, and 24 lines of 40 characters for the Atari 400. In terms of character display quality, I rank the machines in the order: Atari, VIC, TRS-80; with the VIC and Atari both having a very high quality display. It should be noted that the




**Block Diagram**

Atari has more display modes than either of the other computers, and has the ability to display 24 lines of 20 characters, and 12 lines of 20 characters as well as its default (24 x 40) arrangement.

Both the VIC and Atari 400 display upper and lower case characters, and have an alphamosaic (graphics character) display feature also. The TRS-80 does not.

True "bit map" graphics modes are available on all three machines, with each machine's format being unique. The *maximum* display resolution for the VIC or 176 x 176 pixels (picture elements) compares favorably with the TRS-80 limit of 256 x 192 and the Atari maximum resolution of 320 x 192 pixels.

In terms of color control, the Atari is way out in front, since it has independent control over hue and luminance to achieve 128 colors. For most normal programs, however, the Atari user can only work with any four of these 128 colors on the screen at a time, compared with eight fixed colors for the TRS-80 and 16 fixed colors for the VIC.

Just as color is important for many applications, the creative use of sound can do much to enhance one's programs. The VIC supports three musical tone generators (3 octaves each) and one "sound effects" generator. The TRS-80, by comparison, only supports one musical sound channel. The Atari 400 has four sound generators, each of which is capable of musical sounds (4 octaves), or a wide variety of user programable sound effects.

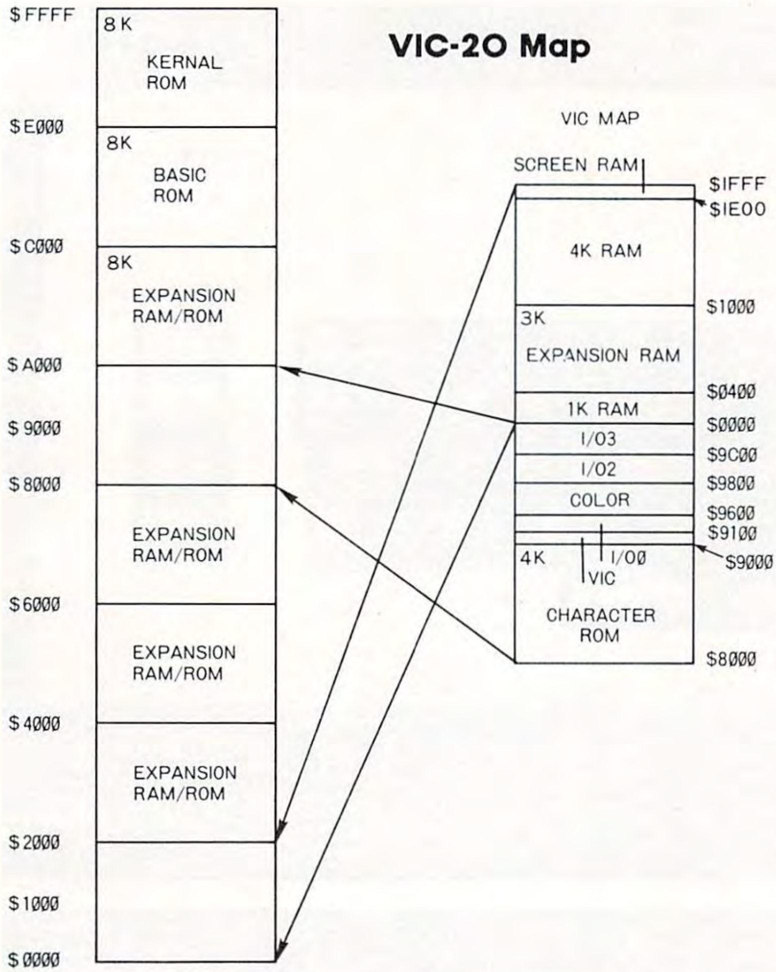
## Input...

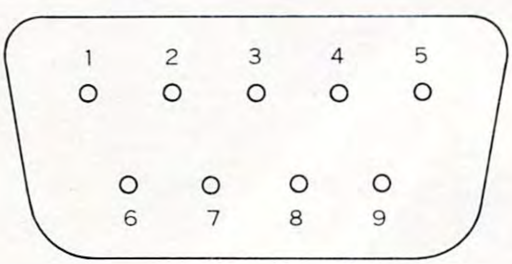
Those of you who appreciate nice keyboards will love the VIC-20. It is supplied with a 66 key arrangement with full typewriter-like key travel. Physically, the VIC keyboard resembles that on the Atari 800. The


commodore
VIC 20
COMPUTER
POWER
CLR HOME
INST DEL
CTRL
RESTORE
RUN STOP
SHIFT LOCK
RETURN
SHIFT
SHIFT
CRSR
CRSR

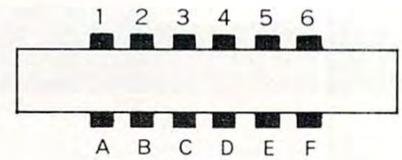

commodore
VIC 20
CONTROL PORT
OFF
ON
9V
AC

## VIC-20 Map

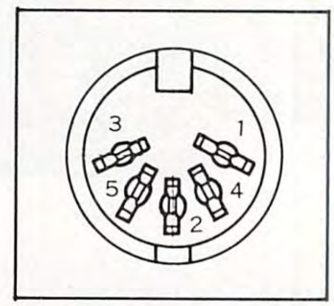


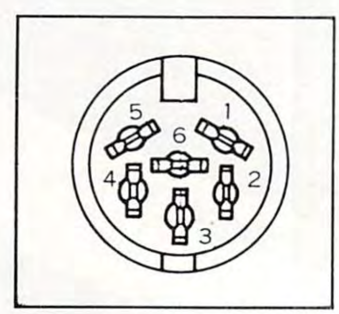

| | | |
|---|---|---|
| 1 | JOYØ | |
| 2 | JOY1 | |
| 3 | JOY2 | |
| 4 | JOY3 | |
| 5 | POT Y | |
| 6 | LIGHT PEN | |
| 7 | +5V | MAX. 1ØØmA |
| 8 | GND | |
| 9 | POT X | |




| | |
|---|---|
| A-1 | GND |
| B-2 | +5V |
| C-3 | CASSETTE MOTOR |
| D-4 | CASSETTE READ |
| E-5 | CASSETTE WRITE |
| F-6 | CASSETTE SWITCH |




| | | |
|---|---|---|
| 1 | +6V | 1ØmA MAX |
| 2 | GND | |
| 3 | AUDIO | |
| 4 | VIDEO LOW | |
| 5 | VIDEO HIGH | |




| | |
|---|---|
| 1 | SERIAL SRQ IN |
| 2 | GND |
| 3 | SERIAL ATN IN/OUT |
| 4 | SERIAL CLK IN/OUT |
| 5 | SERIAL DATA IN/OUT |
| 6 | NC |

TRS-80, on the other hand, has a medium travel ''clicky'' keyboard, and the Atari 400 (which places third in this comparison) has a membrane ''micromotion'' keyboard.

In terms of overall keyboard layout, I find that the keys on the Atari keyboard are slightly easier to find than those on the VIC, but presumably this is a result of my much longer experience with the Atari key arrangement.

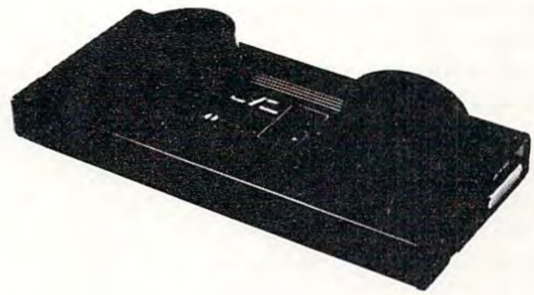
### Externals...

All three computers come with connections to support external cassette tape units for program and data storage. Unlike the Atari and Radio Shack pro-

# Skyles Electric Works

## "If you could own only one peripheral for your PET, it should be this. It opens the whole world to your PET."

**The Cat:** Switches for mode selection and operation. LEDs display unit status. Acoustic self-test is standard. Compact powerpack plugs directly into wall socket.



**Now $325.00*** complete with membership in The Source*, Skyles six-foot cable/interface to the PET user port, together with Skyles cassette program in machine language and in BASIC.
*(If bought separately: $180.00, modem; $80.00, cable/interface and program; $100.00, membership in The Source*)*

The Cat and D-Cat have been specially prepared by Skyles for interfacing to the PET user port (*not* to the IEEE port) and with a special cassette program, allowing communication

...from modem to disk and disk to modem
...from modem to terminal; read on screen, save on disk
...from disk to printer through IEEE
...from disk to screen

**The D-Cat:** FCC-approved for handset jack connection with any modular phone, either single or multi-line. No need for adapters. Can fit under phone; installs in seconds. Mode switch to monitor voice or data transmission. Special "hold" function; complete self-test. Power pack plugs directly into wall socket.

**Now $350.00*** complete with membership in The Source*, Skyles six-foot cable/interface to the user port and Skyles cassette program in machine language and in BASIC.

*(If bought separately: $210.00, the modem; $80.00, cable/interface and program; $100.00, membership in The Source*)*

## *What About the Source?

*Sometimes called "The Information Utility," it's a telecomputing network that gives you, through your Cat or D-Cat modem, thousands of programs and data bases and allows you to communicate with other users interactively and through electronic mail.*

**California residents: please add 6% or 6.5% sales tax as required*

VISA, MASTERCHARGE ORDERS CALL (800) 538-3083 (except California residents)
CALIFORNIA ORDERS PLEASE CALL (408) 257-9140

**Skyles Electric Works** | 231 E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

ducts, the VIC provides its own power to the recorder, thus reducing the number of power connections needed to get everything running. This is an important consideration when one realizes that most power outlets in the house have two sockets on them. With the VIC connected to a television set, both power outlets are used. With the Atari and Radio Shack computers, a third outlet is needed to connect the tape recorder.

Floppy disk drives are plannned for the VIC and TRS-80, and are available for the Atari 400. All three machines support printers, and can be connected (through an external modem and coupler) to the telephone line. This latter feature will be a crucial test for *all* modern computers as the growth of information utilities continues.

Additional input devices are also supported by the three computers. The VIC supports one (Atari-like) joystick, while the TRS-80 accepts two joysticks, and the Atari 400 accepts up to four. Rotary paddles are also supported on all three machines. The VIC can handle one, the TRS-80 can accept four, and the Atari 400 can accept up to eight game paddles. Both the VIC and the Atari 400 work with a light pen, but the TRS-80 does not.

As you can see, the VIC has many features which are not available on the TRS-80.



### Internals...

The VIC uses the 6502 microprocessor, but has one important architectural difference from the PET. In keeping with the designs of the TRS-80 and Atari 400, the VIC uses a special display controller chip (after which the VIC was named). The use of dedicated graphics chips is most beneficial to the end user, since it takes some of the load off the processor and makes the computer capable of feats which would otherwise be quite hard to perform in an eight-bit machine.

The entry level RAM of 5 Kbytes (of which 3583 bytes are available for program space) places VIC above the TRS-80 (4 Kbytes of RAM) and under the Atari 400 (8 Kbytes of RAM). In what appears to be a unique packaging idea, the VIC has a single external cartridge slot which accepts combinations of RAM and ROM. A 3 Kbyte RAM expander will be available from Commodore soon. This plug-in cartridge also has sockets for up to 24 Kbytes of ROM.

The VIC BASIC follows in the Microsoft tradition, and has the same "feel" as the PET BASIC, while adding capabilities associated with the color and sound features of the VIC.

To get some idea for the execution speed of VIC BASIC language, I ran the following program on the VIC, the TRS-80 Color Computer and on the Atari 400.

```
5 FOR J = 1 TO 100
10 FOR I = 1 TO 100
20 A = (I*I)/I
40 NEXT I
50 NEXT J
```

The execution times (in seconds) are shown below for each computer.

| | |
|---|---|
| VIC-20 | 77 seconds |
| TRS-80 Color Computer | 103 seconds |
| Atari 400 | 159 seconds |

As can be seen from these figures, the VIC has the clear lead. Next I added one line to this program:

```
30 PRINT A
```

and ran the experiment again with the following results:

| | |
|---|---|
| VIC-20 | 347 seconds |
| TRS-80 Color Computer | 280 seconds |
| Atari 400 | 540 seconds |

This time the Radio Schack entry is the leader of the BASIC race.

### Additional Comments...

From the comparisons shown above, it is pretty clear that the VIC will provide very stiff competition to the TRS-80 Color Computer. To help cement the VIC in the marketplace, Commodore has announced a plan by which they will help cottage industry software developers in their development of cartridge-based firmware. It is as though Commodore genuinely realizes the positive impact of outside software

**Table I**

| | VIC | TRS-80 | Atari 400 |
|---|---|---|---|
| Price: | $299 | $399 | $499 |
| **OUTPUT** | | | |
| Display Incl.: | NO | NO | NO |
| Color Display: | YES | YES | YES |
| Characters: | 22x23 | 32x16 | 40x24 |
| Upper/Lower: | YES | NO | YES |
| Alphamosaic: | YES | NO | YES |
| Bit Map: | 176x176 | 256x192 | 320x192 |
| Sound Gen.: | 4 | 1 | 4 |
| **INPUT** | | | |
| # Keys: | 66 | 53 | 57 |
| Kbd. Type: | Full Stroke | Med. Stroke | Flat Panel |
| **EXTERNALS** | | | |
| Tape: | YES | YES | YES |
| Disk: | SOON | SOON | YES |
| Printer: | YES | YES | YES |
| Joysticks: | 1 | 2 | 4 |
| Light Pen: | YES | NO | YES |
| Paddles: | 1 | 4 | 8 |
| Modem: | YES | YES | YES |
| **INTERNALS** | | | |
| Processor: | 6502 | 6809 | 6502 |
| Display Chip: | YES | YES | YES |
| RAM: | 5K | 4K | 8K |
| ROM: | ? | 8K | 18K |
| MAX RAM: | see text | 16K | 16K |
| MAX ROM: | see text | 16K | 26K |
| Plug-in ROM: | YES | YES | YES |

on hardware sales — a fact which Texas Instruments, for example, has only begun to appreciate.

I expect that the VIC will be quite popular with children, and that it will thus find its way into primary grade classrooms as well as into homes. Personally, I find the 22 character display to be too small to support any but the most rudimentary business applications — perhaps including home finance applications. But at a price of $299, that is hardly the point. The VIC is a much more valuable computer literacy tool than either the hand held Sharp computer being sold by Radio Shack, or any of the similarly priced single board computers which have been on the market for several years. It comes with a very fast, high quality BASIC, and with some excellent graphic and sound capabilities.

VIC will create its own market, and it will be a big one. ©

*Editor's Note: Two vendors have recently announced memory expansion modules for the Atari 400. These modules bring the max RAM capacity of the 400 to 32K. Both the VIC-20 and the TRS-80 Color Computer have built-in RS-232C serial ports; the Atari 400 does not.*

## Advice to PET Owners:

# How To Be A VIC Expert

Jim Butterfield, Toronto

There are going to be a lot of VIC computers arriving very soon. All those new VIC owners are going to look to you for advice and counsel. After all, you've owned a PET for several months; and a VIC is just a junior version of a PET, right?

You don't want to blow your chance to become the block's VIC guru. It really is very much the same as the PET, but there are new things you will need to get used to.

I'll give a few hints here: Basic is the same as on the PET, right down to the LOG, SQR, and ATN functions. Because the VIC has fewer columns, you can link up to four rows together to generate a line of Basic. Many of the advanced VIC features, like color, are done the same way as programmed cursor on the PET. You clear the PET screen by printing a special reverse-character; you'll set color on the VIC exactly the same way.

The old familiar PEEK the POKE locations have moved around; when you find them, they will work the same as in the PET...but that's a whole other story.

A good way to start is with a sample program. Here's a VIC program which you can type in on your PET, and save on tape. It won't work properly on the PET, but your tape will load in nicely on the VIC, and you'll get an insight into how some of the VIC things work. By the way, your cassette unit itself will plug into the VIC, so carry both tape and cassette unit over to the new machine.

```
100 remark: big letter display
110 rem     by jim butterfield
120 rem   peeks the VIC character generator
130 rem   in hex 8000 to 8FFF
140 input "graphic/text";g$
150 g=asc(g$)
160 if g=71 then b=0:goto 190
170 if g=84 then b=2048:goto 190
180 stop
190 input"character #";n
200 if n<0 or n>255 or n<>int(n) goto 140
210 m=32768+b+8*n
220 print chr$(176);:for j=1 to 8:print chr$(192);:
next j:print chr$(174)
230 for j=m to m+7
240 print chr$(221);
250 x=peek(j)
260 for k=1 to 8
270 y=146:x=x*2:if x>255then x=x-256:y=18
280 print chr$(y);chr$(32);
290 next k
300 print chr$(146);chr$(221)
310 next j
320 print chr$(173);:for j=1 to 8:print chr$(192);:
next j:print chr$(189)
330 goto 140
```

What does the above program do? It prints out the 256 characters used by VIC in large size. The user picks a character (from 0 to 255) and it is displayed on VIC's screen. The characters are screen format, not ASCII, so that a value of 1 gives an A character. You'll find that the characters are similar to those used by the PET.

How does this program behave in the VIC? You'll find a few differences that will cause you to change your programming style when you shift to the new machine. The first thing you'll notice is that you'll have trouble stopping the program. The INPUT statement on line 140 does not stop the machine if you press RETURN with no input. It continues running, and leaves variable G$ at its previous value. That's different: it means that you can set up a default for G$ and the user can invoke it by just pressing RETURN. It also means that you have to find another way of stopping the program. The trick here is to input a character such as X when asked, "GRAPHIC/TEXT". The program will continue only on a response of G or T - or no response, as noted before.

You'll have noticed that the program is very PET-compatible. In fact, it will run on the PET with two small differences. First, the PET can't read its own character generator, so you'll get nonsense displayed. Secondly, the PET behaves a little differently on the INPUT statement as we have noted.

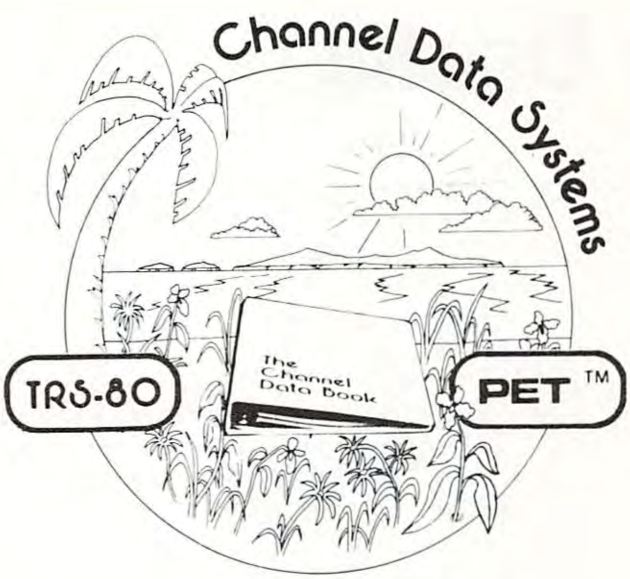
Here's a puzzler: when you punched up your program on the PET, it occupied memory space starting at decimal 1025. On the VIC, the program will want to take up residence starting at decimal 4096. How can your PET program load properly to the VIC? Easy: VIC has a relocating loader; it just moves the program to the new place. Transferring programs the other way — from the VIC to the PET — isn't as easy, since the PET does not relocate programs.

A final note on the coding. There's a lot of use of CHR$ characters: why didn't I use the more familiar characters in quotes, which would certainly work? Answer: it would drive the staff of **COMPUTE!** wild, since they wouldn't be able to typeset all those fancy characters. Then they would substitute their own symbols, with a translation legend somewhere near, and you'd be driven wild in turn trying to type it in. Trust me: it's better this way. As an exercise, you can work out how to recode most of the CHR$ expressions into screen characters.

Thought for the day: if the character generator is accessible in memory, do you think that you might be allowed to code your own set of characters in RAM memory? The answer, of course, is yes; but you'll have to encode the whole character set you need since all characters must be grouped together. But that's another story...

You've generated your first VIC program. Hopefully, you've discovered a few things about how the VIC works. Much of it will be the same as with the PET, but a few features are different.

Now, when all of the new VIC owners on the block beat a path to your door, practice saying wise things like, "Of course, on the big machines, we do it this way..." ©




**OMNIFILE** CBM or TRS-80 $30

**Omnifile** is a versatile, in-memory database program with sorting, formatting, and computational features. Records can be entered, edited, and processed with a single letter command. **Omnifile** applications include inventory records, mailing lists, sales journals and collection lists. Records can be stored on the Commodre floppy disks or on the tape cassette. **Omnifile** uses approximately 6k of RAM memory. Up to 500 records can be contained in memory in a 32k CBM at any time. Multiple files are easily accessed from disk or tape. Items can be sorted, moved, inserted and reformatted. Calculations can be made and totals can be printed. The **Omnifile** package includes the program with sample data, listing and manual, and will operate on the large keyboard Commodore PET or CBM computers with at least 16k memory. Also available on diskette for $36. An abbreviated version, **Data Logger,** requiring only 1k of RAM is available on cassette for $15.

**GENERAL LEDGER/PERSONAL LEDGER** CBM or TRS-80 $30

**General Ledger** is a complete double entry bookkeeping system with provisions for budgeting and keeping records of income, deductible and non-deductible expenses, assets and liabilities. Simple interactive features allow entering transactions, adding or editing accounts, and printing of a detailed income statement and balance sheets. Data can be stored on the Commodore floppy disks or cassette. **General Ledger** occupies about 6.2k of RAM memory, allowing approximately 200 accounts on a 16k machine. Transaction files can be accessed by our **Omnifile** database program for complete analysis, sorting by date, account number, etc. The **General Ledger** program will operate on the new Commodore PET or CBM microcomputer systems and comes with sample data, listing, and manual. Also available on diskette for $36. An abbreviated version allowing about 35 accounts on a 1.0 or 2.0 BASIC 8k PET is available on cassette for $20.

**EXPLORE** CBM $15

Inspired by the computerized fantasy simulation "Adventure," **Explore** is a conversational program which operates on the Commodore PET with only 8k bytes of memory. **Explore** contains four adventures in which you operate a computerized tank, hunt treasure in a magic cave, explore the mall in Washington D.C., and survive in a haunted castle. **Explore** package includes introduction, five data files, and complete manual. Available from Channel Data Systems on cassette for $15. Indication of old or new ROMs is requested.

**CHANNEL DATA BOOK for PET** CBM $20

A complete hardware and software reference service listing descriptions for over 1400 software programs and over 200 peripheral devices for PET. Also includes an information sources section, and addresses for Commodore Dealers in the USA and Canada, Commodore Vendors, and PET user groups. Designed to organize documentation, newsletters, listings, and other user selected information in an attractive 3-ring binder.

**ACCESSORIES**

**PORT NOISE COMPLETE** Speaker-amplifier for Commodore PET or CBM with connector (M & N pin) all ready to plug in—$20.

**IEEE/USER PORT CONNECTOR** 24 pin connector. With backshell—$7. Without backshell—$4.

**C-10 TAPE CASSETTES** High quality AGFA tape with screw type case. $10/10 or $75/100.

Cash, Checks, Money Orders, VISA, Mastercharge
Add $2 shipping for each order—COD-$3.50
For foreign orders please inquire for pricing

CHANNEL DATA SYSTEMS
4141 State Street Santa Barbara, CA 93110 805-964-6695

# Basically Useful BASIC

## Ascending/Descending Sort

Rick Keck
Overland Park, KS

At some point in time, every computer system user needs the services of a sort program. There has been much work done in the field of "sorting theory" and from this there has come a variety of different sorting methods. Some of these include the bubble sort, shell sort, binary sort, and tag sort. The benefit of this exists in the ability to select a method which is best for sorting data based upon the data's unique characteristics (if any). The factors which influence the decision of which sorting method to use include the following data characteristics: volume, relative order, and storage form (random access or sequential files). In a majority of cases, a simple sorting method will work fine. The standard order of sorting data is to have it sorted from smallest to biggest (ascending order). On occasion, sorting of data may need to be done from biggest to smallest (descending order). The following modified bubble sort routine allows the data to be sorted in either ascending or descending order. Note that the data is handled by character string variables so as to allow alphanumeric data to be sorted.

```
100 REM****************************
110 REM*                          *
120 REM* ASCENDING / DESCENDING *
130 REM*                          *
140 REM*       SORT  ROUTINE      *
150 REM*                          *
160 REM*  BY  RICK KECK  01/81   *
170 REM****************************
180 DIM C$(100)
190 DATA "JOHN","BILL","MARY"
200 DATA "CAROLINE","FRED","SUE"
210 DATA "JOE"
220 REM* N HOLDS # OF DATA
230 N=7
240 REM* READ DATA INTO C$
250 FOR J=1 TO N
260 READ C$(J)
270 NEXT J
280 REM* ASCENDING OR DESCENDING
290 PRINT:PRINT:PRINT
300 PRINT:PRINT"WHAT ORDER DO YOU WISH TO SORT:"
310 PRINT:PRINT"      A - ASCENDING (SMALL TO BIG)"
320 PRINT:PRINT"      D - DESCENDING (BIG TO SMALL)"
330 PRINT:PRINT
340 INPUT A$
350 IF A$<>"A" AND A$<>"D" THEN 340
360 PRINT:PRINT
370 REM****************************
380 REM*       SORT  BEGINS       *
390 REM****************************
400 FOR K=1 TO (N-1)
410 IF A$="A" THEN 440
420 IF C$(K)>=C$(K+1) THEN 540
430 IF A$="D" THEN 450
440 IF C$(K)<=C$(K+1) THEN 540
450 FOR J=K TO 1 STEP -1
460 IF A$="A" THEN 490
470 IF C$(J)>=C$(J+1) THEN 540
480 IF A$="D" THEN 500
490 IF C$(J)<=C$(J+1) THEN 540
500 T$=C$(J)
510 C$(J)=C$(J+1)
520 C$(J+1)=T$
530 NEXT J
540 NEXT K
550 REM****************************
560 REM*         SORT  ENDS       *
570 REM****************************
580 FOR L=1 TO N
590 PRINT C$(L)
600 NEXT L
610 PRINT:PRINT"NORMAL TERMINATION"
620 END
```

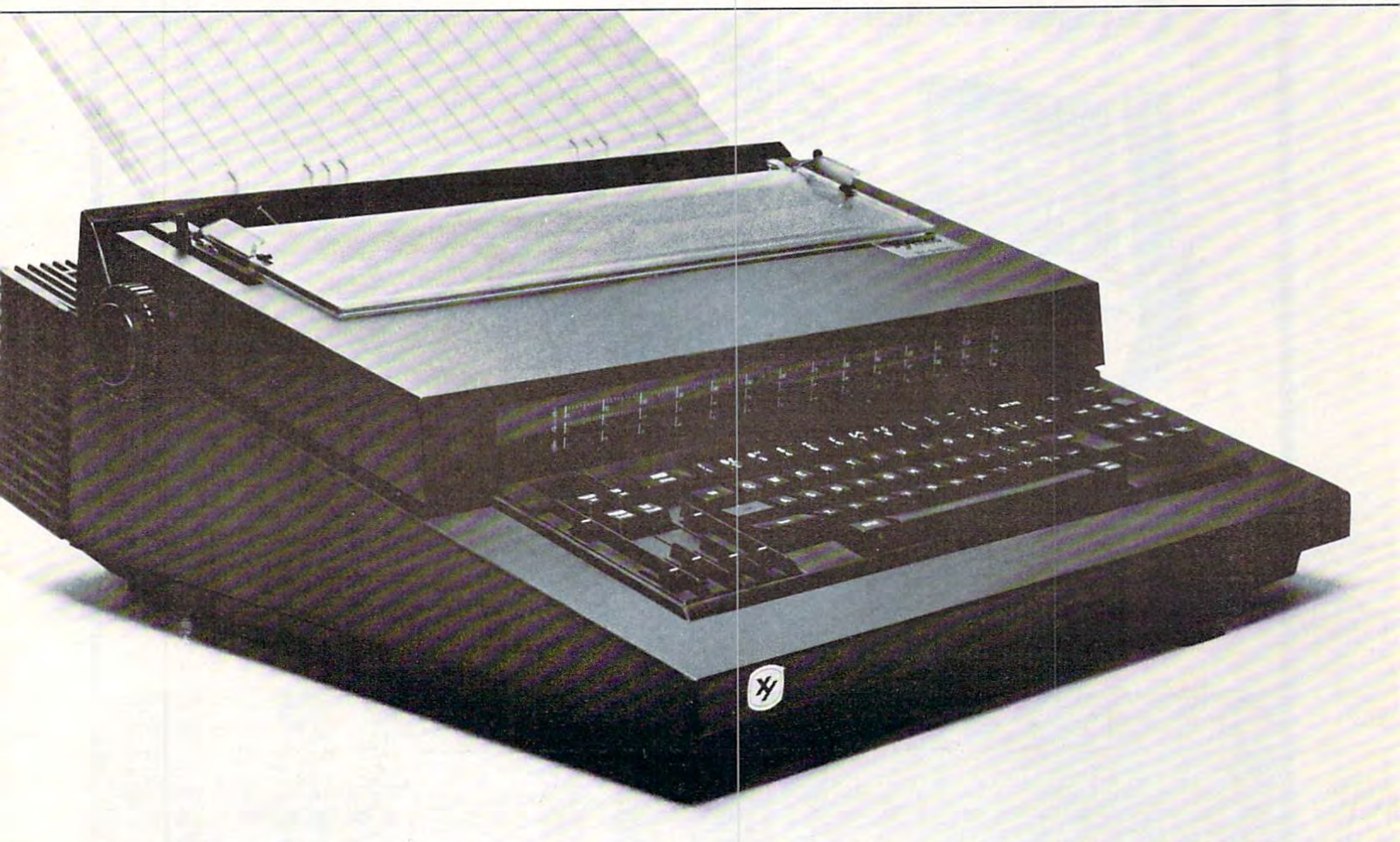
```
BILL
CAROLINE
FRED
JOE
JOHN
MARY
SUE
```

```
SUE
MARY
JOHN
JOE
FRED
CAROLINE
BILL
```

©

# HY-Q 1000

## HY-Q 1000 series: Letter quality, intelligent printers



**Features:**

- Optional RS232-C, IEEE-488 (GPIB) or Centronics 779 compatible interfaces; tractor feed and LED display
- Centers titles, sets columns and right justifies
- Prints in 10, 12 and 15 pitch
- Proportional spacing — with 1/60 inch resolution
- Prints boldface and reverse — for highlighting important information, as in management reports
- Multiple print wheels — changing type styles takes but seconds
- Added versatility — operates as electronic typewriter when not in use as a printer

Xymec's HY-Q 1000™ brings microprocessor efficiency to printing. Now you can expand the capability of your computer or word processing system. Xymec's intelligent printer/typewriter is compatible with most micro/minicomputers, including Commodore, Apple, TRS-80 and Atari. Reliable and easy to service, Xymec intelligent printers won't create a service nightmare. With service centers nationwide, maintenance is easy and convenient. Give your computer system the intelligent edge. Join up with Xymec today. For more information: 17905 Sky Park Circle, Suite J, Irvine, CA 92714. (714) 557-8501. TWX: 910-595-2443.

Xymec™

The Value Standard in Printers

# NEECO

"Your complete source for all CBM Hardware and Software Products"

# PROUDLY ANNOUNCES OUR NEW ONE YEAR WARRANTY ON ALL CBM COMPUTERS!

The 8032 CBM Computer is now available!

commodore


NEW!

operating system and new disk operating system with direct access!


## CBM™ 8000 SERIES BUSINESS COMPUTERS

The new Commodore 8000 series computers offer a wide screen display to show you up to 80-character lines of information. Text editing and report formatting are faster and easier with the new wide-screen display. The 8000 series also provides a resident Operating System with expanded functional capabilities. You can use BASIC on the 8000 computers in both interactive and program modes, with expanded commands and functions for arithmetic, editing, and disk file management. The CBM 8000 series computers are ideally suited for the computing needs of the business marketplace.

## CBM™ 8050 DUAL DRIVE FLOPPY DISK

The CBM 8050 Dual Drive Floppy Disk is an enhanced version of the intelligent CBM 2040 Disk Drive. The CBM 8050 has all of the features of the CBM 2040, and provides more powerful software capabilities, as well as nearly one megabyte of online storage capacity. The CBM 8050 supplies relative record files and automatic diskette initialization. It can copy all the files from one diskette to another without copying unused space. The CBM 8050 also offers improved error recovery and the ability to append to sequential files.

**HARDWARE SPECIFICATIONS**

Dual Drives
Two microprocessors
974K Bytes storage on two 5.25" diskettes (single sided)
Tracks 70
Sectors 17-21
Soft sector format
IEEE-488 interface
Combination power (green) and error (red) indicator lights
Drive Activity indicator lights
Disk Operating System Firmware (12K ROM)
Disk Buffer (4K RAM)

**FIRMWARE**

DOS version 2.1
Sequential file manipulation
Sequential user files
Relative record files
Append to sequential files
Improved error recovery
Automatic diskette initialization
Automatic directory search
Command parser for syntax validation
Program load and save

| CBM | PRODUCT DESCRIPTION | PRICE |
|---|---|---|
| 4008N | 8K RAM-Graphics Keyboard-40 col | $ 795.00 |
| 4016N | 16KN RAM-Graphics Keyboard-40 col | $ 995.00 |
| 4016B | 16K RAM-Business Keyboard-40 col | $ 995.00 |
| 4032N | 32K RAM-Graphics Keyboard-40 col | $1295.00 |
| 4032B | 32K RAM-Business Keyboard-40 col | $1295.00 |
| 8016 | 16K RAM-80 Col.-4.1 O/S | $1495.00 * |
| 8032 | 32K RAM-80 Col.-4.1 O/S | $1795.00 |
| 2023 | Friction Feed Printer | $ 695.00 |
| 2022 | Tractor Feed Printer | $ 795.00 |

**NOTE: All current CBM production computers/disks now contain operating system 4.1/DOS 2.1**

| CBM | PRODUCT DESCRIPTION | PRICE |
|---|---|---|
| 2040 | Dual Floppy-343K-DOS 1.0 | $1295.00 |
| 4040 | Dual Floppy-343K-DOS 2.0 | $1295.00 |
| 8050 | Dual Floppy-974K-DOS 2.0 | $1695.00 * |
| C2N Cassette | External Cassette Drive | $ 95.00 |
| CBM to IEEE | CBM to 1st IEEE Peripheral | $ 39.95 |
| IEEE to IEEE | CBM to 2nd IEEE Peripheral | $ 49.95 |
| 8010 | IEEE 300 Baud Modem | $ 395.00 * |
| 2.0 DOS | DOS Upgrade for 2040 | $ 50.00 |
| 4.0 O/S | O/S Upgrade for 40 Column | $ 100.00 |

*Asterisks indicate fall delivery—all others are immediately available

## SPECIAL OFFER ON CBM COMPATIBLE BUSINESS SOFTWARE!

**Purchasing software has always been difficult due to the "you buy it - you own it" attitude of most vendors. We at NEECO, recognize this problem and can now, on all of the Software Packages listed, offer a full 30 day refund policy to NEECO's customers. Now you can purchase with confidence. Buy it - try it; if the program package is not suitable for any reason, send it back to us within 30 days and we will refund the full purchase price—less shipping charges!**

NEW!

| SOFTWARE | APPLICATION | REQUIRES | AUTHOR | AVAILABILITY | PRICE |
|---|---|---|---|---|---|
| Word Pro I | Word Processing | 8K + cassette | Professional Software | Immediate | $ 29.95 |
| Word Pro II | " | 10K + 2040 | " | " | 99.95 |
| Word Pro III Plus | " | 32K + 2040 | " | " | 395.00 |
| Word Pro IV Plus | " | 8032 + 2040/8050 | " | " | 595.00 |
| BPI Integrated G/L | Business | 32K/8032 + 2040 | BPI | " | 360.00 |
| BPI Inventory | " | " | " | " | T.B.A. |
| BPI Payroll | " | " | " | " | |
| BPI Enhanced A/R | " | " | " | " | |
| CMS G/L | " | " | CMS Software | " | 295.00 |
| CMS A/R | " | " | " | " | 195.00 |
| CMS A/P | " | " | " | " | 195.00 |
| CMS Customer Mail List | " | " | " | " | 195.00 |
| CMS Payroll | " | " | " | " | 350.00 |
| Datasource 1 | All Business | 32K/8032 + 2050/8050 | BMB | August/Sept. | 295.00 |

*Wordprocessing Software requires output printer. We recommend the NEC Spinwriter ($2995) for letter quality.
*PET is a registered trademark of Commodore Business Machines. Small Keyboard PETS require a ROM Retrofit Kit.
Multi-Cluster is available in Canada from BMB Compu Science, P.O. BOX 121, Milton, Ontario, L9T2Y3
All prices and specifications are subject to change without notice.

**NEECO**
679 HIGHLAND AVE.
NEEDHAM, MA 02194

**NEW ENGLAND ELECTRONICS CO., INC.**
"NEW ENGLAND's Largest Computer Showroom"

**(617) 449-1760**
MASTERCHARGE OR VISA ACCEPTED
TELEX NUMBER 951021, NEECO
MON-FRI, 9:00-5:30

# Program Compactor

Edward H. Carlson
Okemos MI

There are two evils that sneak up on you as your programs attain moderate length. The programs begin to take up too much space in memory, and they become increasingly obtuse. These evils combine in positive feedback. Increased internal documentation by **REM**ark statements is a partial antidote to program complexity but this, of course, compounds the problem of fitting the whole glob into memory.

The answer is to have two copies of each program: a "working copy" occupying minimum space, and a fully documented "archives copy" that may,

**The first 2 bytes of a BASIC line are a pointer to the start of the next BASIC line.**

in fact, be too long to be run in your machine. (It must be short enough to fit in your memory as source code, sans variable tables, of course.)

But consider, you say, how much finger tapping, eyeball twitching, and obsessive concentration it takes to go through a program and remove all the REMarks, especially those buried inside lines of active code, and most especially those "invisible" REM statements like this one:

```
120 GOTO 232:NO "REM" NEEDED HERE
```

(In such a statement, the BASIC interpreter jumps to another line before passing the colon and so does not detect the syntax error caused by the omitted REM.)

Looky here, I say, repetitive decisions are just what the logic machine was invented to perform! One needs to write a "program compacting" program, and that is just what I have done, showing my results in Listing 1. The program was written for use with my Ohio Scientific C2-4P, but should work with little change in other Microsoft BASIC machines, such as the PET. All that needs changing is the starting address of the source code, \$0300 for OSI BASIC, and the numerical values of the tokens, which differ in the OSI and PET versions of Microsoft BASIC.

This compactor is a moderately complex program in itself. It is put at high line numbers so as to be out of the way of any program you are writing. When you have a version of your own program that needs compacting, **first save it to tape,** then read in the Compactor (from a tape that does not have the Test Program in front). Do a "RUN 62000". The compacted program will then be POKEd into memory, ready to SAVE to tape or to RUN. The Compactor will still be in memory, but now invisible to you and inaccessible to BASIC.

Listing 1 starts off with a very short Test Program that has most of the features that would give trouble in a poorly contrived Compactor. Then follows the compactor itself. After initializing addresses, etc., a loop over I is started. Each time through, one line is compacted. Line 62036 contains the exit from the compacting process. This occurs when the line number to be processed is above 9999. You may wish to change this, but all my programs use only line numbers below 9000. Next, leading colons and spaces are removed. I haven't used such things in my own code, but it is legal and so I include that case in the Compactor Program.

Following these preliminaries, the program enters a loop over K, at line 62050, which walks through a single line. Spaces are removed, and the line is terminated if a REM, STOP, RETURN, or GOTO is encountered. The compacted line is stored in an array called L(I). This is an artifact left over from the program construction period. Before allowing my infant program to actually POKE into tender source code memory, I had it make a string and print it. Upon reaching voting age, the string became the L array.

During all this, it is necessary to keep a sharp eye out for quotation marks, as you do not want to alter any of the text inside them. Line 62080 detects opening quotes and jumps to a routine to march along looking for the closing quotes so that control can be returned to the main loop. If a colon or a null is found before the closing quotation marks, the statement or line has terminated and analysis of the next is begun.

Every Microsoft BASIC line ends in a null. Detection of a null character sends control to the top of the "I" loop. The next command sends control to the subroutine at 62600 where the compacted line is POKEd into memory. Some tricky address changing is needed here. The first 2 bytes of a BASIC line are a pointer to the start of the next BASIC line. This chain of pointers must remain intact during interpretation of any part of the Compactor that would do a line number search. Such a search would start at the first line of the program to be compacted, even though the code being interpreted is all above line number 62000. So lines 62601 and 2 pick up the starting address of the code that is next to be compacted in POKEs it into the first two bytes of the newly compacted line.

# DYNACOMP

*Quality software for⁺:*

| | |
|---|---|
| **ATARI** | **TRS-80 (Level II)*** |
| **PET** | **NORTH STAR** |
| **APPLE II Plus** | **CP/M 8" Disk** |

## GAMES, SIMULATIONS, EDUCATION and MISCELLANEOUS

**BRIDGE 2.0 (Available for all computers)** — **Price: $17.95 Cassette/$21.95 Diskette**
An all-inclusive version of this most popular of card games. This program both BIDS and PLAYS either contract or duplicate bridge. Depending on the contract, your computer opponents will either play the offense OR defense. If you bid too high, the computer will double your contract! BRIDGE 2.0 provides challenging entertainment for advanced players and is an excellent learning tool for the bridge novice.

**HEARTS 1.5 (Available for all computers)** — **Price: $14.95 Cassette/$18.95 Diskette**
An exciting and entertaining computer version of this popular card game. Hearts is a trick-oriented game in which the purpose is not to take any hearts or the queen of spades. Play against two computer opponents who are armed with hard-to-beat playing strategies.

**STUD POKER (ATARI only)** — **Price: $11.95 Cassette/$15.95 Diskette**
This is the classic gambler's card game. The computer deals the cards one at a time and you (and the computer) bet on what you see. The computer does not cheat and *usually* bets the odds. However, it sometimes bluffs! Also included is a five card draw poker betting practice program. This package will run on a 16K ATARI. Color, graphics, sound.

**POKER PARTY (Available for all computers)** — **Price: $17.95 Cassette/$21.95 Diskette**
POKER PARTY is a draw poker simulation based on the book, POKER, by Oswald Jacoby. This is the most comprehensive version available for microcomputers. The party consists of yourself and six other (computer) players. Each of these players (you will get to know them) has a different personality in the form of a varying propensity to bluff or fold under pressure. Practice with POKER PARTY before going to that expensive game tonight! Apple Cassette and diskette versions require a 32 K (or larger) Apple II.

**VALDEZ (Available for all computers)** — **Price: $14.95 Cassette/$18.95 Diskette**
A simulation of supertanker navigation in the Prince William Sound and Valdez Narrows. The program uses an extensive 256X256 element radar map and employs physical models of ship response and tidal patterns. Chart your own course through ship and iceberg traffic. Any standard terminal may be used for display.

**FLIGHT SIMULATOR (Available for all computers)** — **Price: $17.95 Cassette/$21.95 Diskette**
A realistic and extensive mathematical simulation of take-off, flight and landing. The program utilizes aerodynamic equations and the characteristics of a real airfoil. You can practice instrument approaches and navigation using radials and compass headings. The more advanced flyer can also perform loops, half-rolls and similar aerobatic maneuvers.

**CRIBBAGE 2.0 (TRS-80 only)** — **Price: $14.95 Cassette/$18.95 Diskette**
This is a well-designed and nicely executed two-handed version of the classic card game, cribbage. It is an excellent program for the cribbage player in search of a worthy opponent as well as the beginner wishing to learn the game, in particular the scoring and jargon. The standard cribbage score board is continually shown at the top of the display (utilizing the TRS-80's graphics capabilities), with the cards shown underneath. The computer automatically scores and also announces the points using the traditional phrases.

**CHESS MASTER (North Star and TRS-80 only)** — **Price: $19.95 Cassette/$23.95 Diskette**
This complete and very powerful program provides five levels of play. It includes castling, en passant captures and the promotion of pawns. Additionally, the board may be preset before the start of play, permitting the examination of "book" plays. To maximize execution speed, the program is written in assembly language (by SOFTWARE SPECIALISTS of California). Full graphics are employed in the TRS-80 version, and two widths of alphanumeric display are provided to accommodate North Star users.

**STARTREK 3.2 (Available for all computers)** — **Price: $ 9.95 Cassette/$13.95 Diskette**
This is the classic Startrek simulation, but with several new features. For example, the Klingons now shoot at the Enterprise without warning while also attacking starbases in other quadrants. The Klingons also attack with both light and heavy cruisers and move when shot at! The situation is hectic when the Enterprise is besieged by three heavy cruisers and a starbase S.O.S. is received! The Klingons get even!

**SPACE TILT (Apple only)** — **Price: $10.95 Cassette/$14.95 Diskette**
Use the game paddles to tilt the plane of the TV screen to "roll" a ball into a hole in the screen. Sound simple? Not when the hole gets smaller and smaller! A built-in timer allows you to measure your skill against others in this habit-forming action game.

**GAMES PACK I (Available for all computers)** — **Price: $9.95 Cassette/$13.95 Diskette**
GAMES PACK I contains the classic computer games of BLACKJACK, LUNAR LANDER, CRAPS, HORSERACE, SWITCH and more. These games have been combined into one large program for ease in loading. They are individually accessed by a convenient menu.

**GAMES PACK II (Available for all computers)** — **Price: $9.95 Cassette/$13.95 Diskette**
GAMES PACK II includes the games CRAZY EIGHTS, JOTTO, ACEY-DUCEY, LIFE, WUMPUS and others. As with GAMES PACK I, all the games are loaded as one program and are called from a menu.

Why pay $7.95 or more per program when you can buy a DYNACOMP collection for just $9.95?

**NOMINOES JIGSAW (ATARI and TRS-80 only)** — **Price: $16.95 Cassette/$20.95 Diskette**
NOMINOES JIGSAW is an intriguing and sophisticated graphical puzzle. The jigsaw consists of a 9 by 9 board partially filled with randomly chosen shapes (nominoes), of which there are 60 types. By knowing that the shapes must be legally connected, and by guessing the shape at each location, all the nominoes may be eventually deduced. Scoring is based on the number of guesses required and the difficulty of the board set-up.

**MOVING MAZE (Apple only)** — **Price: $10.95 Cassette/$14.95 Diskette**
MOVING MAZE employs the games paddles to direct a puck from one side of a maze to the other. However, the maze is dynamically (and randomly) built and is continually being modified. The objective is to cross the maze without touching (or being hit by) a wall. Scoring is by an elapsed time indicator, and three levels of play are provided.

**BLACK HOLE (Apple only)** — **Price: $14.95 Cassette/$18.95 Diskette**
This is an exciting graphical simulation of the problems involved in closely observing a black hole with a space probe. The object is to enter and maintain, for a prescribed time, an orbit close to a small black hole. This is to be achieved without coming so near the anomaly that the tidal stress destroys the probe. Control of the craft is realistically simulated using side jets for rotation and main thrusters for acceleration. This program employs Hi-Res graphics and is educational as well as challenging.

**TEACHER'S PET I (Available for all computers)** — **Price: $ 9.95 Cassette/$13.95 Diskette**
This is the first of DYNACOMP's educational packages. Primarily intended for pre-school to grade 3, TEACHER'S PET provides the young student with counting practice, letter-word recognition and three levels of math skill exercises.

**CRYSTALS (ATARI only)** — **Price: $ 9.95 Cassette/$13.95 Diskette**
A unique algorithm randomly produces fascinating graphics displays accompanied with tones which vary as the patterns are built. No two patterns are the same, and the combined effect of the sound and graphics are mesmerizing. CRYSTALS has been used in local stores to demonstrate the sound and color features of the Atari.

**CRANSTON MANOR ADVENTURE (North Star only)** — **Price: $19.95**
At last! A comprehensive Adventure game for the North Star. CRANSTON MANOR ADVENTURE takes you into mysterious CRANSTON MANOR where you attempt to gather fabulous treasures. Lurking in the manor are wild animals and robots who will not give up the treasures without a fight. The number of rooms is greater and the associated descriptions are much more elaborate than the current popular series of Adventure programs, making this game the top in its class. Play can be stopped at any time and the status stored on diskette. Requires 32K.

**NORTH STAR SOFTWARE EXCHANGE (NSSE) LIBRARY** — **Price $9.95 Diskette**
DYNACOMP now distributes the 20+ volume NSSE library. Most of these diskettes offer an outstanding value for the purchase price. Write for details regarding the contents of this library and quantity (four or more) purchases.

## Availability

DYNACOMP software is supplied with complete documentation containing clear explanations and examples. Unless otherwise specified, all programs will run within 16K program memory space (ATARI requires 24K). Except where noted, programs are available on ATARI, PET, TRS-80 (Level II) and Apple (Applesoft) cassette and diskette as well as North Star single density (double density compatible) diskette. Additionally, most programs can be obtained on standard (IBM format) 8" CP/M floppy disks for systems running under MBASIC.

⁺ *ATARI, PET, APPLE II, TRS-80, NORTH STAR, CP/M and IBM are registered trade names and/or trademarks.*

## BUSINESS and UTILITIES

**MAIL LIST II (Apple and North Star diskettes only)** — **Price: $24.95**
This many-featured program now includes full alphabetic and zip code sorting as well as file merging. Entries can be retrieved by user-defined code, client name or Zip Code. The printout format allows the use of standard size address labels. Each diskette can store more than 1100 entries (single density North Star or Apple DOS 3.2; over 2200 with double density North Star or Apple DOS 3.3)!

**FORM LETTER SYSTEM (FLS) (Apple and North Star diskette only)** — **Price: $17.95**
FLS may be employed to generate individually addressed form letters. The user creates the address file and separately composes the letter. FLS will then print form letters using each address. FLS is completely compatible with MAIL LIST II, which may be used to manage your address files.

FLS and MAIL LIST II are available as a combined package for $37.95.

**TEXT EDITOR I (Letter Writer)** — **Price: $14.95 Cassette/$18.95 Diskette**
An easy to use, line-oriented text editor which provides variable line widths and simple paragraph indexing. This text editor is ideally suited for composing letters and is quite capable of handling much larger jobs. Available for all computers.

**PERSONAL FINANCE SYSTEM (ATARI only)** — **Price: $34.95 Diskette**
PFS is a single disk menu oriented system composed of 10 programs designed to organize and simplify your personal finances. Features include a 300 transaction capacity; fast access; 26 optional user codes; data retrieval by month, code or payee; optional printing of reports; checkbook balancing; bar graph plotting and more. Also provides on the diskette is ATARI DOS 2.

**FINDIT (North Star only)** — **Price: $19.95**
This is a three-in-one program which maintains information accessible by keywords of three types: Personal (eg: last name), Commercial (eg: plumbers) and Reference (eg: magazine articles, record albums, etc). In addition to keyword searches, there are birthday, anniversary and appointment searches for the personal records and appointment searches for the commercial records. Reference records are accessed by a single keyword or by cross-referencing two or three keywords.

**DFILE (North Star only)** — **Price: $19.95**
This handy program allows North Star users to maintain a specialized data base of all files and programs in the stack of disks which invariably accumulates. DFILE is easy to set up and use. It will organize your disks to provide efficient locating of the desired file or program.

**COMPARE (North Star only)** — **Price: $12.95**
COMPARE is a single disk utility software package which compares two BASIC programs and displays the file sizes of the programs in bytes, the lengths in terms of the number of statement lines, and the line numbers at which various listed differences occur. COMPARE permits the user to examine versions of his software to verify which are the more current, and to clearly identify the changes made during development.

**COMPRESS (North Star only)** — **Price: $12.95**
COMPRESS is a single-disk utility program which removes all unnecessary spaces and (optionally) REMark statements from North Star BASIC programs. The source file is processed one line at a time, thus permitting very large programs to be compressed using only a small amount of computer memory. File compressions of 20-50% are commonly achieved.

**GRAFIX (TRS-80 only)** — **Price: $12.95 Cassette/$16.95 Diskette**
This unique program allows you to easily create graphics directly from the keyboard. You "draw" your figure using the program's extensive cursor controls. Once the figure is made, it is automatically appended to your BASIC program as a string variable. Draw a "happy face", call it H$ and then print it from your program using PRINT H$! This is a very easy way to create and save graphics.

**TIDY (TRS-80 only)** — **Price: $10.95 Cassette/$14.95 Diskette**
TIDY is an assembly language program which allows you to renumber the lines in your BASIC programs. TIDY also removes unnecessary spaces and REMark statements. The result is a compacted BASIC program which uses much less memory space and executes significantly faster. Once loaded, TIDY remains in memory; you may load any number of BASIC programs without having to reload TIDY!

## STATISTICS and ENGINEERING

**DATA SMOOTHER (Not available for ATARI)** — **Price: $14.95 Cassette/$18.95 Diskette**
This special data smoothing program may be used to rapidly derive useful information from noisy business and engineering data which are equally spaced. The software features choice in degree and range of fit, as well as smoothed first and second derivative calculation. Also included is automatic plotting of the input data and smoothed results.

**FOURIER ANALYZER (Available for all computers)** — **Price: $14.95 Cassette/$18.95 Diskette**
Use this program to examine the frequency spectra of limited duration signals. The program features automatic scaling and plotting of the input data and results. Practical applications include the analysis of complicated patterns in such fields as electronics, communications and business.

**TFA (Transfer Function Analyzer)** — **Price: $19.95 Cassette/$23.95 Diskette**
This is a special software package which may be used to evaluate the transfer functions of systems such as hi-fi amplifiers and filters by examining their response to pulsed inputs. TFA is a major modification of FOURIER ANALYZER and contains an engineering-oriented decibel versus log-frequency plot as well as data editing features. Whereas FOURIER ANALYZER is designed for educational and scientific use, TFA is an engineering tool. Available for all computers.

**HARMONIC ANALYZER (Available for all computers)** — **Price: $24.95 Cassette/$28.95 Diskette**
HARMONIC ANALYZER was designed for the spectrum analysis of repetitive waveforms. Features include data file generation, editing and storage/retrieval as well as data and spectrum plotting. One particularly unique facility is that the input data need not be equally spaced or in order. The original data is sorted and a cubic spline interpolation is used to create the data file required by the FFT algorithm.

FOURIER ANALYZER, TFA and HARMONIC ANALYZER may be purchased together for a combined price of $44.95 (three cassettes) and $56.95 (three diskettes).

**REGRESSION I (Available for all computers)** — **Price: $19.95 Cassette/$23.95 Diskette**
REGRESSION I is a unique and exceptionally versatile one-dimensional least squares "polynomial" curve fitting program. Features include very high accuracy; an automatic degree determination option; an extensive internal library of fitting functions; data editing; automatic data and curve plotting; a statistical analysis (eg: standard deviation, correlation coefficient, etc.) and much more. In addition, new fits may be tried without reentering the data. REGRESSION I is certainly the cornerstone program in any data analysis software library.

**REGRESSION II (PARAFIT) (Available for all computers)** — **Price: $19.95 Cassette/$23.95 Diskette**
PARAFIT is designed to handle those cases in which the parameters are imbedded (possibly nonlinearly) in the fitting function. The user simply inserts the functional form, including the parameters (A(1), A(2), etc.) as one or more BASIC statement lines. Data and results may be manipulated and plotted as with REGRESSION I. Use REGRESSION I for polynomial fitting, and PARAFIT for those complicated functions.

**MULTILINEAR REGRESSION (MLR) (Available for all computers)** — **Price: $19.95 Cassette / $23.95 Diskette**
MLR is a professional software package for analyzing data sets containing two or more linearly independent variables. Besides performing the basic regression calculation, this program also provides easy to use data entry, storage, retrieval and editing functions. In addition, the user may interrogate the solution by supplying values for the independent variables. The number of variables and data size is limited only by the available memory.

REGRESSION I, II and MULTILINEAR REGRESSION may be purchased together for $49.95 (three cassettes) or $61.95 (three diskettes).

**BASIC SCIENTIFIC SUBROUTINES, Volume I (Not available for ATARI)**
DYNACOMP is the exclusive distributor for the software keyed to the text *BASIC Scientific Subroutines, Volume I* by F. Ruckdeschel (see the BYTE/McGraw-Hill advertisement in BYTE magazine, January 1981). These subroutines have been assembled according to chapter. Included with each collection is a menu program which selects and demonstrates each subroutine.

- Collection #1: Chapters 2 and 3: Data and function plotting, complex variables
- Collection #2: Chapter 4: Matrix and vector operations
- Collection #3: Chapters 5 and 6: Random number generators, series approximations

Price per collection: $14.95 Cassette/$18.95 Diskette

All three collections are available for $39.95 (three cassettes) and $49.95 (three diskettes).

Because the text is a vital part of the documentation, *BASIC Scientific Subroutines, Volume I* is available from DYNACOMP for $19.95 plus 75¢ postage and handling.

**ROOTS (Available for all computers)** — **Price $9.95 Cassette/$13.95 Diskette**
In a nutshell, ROOTS simultaneously determines all the zeroes of a polynomial having real coefficients. There is no limit on the degree of the polynomial, and because the procedure is iterative, the accuracy is generally very good. No initial guesses are required as input, and the calculated roots are substituted back into the polynomial and the residuals displayed.

## Ordering Information

All orders are processed and shipped postpaid within 48 hours. Please enclose payment with order along with computer information. If paying by VISA or Master Card, include all numbers on card. For orders outside North America add 10% for shipping and handling.

Add $2.50 to diskette price for 8" floppy disk (IBM format soft sectored, CP/M, Microsoft BASIC)

*TRS-80 diskettes are not supplied with DOS or BASIC.

**Deduct 10% when ordering 3 or more programs.**

Ask for **DYNACOMP** programs at your local software dealer. Write for detailed descriptions of these and other programs from **DYNACOMP**.

**DYNACOMP, Inc.** Dept. E
1427 Monroe Avenue
Rochester, New York 14618
24 hour mail order phone: (716)586-7579
Office phone (9AM-5PM EST): (716)442-8960

VISA — MasterCard

**New York State residents please add 7% NYS sales tax.**

Program Compactor

```
1 A=1:REM *** TEST PROGRAM ***
2 REM "RUN 62000" TO COMPACT THE TEST
PROGRAM
3 :  ::C=3:D=4:REM AAAAA
4 END:DON'T SEE THIS AFTER COMPACTION
5 RETURN:NOR THIS
6 GOTO 11111:NOR THIS
7 A$="SEE THIS":REM NOT THIS
999 STOP
62000 REM
62001 REM    *** COMPACTOR ***
62002 REM
62003 REM         Edward H. Carlson
62004 REM         3872 Raleigh Drive
62005 REM         Okemos MI 48864
62006 REM         (517) 349-1219
62007 REM
62010 PRINT:PRINT:PRINT "COMPACTING":
PRINT:PRINT
62015 DIM L(80):A=3*256:AP=A+1:AD=A-3
62020 FOR I=1 TO 9999:A=A+4:REM NEW
LINE
62025 IF L<>0 THEN GOSUB 62600
62035 L=PEEK(A-1)+PEEK(A)*256:AN=0
62036 IF L>9999 THEN POKE AP,0:POKE
AP+1,0:END
62039 REM REMOVE LEADING COLONS AND SPACES
62040 A=A+1:B=PEEK(A):IF (B=32)OR(B=58) THEN 62040
62050 A=A-1:FOR K=1 TO 255:A=A+1:B=PEEK(A)
62060 IF B=0 THEN NEXT I
62065 IF B=142 THEN GOTO 62100
62068 IF (B=128)OR(B=143)OR(B=141) THEN L(AN)=B:AN=AN+1:GOTO 62100
62070 IF B=58 THEN GOTO 62400
62072 REM STORE CHAR. FOR COMPACT LINE
62073 IF B<>32 THEN L(AN)=B:AN=AN+1
62075 IF B=136 THEN GOTO 62200
62080 IF B=34 THEN GOTO 62300
62090 NEXT K:STOP
62100 FOR K=1 TO 255:A=A+1:B=PEEK(A):REM LOOKING FOR LINE END
62110 IF B=0 THEN NEXT I
62120 NEXT K
62200 FOR K=1 TO 255:A=A+1:B=PEEK(A):REM FOUND "GOTO"
62210 IF B=0 THEN NEXT I
62215 IF B=32 THEN A=A+1:B=PEEK(A):GOTO62210
62220 IF B=58 THEN GOTO 62100
62225 L(AN)=B:AN=AN+1:NEXT K
62300 FOR K=1 TO 255:A=A+1:B=PEEK(A):REM FOUND " CHAR.
62320 IF B=34 THEN L(AN)=B:AN=AN+1:GOTO 62090
62325 IF B=0 THEN NEXT I
62327 IF B=58 THEN 62400
62330 L(AN)=B:AN=AN+1:NEXT K
62400 A=A+1:B=PEEK(A):IF (B=32)OR(B=58) THEN 62400:REM FOUND :
62410 IF B=0 THEN NEXT I
62420 IF B=142 THEN GOTO 62100
```

```
62430 L(AN)=58:L(AN+1)=B:AN=AN+2:
GOTO62120
62600 PRINT L;:REM POKE MEMORY WITH
COMPACTED LINE
62601 AH=INT((A-3)/256):AL=(A-3)
-256*AH
62602 POKE AP,AL:POKE AP+1,AH:PRINT
TAB(8) AL;AH;
62603 REM AN IS LENGTH OF COMPACTED
LINE
62604 IF AN=0 THEN PRINT:RETURN
62605 AH=INT(AP/256):AL=AP-256*AH
62607 PRINT TAB(16) AL;AH;
62608 POKE AD,AL:POKE AD+1,AH:AD=AP:
AP=AP+2
62610 AH=INT(L/256):AL=L-256*AH
62611 POKE AP,AL:AP=AP+1:POKE AP,AH:
AP=AP+1
62616 FOR I=0 TO AN-1:POKE AP,L(I):
PRINT CHR$(L(I));:AP=AP+1:NEXT I
62620 POKE AP,0:AP=AP+1:PRINT:RETURN
62700 REM
62705 REM    VARIABLES
62710 REM
62715 REM  L   LINE NUMBER
62720 REM  A   ADDRESS BEING SCANNED
62725 REM  AP  ADDRESS BEING POKED
62730 REM  AD  ADDRESS OF PREVIOUS LINE
62732 REM  AN  INDEX IN L(I) OF CHAR.
TO BE POKED
62735 REM  B   CHARACTER BEING SCANNED
62739 REM
62740 REM    TOKENS AND CHARACTERS
62741 REM
62745 REM  32     SPACE
62750 REM  34     " CHARACTER
62752 REM  58     : CHARACTER
62755 REM 128     END TOKEN
62760 REM 136     GOTO
62765 REM 141     RETURN
62770 REM 142     REM
62775 REM 143     STOP
OK
```



Now if the compacted line is zero length, one returns to the "I" loop and starts compacting the next line. If not, one has to update the pointer in the previous compacted line and this is done in lines 62605 and 7. This restores continuity to the chain of pointers. Then the line number is POKEd in, and then the line of compacted text and finally the null at the end.

While developing this program, I used a somewhat longer "Test Program" than I am giving here. The final test was made on two of my old war horses. The first is a Knight's Tour and the other I informally call "Godzilla Eats Tanks". Both programs are rather long, involved, and use PEEKed and POKEd graphics extensively. "Godzilla" uses PEEKed keyboard input with ANDed and ORed data. ("Godzilla" is invisible and lays down an invisible trail that is sniffed out by a "stench seeking" guided missile.) Both programs ran successfully after compacting to about 75% of their original length. However, I did need to repair the Knight's Tour because I had some GOTO's that went to free standing REM's. These REM's were removed by the Compactor and had to be put back in by hand. I have since learned my lesson. Never GOTO or GOSUB to a REM.

This program is very useful but not yet the ultimate in compaction technique. At least one of the large software houses sells a compactor that is combined with a "branch locator" so that statements which are not the target lines of a GOSUB or GOTO are candidates to be put on the same line, separated by colons. It would certainly be faster and easier to buy such an efficient compactor in preference to tapping this one in through the keyboard, but if your interest is in playing around with logical tasks, you may prefer to modify this program to have multistatement-per-line capability. ©

# The Mysterious And Unpredictable RND

Bob Albrecht and
George Firedrake

*From a book of the same name by Dymax Publishing Company; copyright ©1980 Dymax. Permission to reprint by teachers for classroom use is granted.*

*Editor's Note: You may reach Bob & George by mail at: P.O. Box 310 Menlo Park, CA 94025*

## Create An Adventure

Have you played *Dungeons and Dragons* or *Runequest* or *Tunnels and Trolls*? These are fantasy adventure games. To play these games, you create a character. You then guide your character through adventures in a world created by a Gamemaster and . . . sigh . . . completely controlled by the Gamemaster.

If you don't know about these games, we suggest you get rule books from the following companies.

- *Dungeons and Dragons* (D & D) from TRS Hobbies, P.O. Box 756, Lake Geneva, WI 53147.
- *Runequest* from the Chaosium, P.O. Box 6302, Albany, CA 94706.
- *Tunnels and Trolls* from Flying Buffalo, Inc., P.O. Box 1467, Scottsdale, AZ 85252.

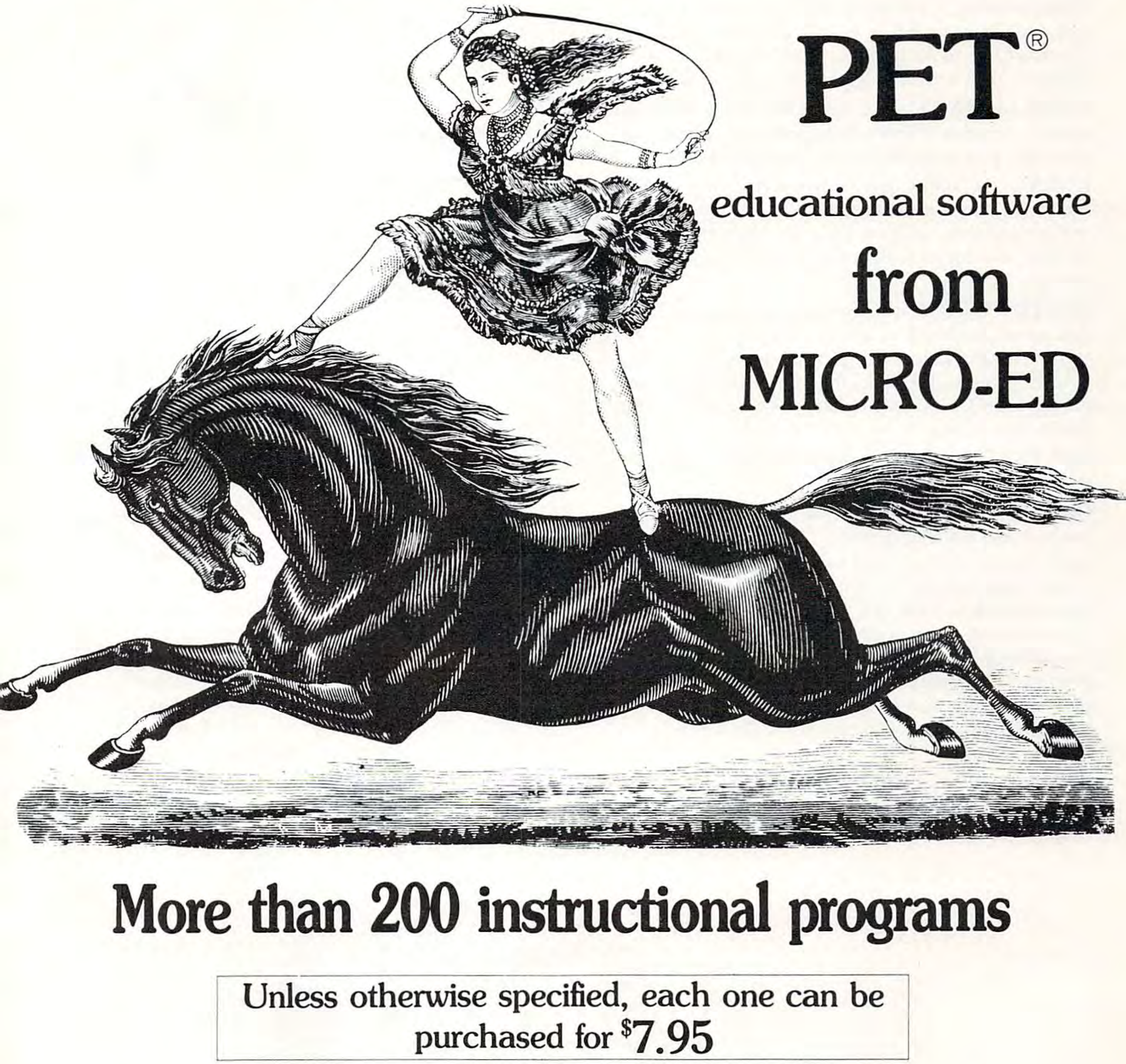
To create a character to play any of these games, you will need to roll three six-sided dice several times. We will demonstrate by rolling a character to play *Tunnels and Trolls.* Why? Because T & T is the simplest game for beginners to begin with.

An adventurer has six *Prime Attributes:* STRENGTH, INTELLIGENCE, LUCK, CONSTITUTION, DEXTERITY and CHARISMA. The six Prime Attributes determine the character's ability to speak languages, use weapons, overcome monsters, sense things, use magic, and . . . go successfully adventuring in the Gamemaster's world.

The Prime Attributes are usually abbreviated, as follows.

| | |
|---|---|
| STRENGTH | STR |
| INTELLIGENCE | INT or IQ |
| LUCK | LK |
| CONSTITUTION | CON |
| DEXTERITY | DEX |
| CHARISMA | CHR |

# Perfectly Balanced



# PET®

educational software

from

MICRO-ED

## More than 200 instructional programs

Unless otherwise specified, each one can be purchased for $7.95

You may wish to order tapes by the MICRODOZEN. Any twelve $7.95 tapes can be purchased for $84.00.

Send for free catalogue:

MICRO-ED, Inc. • P.O. Box 24156 • Minneapolis, MN 55424

or telephone us at (612) 926-2292

All programs work with any 8K PET, old or new.

PET is the registered trademark for Commodore Business Machines, Santa Clara, CA.

In T & T (**Tunnels and Trolls**), these are the Prime Attributes:

**STRENGTH** is primarily the ability to exert force—lifting, shoving, pushing down, etc. It shows how much junk (measured in weight units, see below) the character can move around. If his strength is ever depleted until it goes to 0, the character is dead.

**INTELLIGENCE** is the measure of a character's ability to reason clearly, solve problems, remember well, etc. It is also a factor in language ability.

**LUCK** is the ability to be in the right place at the right time, or to put something else in the right place at the right time. It is useful in avoiding traps, striking luck blows with weapons, and gambling of all sorts.

**CONSTITUTION** is the general measure of a character's health. It is also the measure of endurance and how much punishment the body can absorb before it dies. Hits taken in combat are subtracted from Constitution. If CON ever goes to 0, the character dies.

**DEXTERITY** refers to manual dexterity and general agility.

**CHARISMA** is the measure of one's personal attractiveness and leadership ability. It is not necessarily synonymous with personal beauty, although there is often a high correlation. Charisma is the only attribute which can fall to 0, or even go negative, without resulting in death. Generally speaking, characters with charismas less than 7 are unappreciated in human society, and anything less than 3 is positively unwelcome. Monstrous characters, when rated with attributes, have negative charismas.

And so we come to the final tasks of this booklet.

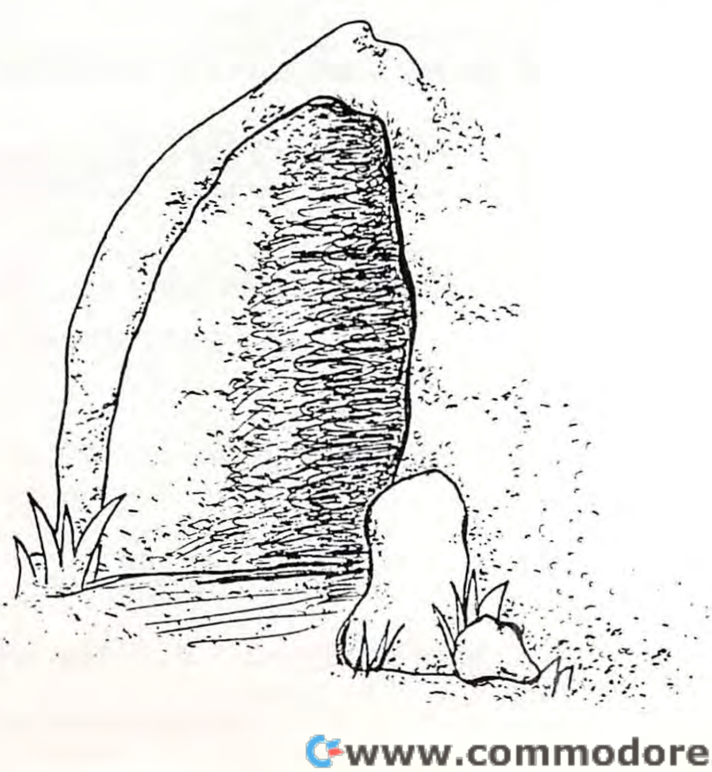
**Exercise 15.** Write a program to roll a T & T character. A RUN might look like this.

*The definitions of the Prime Attributes are taken from the rulebook TUNNELS & TROLLS, copyright 1975, 1977, 1979 by Ken St. Andre, published by Flying Buffalo, Inc., P.O. Box 1467 Scottsdale, AZ 85252 and is reprinted by permission.*

```
STR         15
IQ          11
LK          8
CON         16
DEX         14
CHR         9

READY
■
```


Hmmm...obviously a fighter. Remember, when we roll three dice, the possible outcomes are 3 to 18. So, our character is strong (STR = 15), agile (DEX = 14), with a very high ability to sustain damage (CON = 16). He or she is about average in intelligence (IQ = 11), obviously *not* a leader (CHR = 9) and must depend on skill, not luck (LK = 8).

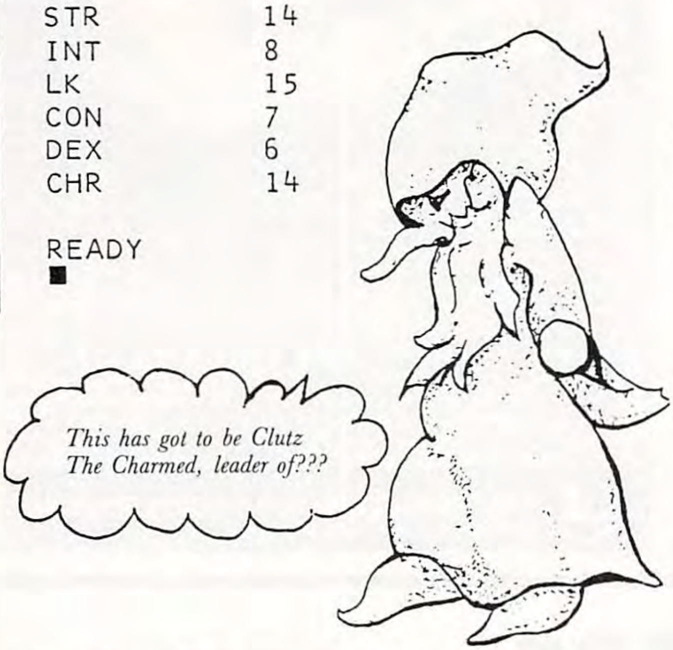


In fantasy games, usually a bunch of characters get together to explore the Gamemaster's universe. So, let's roll some more characters to accompany Fibak the Fighter, who awaits in the frontier town of Ziredrac, hoping to collect a company of adventurers to explore the caves under Mt. Skybison, a place of jagged (but sometimes fuzzy) peaks and mysterious dark valleys that sometimes light up with brilliant flashes of wisdom.

We RUN the program again.

```
STR          14
INT          8
LK           15
CON          7
DEX          6
CHR          14

READY
■
```



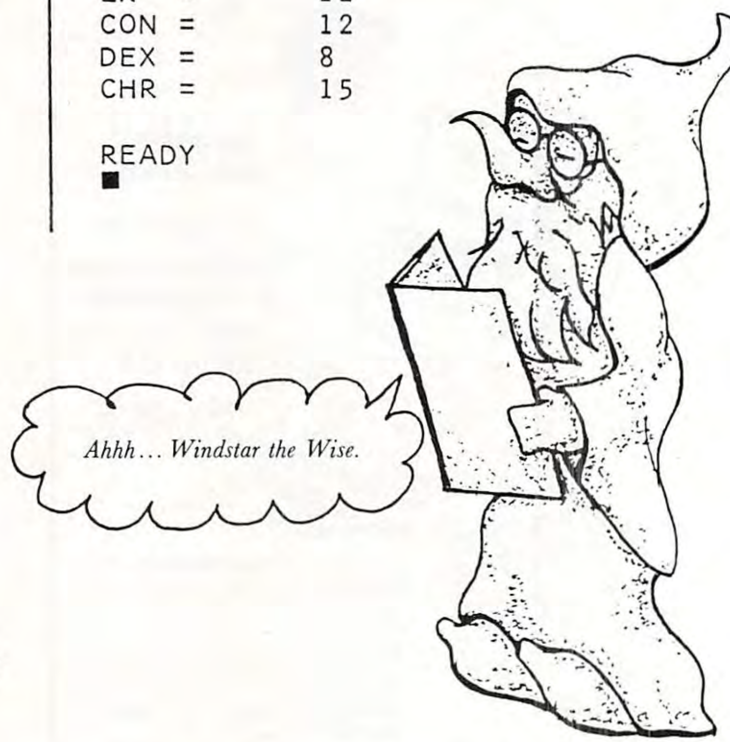

This is, indeed, a strange character! Strong (STR = 14), but easily damaged (CON = 7). Clumsy (DEX = 6). But look at luck (LK = 15) and charisma (CHR = 14). This character will convince others to follow her or him into . . . what? (Again, look at INT = 8.) But, there is that luck.

Help! Let's roll another adventurer.



```
STR =        11
INT =        17
LK  =        12
CON =        12
DEX =        8
CHR =        15

READY
■
```




Saved! Our group is saved! Windstar the Wise wandered by, saw our forlorn little group of adventurers and decided to take charge.

**Exercise 16.** Describe Windstar as we described Fibak and Clutz. Also describe the way in which Windstar, Fibak and Clutz might work together to explore the Gamemaster's universe, overcome monsters, acquire treasure and . . . survive!

**Exercise 17.** (and last of this booklet . . .) The group of adventurers now numbers three: FIBAK, CLUTZ, WINDSTAR.

Too small a group! They could never survive in the Gamemaster's world. They need at least four more adventurers. More are OK. So, you roll up four more adventurers, then tell who they are and how they relate to and work with our three adventurers.

*The Dragons of Pern wish you well in your adventuring.*

*Do it wisely with luck.*

*Editor's Note: Next time we come to the end of a series, we'll be printing "Solutions & Stuff"; the author's solutions to the problems raised in this series. Hope you've enjoyed it.* ©

# Micros With The Handicapped

Susan Semancik and
The Delmarva Computer Club

One of our major objectives at this time is to devise an inexpensive means for providing a listing for a blind programmer. The possibilities we've considered to date are as follows:

a) Have someone continuously available to read the program to the programmer.
b) Buy a speech synthesizer that will vocalize BASIC words, letters, punctuation marks, numbers, graphics characters, and control characters.
c) Build a tactile device that will pulse Braille equivalents of each character encountered in a listing.
d) Sound out the Braille equivalents of each character encountered in a listing by using a different note or sound for each dot in the Braille cell using the CB2 line.

We know that possibility a) works. We have been forced through lack of money and/or equipment to use this means to solve the blind programmer's needs. But, it is certainly not a desirable solution, especially to the programmer. One thing that is soon apparent to anyone working with the handicapped is that the majority have an overwhelming desire for independence and self-reliance. To support this desire, we must look beyond possibility a).

Possibility b) is possible since Commodore has announced the development of a synthesizer for the PET, but we do not yet have access to one. It's price of almost $400 may or may not be a deterrent to its wide-spread use. We'll have to wait and see. Possibility c) is being considered by the Delmarva Computer Club at this time, and hopefully we can give you details on its use and cost in the very near future. The other devices that we know about that are currently on the market are very expensive and/or computer dependent. For example, Maryland Computer Services, Inc., Bel Air, Maryland, has advertised a Hewlett-Packard desktop computer and a talking interface that provides spoken output of 64 ASCII characters for $10,500. Triformation Systems, Inc., Stuart, Florida, has advertised a high-speed braille output on paper tape from computer via built-in acoustic coupler for $2,950. And this past March, ELINFA, Inc., Washington, D.C., announced a portable braille recorder with braille display and computer interface for $4,600.

That brings us to possibility d), which is immediately available to anyone with an amplifier and speaker connected to the PET's CB2 line. Since this is commonly used already to produce sound and music, this would mean no additional expense for most people, and so seems to be a good place to start exploring. It is certainly inexpensive, and an ear plug can be used with the speaker so that only the user can hear the sounds. This would require the blind programmer to learn something new; but, essentially the code is the same; only the method of perceiving the code is new. Rather than feeling the dot's position within the braille cell, the user will listen to determine a note's position within a range of notes.

The program entitled "A Sound Idea for the Blind" is an assembly language program that will run on either an OLD or NEW ROM 8K PET computer. It is designed to teach anyone the Braille equivalents of letters in both a visual and auditory form. After typing RUN, the user can type any letter on the keyboard and will automatically see that letter's braille dot configuration, and hear the notes that correspond to those dots. The user can also hit any number from 1 through 6 to hear singly the notes assigned to each dot position. This program could be used by the visual user to learn Braille or by the non-visual to learn the sounds associated with each position of the Braille cell.

The speed at which the notes are sounded is controlled by the data number in line 240. To speed it up, put a smaller number like 75 in place of the 200. If a change in either the quality of the sound or the pitch is desired, the data numbers in line 310 can be changed. These are listed pairwise for the six dot positions of the cell.

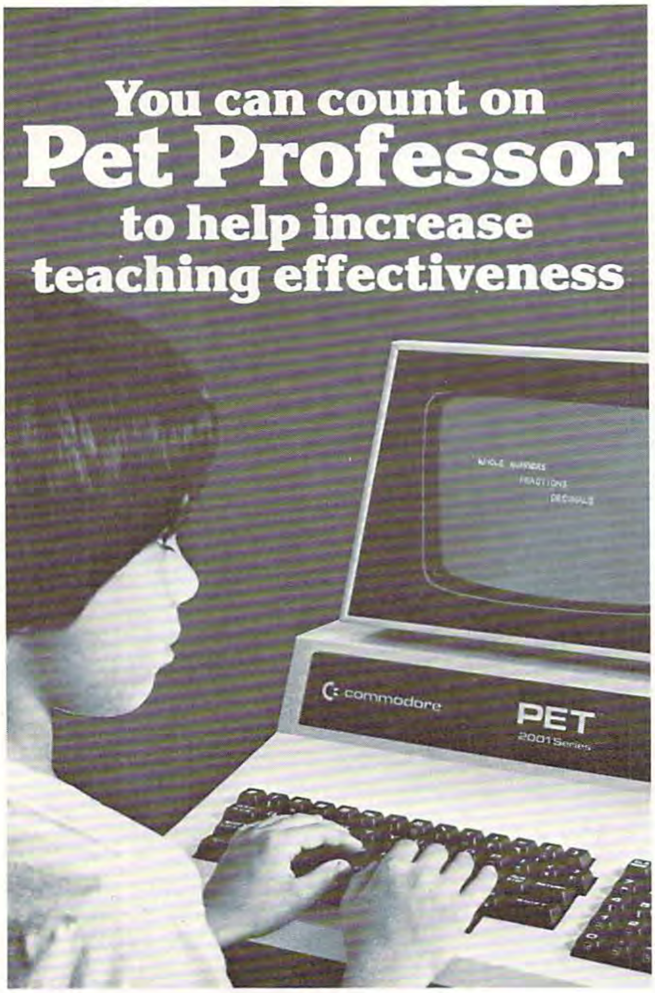
To test how well the sounds have been learned, this program can be modified as in the program entitled "Braille Letter Tester". The Braille codes of the 26 letters of the alphabet are randomly sounded and printed on the screen for the user to identify. The sound can be repeated by pressing the equal sign key instead of a letter. The line

```
455 IF A$<"A" OR A$>"Z" THEN 430
```

must be added so that the program is not terminated with an illegal value when a non-alphabetic character is entered. If the visual user really wants to "see" what the code would sound like to someone blind, the printing of the Braille code on the screen can be eliminated by changing six operation codes to NOP with the addition of the following line:

```
295 POKE 838, 234: POKE 839, 234: POKE 840, 234:
    POKE 883, 234: POKE 884, 234: POKE 885, 234
```

We hope to be able to expand this "sound" concept to realize our full objective of a program's listing for a blind programmer. Let us know if this is a realistic alternative, or if there are other possibilities that we haven't considered.




## It reteaches the 4 basic arithmetic operations step-by-step.

PET PROFESSOR includes 77 programs that reteach the four fundamental arithmetic operations for whole numbers, fractions and decimals by providing more than just practice drills. Each program includes a complete tutorial sequence that takes a problem apart, then leads the student step-by-step through solving it.

Each program covers a single objective to allow the teacher to select the skill to be reinforced, It states the objective, supplies vocabulary, displays an example, then leads the student step-by-step through other examples. When the student is ready, a quiz requiring writing is given, then scored by the computer. Every program has been carefully selected. All problems were written by teachers and tested in classrooms..

Use PET PROFESSOR to supplement classroom instruction in individualized or traditional programs. Available for PET 2001 Series. Will run within 8K on any PET.

- Whole numbers—32 programs @ $235.00
- Fractions—24 programs @ $175.00
- Decimals—21 programs @ $175.00

**Buy all 3 for $499.00. Save $86.00!**

Send $5.00 for more info., sample tape & shipping (deducted from total purchase).

List of other educational programs available on request.

Melad Associates, Inc.
66 Main Street, South River, N.J. 08882 • (201) 238-7330

---

**By Netronics**

# ASCII/BAUDOT, STAND ALONE Computer Terminal

**COMPLETE FOR ONLY . . . $149.95**

ASCII/BAUDOT
Computer Terminal Kit

The Netronics ASCII/BAUDOT Computer Terminal Kit is a microprocessor-controlled, stand alone keyboard/terminal requring no computer memory or software. It allows the use of either a 64 or 32 character by 16 line professional display format with selectable baud rate, RS232-C or 20 ma. output, full cursor control and 75 ohm composite video output.

The keyboard follows the standard typewriter configuration and generates the entire 128 character ASCII upper/lower case set with 96 printable characters. Features include onboard regulators, selectable parity, shift lock key, alpha lock jumper, a drive capability of one TTY load, and the ability to mate directly with almost any computer, including the new Explorer/85 and ELF products by Netronics.

The Computer Terminal requires no I/O mapping and includes 1k of memory, character generator, 2 key rollover, processor controlled cursor control, parallel ASCII/BAUDOT to serial conversion and serial to video processing—fully crystal controlled for superb accuracy. PC boards are the highest quality glass epoxy for the ultimate in reliability and long life.

### VIDEO DISPLAY SPECIFICATIONS

The heart of the Netronics Computer Terminal is the microprocessor-controlled Netronics Video Display Board (VID) which allows the terminal to utilize either a parallel ASCII or BAUDOT signal source. The VID converts the parallel data to serial data which is then formatted to either RS232-C or 20 ma. current loop output, which can be connected to the serial I/O on your computer or other interface, i.e., Modem.

When connected to a computer, the computer must echo the character received. This data is received by the VID which processes the information, converting to data to video suitable to be displayed on a TV set (using an RF modulator) or on a video monitor. The VID generates the cursor, horizontal and vertical sync pulses and performs the housekeeping relative to which character and where it is to be displayed on the screen.

**Video Output:** *1.5 P/P into 75 ohm (EIA RS-170)* • **Baud Rate:** *110 and 300 ASCII* • **Outputs:** *RS232-C or 20 ma. current loop* • **ASCII Character Set:** *128 printable characters—*

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

**BAUDOT Character Set:** *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - ? : • 3 $ # ( ) . , 9 0 1 4 ! 5 7 ; 2 / 6 8* • **Cursor Modes:** *Home, Backspace, Horizontal Tab, Line Feed, Vertical Tab, Carriage Return. Two special cursor sequences are provided for absolute and relative X-Y cursor addressing* • **Cursor Control:** *Erase, End of Line, Erase of Screen, Form Feed, Delete* • **Monitor Operation:** *50 or 60Hz (jumper selectable.*

Continental U.S.A. Credit Card Buyers Outside Connecticut

**CALL TOLL FREE 800-243-7428**

To Order From Connecticut Or For Technical Assistance, Etc. **Call (203) 354-9375**

**Netronics R&D Ltd., Dept.**
**333 Litchfield Road, New Milford, CT 06776**

*Please send the items checked below—*

- ☐ **Netronics Stand Alone ASCII Keyboard/Computer Terminal Kit, $149.95** plus $3.00 postage & handling.
- ☐ **Deluxe Steel Cabinet** for Netronics Keyboard/Terminal In Blue/Black Finish, **$19.95** plus $2.50 postage and handling.
- ☐ **Video Display Board Kit** alone (less keyboard), **$89.95** plus $3 postage & handling.
- ☐ **12" Video Monitor** (10 MHz bandwidth) fully assembled and tested, **$139.95** plus $5 postage and handling.
- ☐ **RF Modulator Kit** (to use your TV set for a monitor), **$8.95** postpaid.
- ☐ **5 amp Power Supply Kit** In Deluxe Steel Cabinet (±8VDC @ 5 amps, plus 6-8 VAC), **$39.95** plus $2 postage & handling.

Total Enclosed (Conn. res. add sales tax) $______
By—
☐ Personal Check ☐ Cashiers Check/Money Order
☐ **Visa** ☐ **Master Charge** (Bank # ____)
Acct. # ______
Signature ______ Exp. Date ______
Print Name ______
Address ______
City ______
State ______ Zip ______
☐ **Send Me More Information**

---

# NEW!!!-THE ELECTRIC MOUTH*

**for S100, Elf II, Apple, TRS-80 Level II***

**From $99.95 kit**

*Now — teach your computer to talk, dramatically increasing the interaction between you and your machine.*


That's right: the ELECTRIC MOUTH actually lets your computer talk! Installed and on-line in just minutes, it's ready for spoken-language use in office, business, industrial and commercial applications, in games, special projects, R&D, education, security devices — there's no end to the ELECTRIC MOUTH's usefulness. Look at these features:

- Supplied with 143 words/letters/phonemes/numbers, capable of producing hundreds of words and phrases.
- Expandable on-board up to thousands of words and phrases (just add additional speech ROMs as they become available).
- Four models, which plug directly into S100, Apple, Elf II and TRS-80 Level II computers.
- Get it to talk by using either Basic or machine language (very easy to use, complete instructions with examples included).
- Uses National Semiconductor's "Digitalker" system.
- Includes on-board audio amplifier and speaker, with provisions for external speakers and amplifier.
- Adds a new dimension and excitement to programming; lets you modify existing programs and games to add spoken announcements of results, warnings, etc.
- Installs in just minutes.

*"Elf II" and "The Electric Mouth" are reg. trademarks of Netronics R&D Ltd. "Apple" is a reg. trademark of Apple Computer Inc. "TRS-80 Level II" is a reg. trademark of Tandy Corp.*

**Principle of Operation** The ELECTRIC MOUTH stores words in their digital equivalents in ROMs. When words, phrases, and phonemes are desired, they are simply called for by your program and then synthesized into speech. The ELECTRIC MOUTH system requires none of your valuable memory space except for a few addresses if used in memory mapped mode. In most cases, output ports (user selectable) are used.

### Spoken Material Included

| | | | | |
|---|---|---|---|---|
| ONE | SEVENTY | CONTROL | LEFT | POUND |
| TWO | EIGHTY | DANGER | LESS | PULSES |
| THREE | NINETY | DEGREE | LESSER | RATE |
| FOUR | HUNDRED | DOLLAR | LIMIT | RE |
| FIVE | THOUSAND | DOWN | LOW | READY |
| SIX | MILLION | EQUAL | LOWER | RIGHT |
| SEVEN | ZERO | ERROR | MARK | SS |
| EIGHT | AGAIN | FEET | METER | SECOND |
| NINE | AMPERE | FLOW | MILE | SET |
| TEN | AND | FUEL | MILLI | SPACE |
| ELEVEN | AT | GALLON | MINUS | SPEED |
| TWELVE | CANCEL | GO | MINUTE | STAR |
| THIRTEEN | CASE | GRAM | NEAR | START |
| FOURTEEN | CENT | GREAT | NUMBER | STOP |
| FIFTEEN | 400HERTZ TONE | GREATER | OF | THAN |
| SIXTEEN | 80HERTZ TONE | HAVE | OFF | THE |
| SEVENTEEN | 20MS SILENCE | HIGH | ON | TIME |
| EIGHTEEN | 40MS SILENCE | HIGHER | OUT | TRY |
| NINETEEN | 80MS SILENCE | HOUR | OVER | UP |
| TWENTY | 160MS SILENCE | IN | PARENTHESIS | VOLT |
| THIRTY | 320 MS SILENCE | INCHES | PERCENT | WEIGHT |
| FORTY | CENTI | IS | PLEASE | |
| FIFTY | CHECK | IT | PLUS | |
| SIXTY | COMMA | KILO | POINT | |

Also, the complete alphabet, spoken letter by letter.

Continental U.S.A. Credit Card Buyers Outside Conn. **TO ORDER CALL TOLL FREE 800-243-7428**

To Order From Connecticut Or For Technical Assistance, Etc., **Call (203) 354-9375**

**NETRONICS R&D LTD. Dept.**
**333 Litchfield Road, New Milford, CT 06776**

*Please send the items checked below:*

- ☐ S100 "Electric Mouth" kit ................ $99.95
- ☐ Elf II "Electric Mouth" kit ................ $99.95
- ☐ Apple "Electric Mouth" kit ................ $119.95
- ☐ TRS-80 Level II "Electric Mouth" kit ...... $119.95

Add $20.00 for wired & tested units. All plus $3.00 postage & insurance. Conn. res. add sales tax.

Total Enclosed $______

☐ Personal Check ☐ Cashier's Check/Money Order
☐ Visa ☐ Master Charge (Bank No. ______)
Acct. No. ______
Signature ______ Exp. Date ______
Print Name ______
Address ______
City ______
State ______ Zip ______

```
10 REM *** BRAILLE LETTER TESTER ***
20 REM - YOU WILL HEAR & SEE THE CODE ¬
     ¬FOR A RANDOM LETTER
30 REM - TYPE THE LETTER OR TYPE = TO ¬
     ¬HEAR IT AGAIN
40 REM - ASSEMBLY PROGRAM CODE
50 I=826
60 READ OP:IF OP<>-1 THEN POKE I,OP:
     ¬I=I+1:GOTO 60
100 I=5000
110 READ OP:IF OP<>-1 THEN POKE I,OP:
     ¬I=I+1:GOTO 110
120 REM %%% OLD/NEW CONVERSION %%%
130 ON PEEK(50003) GOSUB 310
140 DATA 169,147,32,210,255
150 DATA 160,0,177,6,170,169,46,157,121,
     ¬129,200,192,6,208,243,96
160 DATA 169,168,133,0,169,19,133,1
170 DATA 169,136,133,6,169,19,133,7
180 DATA 32,58,3,160,0
190 DATA 177,6,160,0,24,106,144,15,133,
     ¬252,177,6,170,169,81
200 DATA 157,121,129,32,129,3,165,252,
     ¬200,192,6,208,232,96,169
210 REM +++ TO CHANGE SPEED OF SOUND,
     ¬ CHANGE DATA LINE 220 +++
220 DATA 200
230 DATA 133,253,152,10,168,177,0,141,
     ¬74,232,200,177,0
240 DATA 141,72,232,136,152,74,168,162,
     ¬255,202,208,253,198,253,208,247
250 DATA 169,0,141,72,232,96,-1
260 DATA 0,80,160,2,82,162,1,3,9,25,17,
     ¬11,27,19,10,26,5,7,13,29,21,15,31,
     ¬23
270 DATA 14,30,37,39,58,45,61,53
280 REM +++ TO CHANGE SOUND, CHANGE ¬
     ¬DATA LINE 290 +++
290 DATA 15,240,200,180,15,120,35,90,
     ¬100,60,15,30,-1
300 GOTO 360
310 REM %%% CONVERSION ROUTINE %%%
320 POKE 834,15:POKE 858,15:POKE 862,16:
     ¬POKE 869,15:POKE 877,171:POKE 879,
     ¬15
330 POKE 890,171:POKE 900,172:POKE 925,
     ¬172
340 RETURN
350 REM - SET SOUND PARAMETERS AND ¬
     ¬CURSOR CONTROL VALUE
360 POKE 59467,16:POKE 59466,51:
     ¬X=RND(-TI):B=224-PEEK(50003)*28
370 L=26:DIM L(26):C=0:FOR I=1 TO 26:
     ¬L(I)=I:NEXT I
380 REM - GET A RANDOM LETTER FROM ¬
     ¬THOSE REMAINING
390 N=INT(L*RND(1))+1:Y=L(N):A=Y+64
400 REM ACTIVATE THE SOUND AND PRINT ¬
     ¬ROUTINE FOR THE CELL AND PRINT A ?
410 POKE 867,Y+5:SYS (847):POKE 33448,63
420 REM - GET RESPONSE FROM THE USER
430 GET A$:IF A$="" THEN 430
440 REM - IF RESPONSE IS =, THEN SOUND ¬
     ¬IT AGAIN
450 IF A$="=" THEN 410
460 REM - INCREASE THE COUNTER FOR ¬
     ¬NUMBER OF TRIES
470 C=C+1
480 REM - PRINT THE USER'S RESPONSE
490 POKE 33450,ASC(A$)-64
500 REM -PRINT THE RIGHT ANSWER AND ¬
     ¬POSITION THE CURSOR FOR COMMENT
510 POKE 33466,Y:POKE B,168:POKE B+1,
     ¬130:POKE B+2,5
520 REM - IF ANSWER IS CORRECT, ¬
     ¬ELIMINATE THE LETTER FROM THE LIST
530 IF A$=CHR$(A) THEN PRINT "CORRECT!":
     ¬X=L(N):L(N)=L(L):L(L)=X:L=L-1:
     ¬GOTO 550
540 PRINT "WRONG."
550 PRINT TAB(4);"ɾHIT ANY KEY TO ¬
     ¬CONTINUE."
560 REM - WAIT FOR USER TO HIT A KEY ¬
     ¬BEFORE CONTINUING
570 GET A$:IF A$="" THEN 570
580 REM - CONTINUE IF THERE ARE ANY ¬
     ¬MORE LETTERS
590 IF L>0 THEN 390
600 REM - TELL THE USER THE NUMBER OF ¬
     ¬TRIES TAKEN TO COMPLETE THE TEST
610 PRINT "ĥYOU COMPLETED THE 26 ¬
     ¬LETTERS IN";C:PRINT "TRIES."
620 REM - RESET PARAMETERS BEFORE ENDING
630 POKE 59467,0:POKE 59466,0:POKE ¬
     ¬59464,12
640 END
READY.
```

```
10 REM *** A SOUND IDEA FOR THE BLIND ¬
     ¬***
20 REM ===  ASSEMBLY PROGRAM CODE ===
30 I=826
40 READ OP:IF OP<>-1 THEN POKE I,OP:
     ¬I=I+1:GOTO 40
50 REM === SCREEN LOCATIONS FOR BRAILLE ¬
     ¬DOT POSITIONS (5000-5005) ===
60 REM === AND BRAILLE CODES (5006-5031)
     ¬ ===
70 REM === AND ALTERNATING TIMBRE AND ¬
     ¬NOTE VALUES (5032-5043)
80 I=5000
90 READ OP:IF OP<>-1 THEN POKE I,OP:
     ¬I=I+1:GOTO 90
100 REM %%% OLD/NEW CONVERSION %%%
110 ON PEEK(50003) GOSUB 330
120 PRINT "ĥ": SYS 847
130 DATA 169,147,32,210,255
140 DATA 160,0,177,6,170,169,46,157,121,
     ¬129,200,192,6,208,243,96
150 DATA 169,168,133,0,169,19,133,1,169,
     ¬16,141,75,232,169,15,141,74,232
160 DATA 169,136,133,6,169,19,133,7
170 DATA 32,228,255,240,251,201,3,208,
     ¬14,169,0,141,75,232,141,74,232,
     ¬169,12
180 DATA 141,76,232,96,133,252,32,58,3,
     ¬165,252,201,65,48,7,201,91,16,3,
     ¬76,172,3
190 DATA 201,49,48,211,201,55,16,207,56,
     ¬233,49,168
200 DATA 177,6,170,169,81,157,121,129,
     ¬32,207,3,76,105,3
210 DATA 56,233,59,168,177,6,160,0,24,
     ¬106,144,15,133,252,177,6,170,169,
     ¬81
220 DATA 157,121,129,32,207,3,165,252,
     ¬200,192,6,208,232,76,105,3,169
```

```
230 REM +++ TO CHANGE SPEED OF SOUND,
      ¬ CHANGE DATA LINE 240 +++
240 DATA 200
250 DATA 133,253,152,10,168,177,0,141,
      ¬74,232,200,177,0
260 DATA 141,72,232,136,152,74,168,162,
      ¬255,202,208,253,198,253,208,247
270 DATA 169,0,141,72,232,96,-1
280 DATA 0,80,160,2,82,162,1,3,9,25,17,
      ¬11,27,19,10,26,5,7,13,29,21,15,31,
      ¬23
300 REM TO CHANGE SOUND, CHANGE DATA ¬
      ¬LINE 310
310 DATA 15,240,200,180,15,120,35,90,
      ¬100,60,15,30,-1
320 END
330 REM %%% CONVERSION ROUTINE %%%
340 POKE 834,15:POKE 868,15:POKE 872,16:
      ¬POKE 897,171:POKE 902,171:
      ¬POKE 927,15
350 POKE 945,15:POKE 953,171:POKE 955,
      ¬15:POKE 966,171:POKE 978,172:
      ¬POKE 1003,172
360 RETURN
```

©

**The Delmarva Computer Club**
**P.O. Box 36**
**Wallops Island, VA 23337**

# Matrix Row Operations

## A mathematics classroom-teaching aid

William L. Hinrichs
Rockford, Ill.

After teaching the elementary row operations on matrices and their applications on a chalkboard or overhead projector for many years, I have found a better teaching tool. Using a computer and this program, Matrix Row Operations, has eliminated a major stumbling block to learning. Detailed fraction arithmetic always obscured the big picture concepts and techniques. All of the input and output for this program is in the common fraction form used in textbooks and familiar to the students. However, all fraction arithmetic including multiplication, addition, and reducing is done by the computer. This allows us to concentrate on the big picture while avoiding the details during classtime.

### Easy Conversion For Apple & TRX-8O

This program was written for a Commodore PET computer, but with minor modifications it will run on an Apple, TRS-80, or other machine. No machine language is used and the BASIC used is standard. Due to a lack of standardization of screen control statements, most of the PRINT statements will have to be modified. Loop index variables have been left off some NEXT statements and may have to be added.

### Classroom Equipment Set-up

All that is required to use this program with a small group is a PET computer with 8-K RAM and any ROM version. Due to the array size limitation of version 1.0, the program's use is limited but it will run and can be used effectively. For a large group, I use a Petunia video interface board from HUH Electronics to connect our PET to a TV monitor and then connect that monitor to a second monitor with very satisfactory results. This arrangement has worked well with up to 40 students in a class.

### Sample Run

Before I describe the program, a sample run is in order. We will solve a system of two equations in two unknowns by matrix row reduction.

*Problem:*

Find all solutions, if any, of this system of linear equations:

**$5x + 3y = 9$ and $-4x + 2y = -16$.**

### Mathematical Analysis

We will set up a matrix representation of this system of equations and manipulate it by applying the elementary row operations. At each step, we will have a matrix representation of a system of equations with the same solutions as the original one. Our goal is to end up with a matrix representation in row-reduced form, from which the solution to all the systems can be easily read. The original and row-reduced matrices are illustrated in the computer solution below. For more information on the mathematics of matrices, I would recommend College Algebra, Gustafson & Frisk, Brooks/Cole Pub. Co., 1980, pp. 116-175. Most other college algebra texts also have a chapter on matrices.

### Computer Demonstration

1. LOAD the program and RUN it.
   The displays which follow are exact copies of the screen. Each display has had all inputs added to the right of the ?, and when the RETURN key is pressed we go on to the next screen. COMMAND? is always the first input requested on each screen. For some commands, additional inputs are requested above the COMMAND? line.
2. The command menu will be displayed and our response to COMMAND? is 'EM' for enter matrix.

```
**************************************
*                                    *
* COMMAND MENU                       *
*                                    *
*    ?  DISPLAY MENU                 *
*   EM  ENTER MATRIX                 *
*   *R  MULTIPLY ROW                 *
*  *R+  MULTIPLY ROW AND ADD         *
*   IR  INTERCHANGE ROWS             *
*  DOM  DISPLAY ORIGINAL MATRIX      *
*  DIM  DISPLAY CURRENT-1 MATRIX     *
*  DCM  DISPLAY CURRENT MATRIX       *
*    B  BACKUP 1 STEP                *
*    Q  QUIT                         *
*                                    *
* COMMAND? EM                        *
*                                    *
**************************************
```

# 80 COLUMN GRAPHICS


The Integrated Visible Memory for the PET has now been redesigned for the new 12" screen 80 column and forthcoming 40 column PET computers from Commodore. Like earlier MTU units, the new K-1008-43 package mounts inside the PET case for total protection. To make the power and flexibility of the 320 by 200 bit mapped pixel graphics display easily accessible, we have designed the Keyword Graphic Program. This adds 45 graphics commands to Commodore BASIC. If you have been waiting for easy to use, high resolution graphics for your PET, isn't it time you called MTU?

The image on the screen was created by the program below.

```
 10 VISMEM: CLEAR
 20 P=160: Q=100
 30 XP=144: XR=1.5*3.1415927
 40 YP=56: YR=1: ZP=64
 50 XF=XR/XP: YF=YP/YR: ZF=XR/ZP
 60 FOR ZI=-Q TO Q-1
 70 IF ZI<-ZP OR ZI>ZP GOTO 150
 80 ZT=ZI*XP/ZP: ZZ=ZI
 90 XL=INT(.5+SQR(XP*XP-ZT*ZT))
100 FOR XI=-XL TO XL
110 XT=SQR(XI*XI+ZT*ZT)*XF: XX=XI
120 YY=(SIN(XT)+.4*SIN(3*XT))*YF
130 GOSUB 170
140 NEXT XI
150 NEXT ZI
160 STOP
170 X1=XX+ZZ+P
180 Y1=YY-ZZ+Q
190 GMODE 1: MOVE X1,Y1: WRPIX
200 IF Y1=0 GOTO 220
210 GMODE 2: LINE X1,Y1-1,X1,0
220 RETURN
```

**K-1008-43M** Manual only $10 (credited toward purchase)
**k-1008-43** Complete ready to install package $495

Mastercharge and Visa accepted

Write or call today for our full line catalog describing all MTU 6502 products, including our high speed 8" Floppy Disk Controller for up to 4 megabytes of PET storage.

MTU
**Micro Technology Unlimited**
2806 Hillsborough Street
P.O. Box 12106
Raleigh, NC 27605, U.S.A.
(919) 833-1458

**NOW 80 COLUMN PETS CAN HAVE MTU HIGH RESOLUTION GRAPHICS**

3. We can now enter our matrix row by row. In response to the prompt, we follow each entry by pressing the RETURN key. The number of rows is 2 and the number of columns is 3. We follow these responses with the matrix values by row. The COMMAND? response now is 'DOM' which causes the original matrix to be displayed in row, column form.

```
************************************
*                                  *
* HOW MANY ROWS? 2                 *
* HOW MANY COLUMNS? 3              *
* M( 1 , 1 ) =? 5                  *
* M( 1 , 2 ) =? 3                  *
* M( 1 , 3 ) =? 9                  *
* M( 2 , 1 ) =? -4                 *
* M( 2 , 2 ) =? 2                  *
* M( 2 , 3 ) =? -16                *
*                                  *
*                                  *
*                                  *
*                                  *
*                                  *
* COMMAND? DOM                     *
************************************
```

4. We need a 1 in the first row, first column so our COMMAND? is *R for multiply a row. We are then prompted to enter ROW TO BE MULTIPLIED?, to which we reply 1. Next we enter 1/5 to the prompt MULTIPLIER?. Note that the entry is a common fraction. Following the last entry the screen clears and the new or current matrix is displayed with all fractions reduced.

```
************************************
*                                  *
*                                  *
*   5     3     9                  *
*                                  *
*                                  *
*                                  *
*  -4     2   -16                  *
*                                  *
*                                  *
*                                  *
*                                  *
*                                  *
* ROW TO BE MULTIPLIED? 1          *
* MULTIPLIER? 1/5                  *
* COMMAND? *R                      *
*                                  *
************************************
```

5. We now need a 0 in the second row, first column so our COMMAND? is *R + for multiply a row and add to another row. We answer ROW TO BE MULTIPLIED? with a 1, MULTIPLIER? with a 4, and ROW TO BE ADDED TO? with a 2. Those operations are performed and the result is displayed.

```
************************************
*                                  *
*         3     9                  *
*   1    ---   ---                 *
*         5     5                  *
*                                  *
*                                  *
*  -4     2   -16                  *
*                                  *
*                                  *
*                                  *
* ROW TO BE MULTIPLIED? 1          *
* MULTIPLIER? 4                    *
* ROW TO BE ADDED TO? 2            *
*                                  *
* COMMAND? *R+                     *
*                                  *
************************************
```

6. We now get a 1 in the second row, second column by responding with *R for COMMAND?, 2 for ROW TO BE MULTIPLIED?, and 5/22 for MULTIPLIER?.

```
************************************
*                                  *
*         3     9                  *
*   1    ---   ---                 *
*         5     5                  *
*                                  *
*        22   -44                  *
*   0    ---   ---                 *
*         5     5                  *
*                                  *
*                                  *
*                                  *
*                                  *
* ROW TO BE MULTIPLIED? 2          *
* MULTIPLIER? 5/22                 *
* COMMAND? *R                      *
*                                  *
************************************
```

7. Our last row operation creates a 0 in the first row, second column by responding a COMMAND? with *R +, ROW TO BE MULTIPLIED? by 2, MULTIPLIER? by −3/5, and ROW TO BE ADDED TO? by 1. The final or row-reduced matrix will then be displayed.

```
************************************
*                                  *
*         3     9                  *
*   1    ---   ---                 *
*         5     5                  *
*                                  *
*                                  *
*   0     1    -2                  *
*                                  *
*                                  *
*                                  *
* ROW TO BE MULTIPLIED? 2          *
* MULTIPLIER? -3/5                 *
* ROW TO BE ADDED TO? 1            *
*                                  *
* COMMAND? *R+                     *
*                                  *
************************************
```

8. We now interpret this row reduced matrix as a system of equations with the same solution as the original system
   x = 3 and y = −2.

```
************************************
*                                  *
*                                  *
*   1     0     3                  *
*                                  *
*                                  *
*                                  *
*   0     1    -2                  *
*                                  *
*                                  *
*                                  *
*                                  *
*                                  *
*                                  *
*                                  *
* COMMAND? Q                       *
*                                  *
************************************
```

9. Our last COMMAND? is Q for quit. This allows a clean exit from the program.

In class, we can handle systems of 5, 6, or more equations with no more difficulty than solving two on the chalkboard. The example was kept small to conserve space in the article. Please note that the input concerned strategy decisions involved in the row-reduction method and not with the details of the matrix row operations. Another nice command in the menu is B for backup one step. If we don't get the desired results at any step, B returns us to the status prior to the last command. At any step, we may display the original, current, or current-1 matrix. If an interchange of rows is desired the command IR causes it to happen.

### Mathematics Topics Taught

I have found this program useful in teaching the strategies and application of row-reduction to solve systems of linear equations, finding the inverse of any square matrix that has one, and finding the value of a determinant of a square matrix by first zeroing all entries under the diagonal and then finding the product of the diagonal entries. Discussion of the algebra of the simplex method is clarified and simplified when using this program. The simplex method is a matrix technique for the solution of linear programming problems.

### Non-Classroom Uses

Students can use this program on a PET in a learning center or library to practice and reinforce their understanding of matrix strategies. I have also found the program valuable for producing step by step handouts to supplement the examples contained in our textbook. I have been aided greatly in this last application by the KEYPRINT program written by Charles Brannon, **COMPUTE!** #7, pp. 84, 86. I LOAD and initialize KEYPRINT first, then whenever I want a copy of the current screen contents I proceed as usual.

### Structure Of The Program

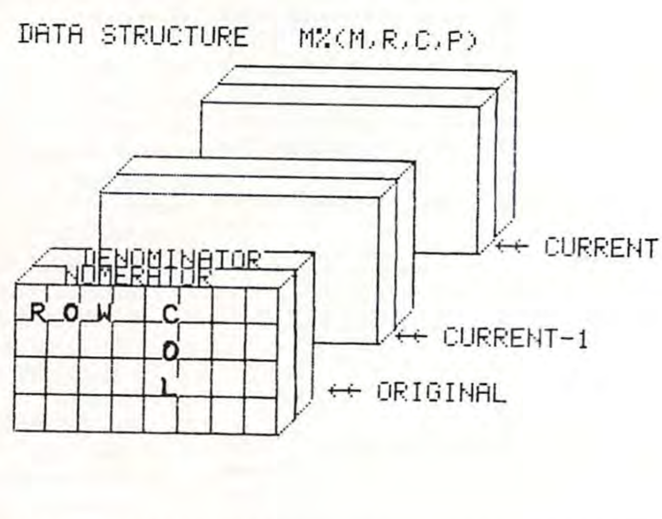
The program is a menu driven set of ten subroutines referenced from the main program by one of the menu commands. Their locations and functions are:

| COMMAND | LOCATION | FUNCTION |
|---|---|---|
| ? | 2200-2310 | Displays menu. |
| *R | 1900-1996 | Multiplies row. |
| *R + | 1700-1880 | Multiplies entries of one row and adds to another. |
| IR | 2000-2090 | Interchanges any two rows. |
| B | 2400-2430 | Backs up 1 step. Current and current-1, matrices are switches. |
| DOM | 1205-1310 | Displays original matrix. |
| DIM | 1210-1310 | Displays intermediate (current-1) matrix. |
| DCM | 1215-1310 | Displays current matrix. |
| EM | 1000-1090 | Accepts input of the original matrix. |
| Q | 2100-2120 | Clears screen and exits program. |

In addition, there are four workhorse subroutines that are called by most of the main subroutines listed above. A fraction reducer routine at 1400-1470 reduces all fractions and also serves to find least common denominators for additions. A matrix switcher at 1500-1560 is the entry routine and the *R and *R + routines. The move current to current-1 routine at 1600-1650 is used by the *R, *R +, IR, and B routines.

**DATA STRUCTURE**

****************************************




****************************************

Three matrices are stored as one four-dimensioned array M%(M,R,C,P). M is the matrix number with 1 = original, 2 = current-1, and 3 = current. R is the

# Microphys Programs

PET

Software Specialists

Science and Education

APPLE

Microphys is pleased to announce the availability of its educational software for use with the Commodore PET/CBM and Apple/Bell & Howell microcomputers. These programs have been successfully employed in **Chemistry, Physics, Calculus** and **Mathematics** classes on both the high school and college levels.

The programs are supplied on C-10 cassettes and are accompanied by complete instructions so that even those with little or no computer experience may immediately utilize the software in their classrooms. Each cassette retails for $20 and may be obtained from leading computer dealers or directly from Microphys.

Each **Physics** and **Chemistry** cassette has both a computer-assisted and individualized-instruction program recorded on opposite sides of the cassette. The **CAI program** guides the student through interacts with the computer and receives immediate evaluation of his responses and/or assistance when needed. The **I/I program** generates a unique set of problems for each student. The computer can supply answers so that the student may check his own work. If the teacher directs the computer to suppress these answers, the student completes his work at home and then feeds his results into the computer which grades his work, supplying the answers to those questions incorrectly solved by the student. **NOTE:** each time a particular program is run, a different set of numerical values is generated. In most instances, an entirely new problem is presented. The Mathematic and Calculus cassettes have only the individualized-instruction feature.

For those using disk drives, the programs have been coherently grouped and are available on diskettes. The price of each diskette is $180 which represents a considerable savings with regard to the individual cassette price.

A partial list of the programs available appears below. Please write for the Microphys Winter Catalog which describes the complete line of educational software for use on the PET/CBM and Apple/Bell & Howell microsystems.

### CALCULUS CASSETTES

PC726-Differentiation of Algebraic Functions
PC727-Maxima/Minima Problems: Part I
PC728-Maxima/Minima Problems: Part II
PC729-Relative Rate Problems: Part I
PC730-Relative Rate Problems: Part II
PC731-Integration of Algebraic Functions
PC732-Differentiation of Trigonometric Functions
PC733-Integration of Trigonometric Functions
PC734-Integration: Areas of Plane Figures
PC735-Integration: Volumes of Solids
PC736-Integration: Arc Lengths
PC737-Integration: Surface Areas of Solids

M4 **Calculus I Diskette** - contains 726 - 737

### PHYSICS AND CHEMISTRY CASSETTES

1. Linear Kinematics
2. Projectile Motion
3. Momentum and Energy
4. Energy and the Inclined Plane
5. Inelastic Collisions
6. Centripetal Force
7. Pulley Systems — Machines
8. Specific Heat Capacity
9. Calorimetry
10. Heats of Fusion/Vaporization
11. Specific Gas Laws
12. General Gas Laws
13. Thermodynamics I
14. Thermodynamics II
15. Transverse Standing Waves
16. Longitudinal Standing Waves
17. Lenses and Mirrors
18. Refraction of Light
19. Series Circuit Analysis
20. Parallel Circuit Analysis I
20A. Parallel Circuit Analysis II
21. Series Parallel Circuit Analysis
22. Faraday's Law
23. Gram-Molecular Mass
24. The Mole Concept
25. The Molarity Concept
26. The Normality Concept
27. The Molality Concept
28. Stiochiometry: Mass/Mass
29. Stoichiometry: Mass/Volume
30. Stoichiometry: Volume/Volume
31. Stoichiometry: General
32. Percent Concentration
33. pH Concept
34. EMF of Electrochemical Cells
35. Electric Field Analysis
36. Photoelectric Effect
37. Symbols and Valence Drill
38. Names of Compounds Drill
39. Formulas of Compounds Drill
40. Total Internal Reflection

P1 **Physics I Diskette** - contains the following programs
1 2 3 4 5 6 7 8 9 10 301 302 305 306

P2 **Physics II Diskette** - contains the following programs
11 12 13 14 15 16 17 18 19 20 20A 21 35 36 40 303 304

## Microphys Programs

**2048 Ford Street**
**Brooklyn, New York 11229**
**(212) 646-0140**

## GAME DESIGNERS AND PROGRAMMERS

Join the professionals who use storyboards to lay out their designs before writing their programs. The Program Concept Sheet™ separates your creative design task from the analytical task of actually writing the program. At just $4.50 per pad, this just might be the best design tool you will ever use.

## ATTENTION ATARI, APPLE, AND COMMODORE USERS

The PrestoDigitizer™ tablet is now in stock for the computer you use, be it the Atari 400 or 800, the Apple II or II+, or the Commodore CBM, PET or VIC. At $50 ($65 for the Apple version), this device is the lowest priced tool for graphics input and character recognition on the market today. The PrestoDigitizer comes complete with software and full documentation.

The Program Concept Sheet and PrestoDigitizer are products of Innovision, P.O. Box 1317, Los Altos, CA 94022, (415) 964-2885.

Dealer Inquiries Invited.

micromath

# MICROMATH revolutionizes the teaching of math!

**Grades 6 to 12**

MicroMath is a complete review mathematics course which will be of great value to students from grade 6 through 12.

**93 Lessons, 16 Tests, 150 Page Workbook**

Students can choose from 93 separate lessons (each with pre-test and lesson) and work comfortably at their own speed, with peers or individually. Each unit has its own post test and the entire package is supplemented with a 150 page workbook.

**Runs on COMMODORE PET 16K and APPLE II+**

PET version consists of 5 disks; the APPLE version has 10 disks.

**Presently in use across Canada and the U.S.**

MicroMath is the only full, one-semester course of its kind in North America and is highly praised by those instructors using it.

**$500.00 for complete package (U.S. funds)**

A demonstration disk, consisting of 8 sample lessons and 2 tests, is available for only $25.00

***For more information***

*Write Mr. F. Winter,*
**SHERIDAN COLLEGE**
*1430 Trafalgar Road,*
*Oakville, Ontario,*
*Canada L6H 2L1*

*or call (416) 845-9430.*

row number and C is the column number. P = 1 means numerator and P = 2 means denominator. The denominator for any integer has a value of 1. An integer array name (M%) was chosen to conserve memory so that this program can be effective in an 8K machine. This limits any numerator or denominator to the maximum integer size and occasionally causes overflow problems. This occurs in large matrix problems with many relatively prime factors. It happens seldom but a check has been patched in to detect this and give you the chance to back up one step and try something else.

## Summary

This program has three notable weaknesses. It has no provisions for work on matrices with decimal entries and it's use in the classroom does require a minimal amount of time and planning for equipment set-up. It also bombs occasionally as mentioned above under data structure. The strengths of this program far outweight the weaknesses and make it an effective teaching tool. It allows concentration on strategy and concepts rather than details. It handles fractions the way the students are expected to, displaying results and accepting input in a human rather than a computer way. It has saved my students and me valuable classtime and has led them to a better understanding of the ideas being taught. In addition, it can be used outside of class for a variety of applications.

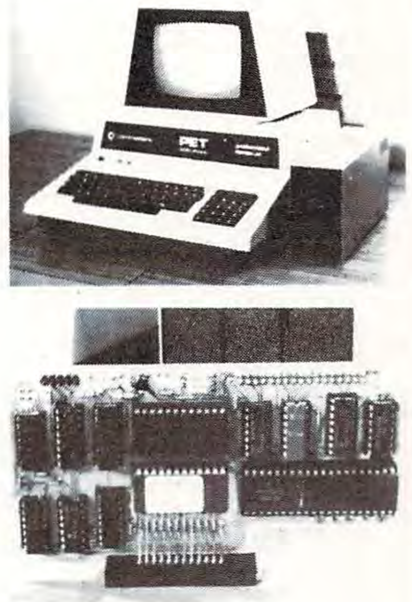
I hope other mathematics teachers will find this program as useful as I have. If you don't wish to spend your time keying the program into your machine, I will be happy to send you a copy on tape. Send a check for $10.00 and your name and complete address to William L. Hinrichs, 5056 Welsh Ct., Rockford, Ill., 61107.

```
5 REM ***MATRIX ROW OPERATIONS****
6 REM ****************************
7 REM **(C)1981 WILLIAM HINRICHS**
8 REM ****************************
9 REM *** ROCK VALLEY COLLEGE ****
10 EX=Ø
20 PRINT"ĥ↓↓↓↓↓↓↓→→→→→  MATRIX ROW ¬
      ¬OPERATIONS"
30 PRINT"↓→→→→→ CLASSROOM DEMONSTRATION"
40 PRINT"↓→→→→→(C)198Ø  WILLIAM ¬
      ¬HINRICHS"
5Ø T=TI
6Ø IF TI-T<6ØØ THEN 6Ø
7Ø PRINT"ĥ"
1ØØ GOSUB22ØØ
11Ø GOSUB9ØØ:GOTO11Ø
9ØØ REM*** COMMAND INPUT ***
91Ø DATA?,*R,*R+,IR,B,DOM,DIM,DCM,EM,Q
92Ø PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓COMMAND
      ¬";
93Ø INPUTA$
94Ø I=1
95Ø READB$
96Ø IFA$=B$THEN98Ø
965 I=I+1:IFI>1ØTHENRESTORE:GOTO92Ø
97Ø GOTO95Ø
98Ø ONIGOSUB22ØØ,19ØØ,17ØØ,2ØØØ,24ØØ,
      ¬12Ø5,121Ø,1215,1ØØØ,21ØØ
99Ø RESTORE:RETURN
1ØØØ REM*** INPUT MATRIX ***
1ØØ5 PRINT"ĥ"
1Ø1Ø INPUT"HOW MANY ROWS";NR
1Ø2Ø INPUT"HOW MANY COLUMNS";NC
1Ø3Ø DIM M%(3,NR,NC,2)
1Ø4Ø FOR R=1 TO NR
1Ø5Ø FOR C=1 TO NC
1Ø6Ø PRINT"M("R","C") =";:INPUTA$
1Ø7Ø GOSUB11ØØ
1Ø8Ø M%(1,R,C,1)=N%:M%(1,R,C,2)=D%
1Ø85 M%(2,R,C,1)=N%:M%(2,R,C,2)=D%
1Ø86 M%(3,R,C,1)=N%:M%(3,R,C,2)=D%
1Ø9Ø NEXT:NEXT:RETURN
11ØØ REM*** FRACTION SPLITTER ***
111Ø L=LEN(A$)
112Ø FOR I=1TO L
113Ø IFMID$(A$,I,1)="/"THENN%=VAL(LEFT$(
      ¬A$,I-1)):D%=VAL(RIGHT$(A$,L-I)):
      ¬I=L:K=1
114Ø NEXT I
115Ø IF K<>1THENN%=VAL(A$):D%=1
116Ø K=Ø:RETURN
12ØØ REM*** MATRIX PRINTER ***
12Ø5 M=1:GOTO1217
121Ø M=2:GOTO1217
1215 M=3
1217 PRINT"ĥ"
1218 IFNC>7THENT=4
1219 IFNC<8THENT=5
122Ø FOR R=1 TO NR
123Ø FOR C=1TONC
124Ø N%=M%(M,R,C,1):D%=M%(M,R,C,2)
1246 GOSUB14ØØ
1248 IFD%=-1THENN%=-N%:D%=1
125Ø N$=STR$(N%):LN=LEN(N$):D$=STR$(D%):
      ¬LD=LEN(D$)
126Ø IFD%=1AND(T-LN)<>ØTHENPRINT"↓"SPC(T
      ¬-LN)N$"↑";:GOTO128Ø
1262 IFD%=1AND(T-LN)=ØTHENPRINT"↓"N$"↑";
      ¬:GOTO128Ø
1265 IF(T-LN)>ØTHENPRINTSPC(T-LN);
127Ø PRINTN$"←←←↓CCC←←←←←↓";:IF(T-LD)>ØT
      ¬HENPRINTSPC(T-LD);
1275 PRINTD$"↑↑";
128Ø NEXT C
129Ø PRINT"↓↓↓"
13ØØ NEXT R
131Ø RETURN
14ØØ REM*** FRACTION REDUCER ***
14Ø5 IFN%=ØTHEND%=1:B=1:RETURN
141Ø IF N%>D% THEN A=N%:B=D%:GOTO144Ø
142Ø IF N%<D% THEN A=D%:B=N%:GOTO144Ø
143Ø IF N%=D% THENN%=1:D%=1:B=1:RETURN
144Ø Q=INT(A/B):RZ=A-Q*B
145Ø IFRZ=ØTHEN N%=N%/B:D%=D%/B:RETURN
146Ø A=B:B=RZ
147Ø GOTO 144Ø
15ØØ REM*** MATRIX SWITCHER ***
151Ø REM SWITCH MATRIX M1 WITH MATRIX M2
152Ø FOR R=1 TO NR
153Ø FOR C=1 TO NC
154Ø FOR P=1 TO 2
155Ø T%=M%(M1,R,C,P):M%(M1,R,C,P)=M%(M2,
      ¬R,C,P):M%(M2,R,C,P)=T%
156Ø NEXT P,C,R:RETURN
```

```
1600 REM*** MOVE CURRENT MATRIX TO PRIOR
1610 FORR=1TONR
1620 FORC=1TONC
1630 FORP=1TO2
1640 M%(2,R,C,P)=M%(3,R,C,P)
1650 NEXTP,C,R:RETURN
1700 REM*** R2=R1*CONSTANT+R2 ***
1725 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓"
1730 INPUT"ROW TO BE MULTIPLIED";R1
1732 IFR1<1ORR1>NRTHEN1725
1735 GOSUB1600
1740 INPUT"MULTIPLIER";A$
1750 GOSUB1100
1760 INPUT"ROW TO BE ADDED TO";R2
1770 IFR1<1ORR1>NRORR2<1ORR2>NRORR1=R2TH
     ¬EN1725
1780 GOSUB1400:NC%=N%:DC%=D%
1790 FORC=1TONC
1791 N%=NC%:D%=M%(2,R1,C,2):GOSUB1400
1792 N1%=N%:D2%=D%
1793 N%=M%(2,R1,C,1):D%=DC%:GOSUB1400
1794 N2%=N%:D1%=D%
1795 P=N1%:Q=N2%:IFABS(P*Q)>32767THEN245
     ¬0
1796 P=D1%:Q=D2%:IFABS(P*Q)>32767THEN245
     ¬0
1800 N%=N1%*N2%
1810 D%=D1%*D2%
1815 GOSUB1400
1816 N1%=N%:D1%=D%
1818 N%=D1%:D%=M%(2,R2,C,2)
1819 GOSUB 1400
1820 LC=D1%/B*M%(2,R2,C,2)
1825 N%=LC/D1%*N1%+LC/M%(2,R2,C,2)*M%(2,
     ¬R2,C,1)
1830 D%=LC
1840 GOSUB1400
1850 M%(3,R2,C,1)=N%:M%(3,R2,C,2)=D%
1860 NEXTC
1870 GOSUB1215
1880 RETURN
1900 REM*** MULTIPLY A ROW ***
1910 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓"
1930 INPUT"ROW TO BE MULTIPLIED";R1
1935 IFR1<1ORR1>NRTHEN1910
1938 GOSUB1600
1940 INPUT"MULTIPLIER";A$
1950 GOSUB1100:NC%=N%:DC%=D%
1960 FOR C=1TONC
1961 N%=NC%:D%=M%(3,R1,C,2):GOSUB1400
1962 N1%=N%:D2%=D%
1963 N%=M%(3,R1,C,1):D%=DC%:GOSUB1400
1964 N2%=N%:D1%=D%
1965 P=N1%:Q=N2%:IFABS(P*Q)>32767THEN245
     ¬0
1966 P=D1%:Q=D2%:IFABS(P*Q)>32767THEN245
     ¬0
1970 N%=N1%*N2%:D%=D1%*D2%:GOSUB1400
1990 M%(3,R1,C,1)=N%:M%(3,R1,C,2)=D%
1995 NEXTC
1996 GOSUB1215:RETURN
2000 REM *** ROW INTERCHANGE ***
2010 PRINT"h↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓"
2020 INPUT"INTERCHANGE WHICH TWO ¬
     ¬ROWS";R1,R2
2030 GOSUB1600
2040 FORC=1TONC
2050 T%=M%(3,R1,C,1):M%(3,R1,C,1)=M%(3,
     ¬R2,C,1):M%(3,R2,C,1)=T%
2060 T%=M%(3,R1,C,2):M%(3,R1,C,2)=M%(3,
     ¬R2,C,2):M%(3,R2,C,2)=T%
2070 NEXTC
2080 EX=EX+1:REM EX IS POWER OF -1 TO ¬
     ¬USE IN FINDING THE DETERMINANT
2085 GOSUB1215
2090 RETURN
2100 REM*** QUIT ***
2110 PRINT"ĥ"
2120 END
2200 REM*** DISPLAY MENU ***
2210 PRINT"ĥ     COMMAND MENU↓"
2220 PRINT"      ?   DISPLAY MENU
2230 PRINT"↓     EM   ENTER MATRIX
2240 PRINT"↓     *R   MULTIPLY ROW
2250 PRINT"↓    *R+   MULTIPLY ROW AND ¬
     ¬ADD
2260 PRINT"↓     IR   INTERCHANGE ROWS
2270 PRINT"↓    DOM   DISPLAY ORIGINAL ¬
     ¬MATRIX
2280 PRINT"↓    DIM   DISPLAY PRIOR ¬
     ¬MATRIX
2290 PRINT"↓    DCM   DISPLAY CURRENT ¬
     ¬MATRIX
2295 PRINT"↓      B   BACKUP 1 STEP
2300 PRINT"↓      Q   QUIT
2310 RETURN
2400 REM*** BACK UP 1 STEP ***
2410 M1=2:M2=3
2420 GOSUB1500:GOSUB1600:GOSUB1215
2430 RETURN
2449 REM PRODUCT TOO LARGE ERROR MESSAGE
2450 PRINT"ĥ↓↓↓↓↓↓↓↓↓↓↓↓VALUE TOO ¬
     ¬LARGE!!!":PRINT"TO CONTINUE,
     ¬ PRESS RETURN."
2460 GETA$:IFA$<>CHR$(13)THEN2460
2470 RETURN
READY.
```

©

# NEW—LOW COST DISK FOR PET CBM



PEDISK II from cgrs MICROTECH is a new floppy disk controller board for the Commodore PET-/CBM (tm) computer. PEDISK II contains control ROM and all logic required to read, write, and control up to two standard floppy disk drives (either 5.25" or 8"). The PEDISK II controller mounts inside the PET and occupies previously unused memory space. The circuit board plugs into an existing ROM socket in your machine, and is connected by cable to an external disk drive.

A complete disk system includes the PEDISK II Controller, Disk Operating System software, connecting cable, and an external disk drive set. Simply plug in the controller board, and your disk system is ready to run. Complete plug and go systems start at just $560.00

The PEDISK II system offers a lot more than low cost:

- SPEED–because data goes directly to memory, load and save are much faster than an IEEE drive.
- RELIABILITY–single chip LSI controller, simple design, no complex circuits to fail.
- COMPATABILITY–uses standard IBM disk encoding methods. Data can be transferred from non-CBM systems.

The Disk Operating System software is easy to use, and provides capability to save and load programs, along with a good file handling package.

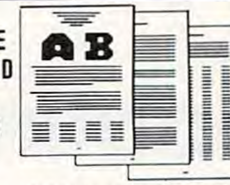
**Complete System prices with DOS and cable:**

| | |
|---|---|
| 5.25" 40 track, 1 drive, 143K | $560 |
| 5.25" 80 track, 1 drive, 286K | 690 |
| 8" IBM 3740 format, 77 track, 250K | 995 |

**Individual Component Prices:**

| | |
|---|---|
| PEDISK II Controller Card | $229 |
| DOS for PEDISK II | 75 |
| Cable assembly for 2 drives | 35 |

## PAPER-MATE 60 COMMAND WORD PROCESSOR




Paper-Mate is a full-featured word processor for $29.00 by Michael Riley. Paper-Mate incorporates 60 commands to give you full screen editing with graphics for all 16k or 32K PETs, all printers, and disk or tape drives. It also includes most features of the CBM WordPro III, plus many additional features.

For writing text, Paper-Mate has a definable keyboard so you can use either Business or Graphics machines. Shift lock on letters only, or use keyboard shift lock. All keys repeat.

Paper-Mate text editing includes floating cursor, scroll up or down, page forward or back, and repeating insert and delete keys. Text Block handling includes transfer, delete, append, save, load, and insert.

All formatting commands are imbedded in text for complete control. Commands include margin control and release, column adjust, 9 tab settings, variable line spacing, justify text, center text, and auto print form letter (variable block). Files can be linked so that one command prints an entire manuscript. Auto page, page headers, page numbers, pause at end of page, and hyphenation pauses are included.

Unlike most word processors, PET graphics as well as text can be used. Paper-Mate can send any ASC11 code over any secondary address to any printer.

Paper-Mate works on 16K or 32K PETs with any ROM, cassette or disk, and CBM or non-CBM printers. An 8K version is in the planning.

To order Paper-Mate, specify machine and ROM type.

| | |
|---|---|
| On Tape (with manual): | **$29.95** |
| On Disk (with manual): | **$32.95** |
| Manual Separate: | **$ 1.00** |

## KMMM Pascal for PET/CBM $75

A subset of standard Pascal with extensions • Machine Language Pascal Source Editor • Machine Language P-Code Compiler • P-Code Interpreter (for debugging and learning) • P-Code to machine language translator for optimized object code • Run-time package • Floating point capability • User manual and sample programs • Includes source code editor

Specify ROM version (16K minimum), disk or tape.

## EARL for PET (disk file based) $65

Editor, Assembler, Relocater, Linker. Generates relocatable object code using MOS Technology mnemonics.

Disk file input (can edit files larger than memory). Links multiple object programs as one memory load. Listing output to screen or printer. Enhanced editor operates in both command mode and cursor oriented "window" mode.

## SUPERSORT by James Strasma $35

Supersort is an excellent general purpose machine language sort routine for PET/CBM computers. Sorts both one and two dimensioned arrays at lightning speed in either ascending or descending order. Other fields can be subsorted when a match is found, and fields need not be in any special order. Sort arrays may be specified by name, and fields are random length. Allows sorting by bit to provide 8 categories per byte. The routine works with all PET BASICs, adjusts to any memory size, and can co-exist with other programs in high memory.

## FLEX-FILE

For CBM/PET Computers

Self Calculating
**DATA BASE**
REPORT WRITER
MAILING LIST

Flex File is a set of flexible, friendly programs to allow you to set up and maintain a data base as well as print files with a versatile Report Writer or a Mail Label routine. Programmers will find it easy to add subroutines to their own programs to make use of Data Base files.

### RANDOM ACCESS DATA BASE

Record size limit is 250 characters. The number of records per disk is limited only by the size of each record and the amount of free space on the disk. File maintenance lets you step forward or backward through a file, add, delete or change a record, go to a numbered record, or find a record from a specified field. The Find command locates any record when you enter all (or a portion of) the desired key field. Field lengths can vary from record to record provided the sum of the fields does not exceed the size of the record. This allows maximum packing of information. The file can be sorted by any field. Any field can be specified as a key field at any time. Sequential files from other programs can be converted to random files, and random can be converted to sequential. Maximum record size, fields per record, and order of fields can be changed at any time.

### MAILING LABELS

When record size is 127 characters (typical for mailing list), each disk can handle over 1000 records (about 2800 with the 8050 drive). Labels can be printed any number of labels across, and in any column position. Any number of fields can be printed on a label in any order, and two or three fields can be joined together on one line (like first name, last name, and title). A "type of customer" field allows selective printing.

### REPORT WRITER

The contents of any field can be placed in any column. Numerics can be decimal point justified and rounded to any accuracy. Any column can be defined as a series of mathematical functions performed on other columns. These functions may include +, −, x, /, %, and various log and trig functions. Results of operations such as running total may be passed from row to row. At the end of the report a total and/or average can be calculated for any column. Complete record selection, including field within range, pattern match, and logical functions can be specified individually or in combination with other parameters.

**Flex File was developed by Michael Riley.**

| | |
|---|---|
| **Flex File System** | **$60** |

Specify machine size (32K recommended) and ROM type for both disk and computer.

## Good things coming!

CBM VIC 20 Personal Computer with Color Graphics
CBM 8096–96K user memory space
CBM Single Disk Drive

SM-KIT–a "super toolkit" for PET/CBM 40/80 column BASIC 4.0 machines. Includes all the standard functions in much enhanced fashion, plus disk commands (including Disk Merge and DOS Support), Screenprint, etc.

## SAVE ENERGY with Commodore Programmable Thermostat

Installs easily in place of your existing unit to save up to 30% on your heating/cooling bills.

| | |
|---|---|
| **Introductory SPECIAL** | **$114** |

115 E. Stump Road
Montgomeryville, PA 18936 **215-699-5826** **A B Computers** WRITE FOR CATALOG
Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders.

# A Floating-Point Binary To BCD Routine

Marvin L. DeJong
Department of
Mathematics-Physics
The School of the Ozarks
Pt. Lookout, MO 65726

## Introduction

A previous issue of **COMPUTE!** carried a BCD to Floating-Point Binary Routine that can be used to convert a series of decimal digits and a decimal exponent to a binary number in a floating-point format. The purpose of such a routine is to enable the user to perform floating-point arithmetic. The program described in this article performs the reverse operation; that is, it converts a floating-point binary number to a decimal number and a decimal exponent. With these two routines and an Am9511 Arithmetic Processing Unit one can do most of the functions found on scientific calculators. I hope to provide a few simple arithmetic routines in the near future. In the meanwhile, you can amuse yourself by converting numbers to floating-point binary numbers and then back to decimal numbers.

## Hindsight

The BCD to floating-point binary routine described previously used a divide-by-ten routine that was part of the main program. With my excellent hindsight I now realize that the divide-by-ten routine should have been written as a *subroutine,* to

**Listing 1. A New Divide-by-Ten Routine.**

```
$0EBF 20 C5 0E  ONCMOR  JSR DIVTEN   Jump to divide-by-ten subroutine.
 0EC2 B8                CLV          Force a jump around the routine.
 0EC3 50 51             BVC ARND     The new subroutine is inserted
 0EC5 A9 00     DIVTEN  LDA $00      here. Clear accumulator for use
 0EC7 A0 28             LDY $28      as a register. Do $28 = 40 bit
 0EC9 06 00     BRA     ASL OVFLO    divide. OVFLO will be used as
 0ECB 26 04             ROL LSB      "guard" byte.
 0ECD 26 03             ROL NLSB     Roll one bit at a time into the
 0ECF 26 02             ROL NMSB     accumulator which serves to hold
 0ED1 26 01             ROL MSB      the partial dividend.
 0ED3 2A                ROL A        Check to see if A is larger than
 0ED4 C9 0A             CMP $0A      the divisor, $0A = 10.
 0ED6 90 05             BCC BRB      No. Decrease the bit counter.
 0ED8 38                SEC          Yes. Subtract divisor from A.
 0ED9 E9 0A             SBC $0A
 0EDB E6 00             INC OVFLO    Set a bit in the quotient.
 0EDD 88        BRB     DEY          Decrease the bit counter.
 0EDE D0 E9             BNE BRA
 0EE0 C6 05     BRC     DEC BEXP     Division is finished, now normalize.
 0EE2 06 00             ASL OVFLO    For each shift left, decrease the
 0EE4 26 04             ROL LSB      binary exponent.
 0EE6 26 03             ROL NLSB     Rotate the mantissa left until a
 0EE8 26 02             ROL NMSB     one is in the most-significant bit.
 0EEA 26 01             ROL MSB
 0EEC 10 F2             BPL BRC
 0EEE A5 00             LDA OVFLO    If the most-significant bit in the
 0EF0 10 12             BPL BRE      guard byte is one, round up.
 0EF2 38                SEC          Add one.
 0EF3 A2 04             LDX $04      X is byte counter.
 0EF5 B5 00     BRD     LDA ACC,X    Get the LSB.
 0EF7 69 00             ADC $00      Add the carry.
 0EF9 95 00             STA ACC,X    Result into mantissa.
 0EFB CA                DEX
 0EFC D0 F7             BNE BRD      Back to complete addition.
 0EFE 90 04             BCC BRE      No carry from MSB so finish.
 0F00 66 01             ROR MSB      A carry, put in bit seven,
 0F02 E6 05             INC BEXP     and increase the binary exponent.
 0F04 A9 00     BRE     LDA $00      Clear the OVFLO position, then
 0F06 85 00             STA OVFLO    get out.
 0F08 60                RTS
   .                     .
   .                     .           Empty memory locations here.
   .                     .
 0F16 A9 00     ARND    LDA $00      Remainder of BCD-to-floating
   .                     .
   :                     :           point routine is here.
```

**Listing 2. Modifications to the BCD-to-Floating-Point Binary Routine.**

```
$0E54 18                CLC          Clear carry for addition.
 0E55 A5 05             LDA BEXP     Get binary exponent.
 0E57 69 20             ADC $20      Add $20 = 32 to place binary
 0E59 85 05             STA BEXP     point properly.
 0E5A EA                NOP
 0E5B EA                NOP
$0D53 A0 20     BR7     LDY $20      Y will limit the number of
 0D55 A5 01     BR10    LDA MSB      left shifts to 32.
 0D57 30 0D             BMI BR11     If mantissa has a one in its
 0D59 18                CLC          most-significant bit, get out.
 0D5A A2 04             LDX $04
 0D5C 36 00     BR9     ROL ACC,X    Shift accumulator left one bit.
 0D5E CA                DEX
 0D5F D0 FB             BNE BR9
 0D61 C6 05             DEC BEXP     Decrement binary exponent for each
 0D63 88                DEY          left shift.
 0D64 D0 EF             BNE BR10     No more than $20 = 32 bits shifted.
 0D66 60        BR11    RTS          That's it.
```

be called by *both* the BCD to floating-point binary routine and the binary to decimal routine described here. So my first task was to rewrite the divide-by-ten routine as a subroutine. I also discovered that the divide-by-ten routine described in the previous article did not give sufficient precision. In any case, the divide-by-ten routine was completely revised and appears in Listing 1 in this article. It uses the location $0000, called OVFLO, as a "guard" byte to give the necessary precision. It actually starts at $0EC5, but our listing starts at $0EBF to indiciate a few changes that must be made in the original listing to insert the subroutine.

Some other minor modifications to the program are given in Listing 2. Although the BCD to Floating-Point Binary program will work without these changes, it will work better if you introduce the changes shown in Listing 2. The development of the program described in this article enabled me to find some places to improve the other routine. The modifications are simple and short.

## The Conversion Routine

The program to convert a normalized floating-point binary number and its exponent to a BCD number and then output the result is given in Listing 3. A 32-bit binary to BCD conversion subroutine is called by this program and it is found in Listing 5. A flowchart of the entire process is given in Figure 1. The normalized floating-point binary mantissa is operated on by a series of "times ten" or "divide by ten" operations until the binary point is moved from the left of the mantissa to the right of the 32 bit mantissa. In other words, we multiply by ten or divide by ten until the binary exponent is 32. Then the mantissa represents an integer and can be converted to a BCD number using the subroutine in Listing 5. The algorithm for this latter routine is from Peatman's (John B)

**Listing 3. A Floating-Point Binary to BCD Routine.**

```
$0B00 A5 01      BEGIN    LDA MSB       Test MSB to see if mantissa is zero.
 0B02 D0 0E               BNE BRT       If it is, print a zero and then
 0B04 20 9B 0F            JSR CLDISP    get out. Clear display.
 0B07 A9 30               LDA $30       Get ASCII zero.
 0B09 20 A6 0F            JSR OUTCH     Jump to output subroutine.
 0B0C A9 0D               LDA $0D       Get "carriage return."
 0B0E 20 A6 0F            JSR OUTCH     Output it.
 0B11 60                  RTS           Return to calling routine.
 0B12 A9 00      BRT      LDA $00       Clear OVFLO location.
 0B14 85 00               STA OVFLO
 0B16 A5 05      BRY      LDA BEXP      Is the binary exponent negative?
 0B18 10 0B               BPL BRZ       No.
 0B1A 20 00 0D            JSR TENX      Yes. Multiply by ten until the
 0B1D 20 30 0D            JSR NORM      exponent is not negative.
 0B20 C6 17               DEC DEXP      Decrement decimal exponent.
 0B22 B8                  CLV           Force a jump.
 0B23 50 F1      BVC BRY  Repeat.
 0B25 A5 05      BRZ      LDA BEXP      Compare the binary exponent to
 0B27 C9 20               CMP $20       $20 = 32.
 0B29 F0 48               BEQ BCD       Equal. Convert binary to BCD.
 0B2B 90 08               BCC BRX       Less than.
 0B2D 20 C5 0E            JSR DIVTEN    Greater than. Divide by ten until
 0B30 E6 17               INC DEXP      BEXP is less than 32.
 0B32 B8                  CLV           Force a jump.
 0B33 50 F0               BVC BRZ
 0B35 A9 00               LDA $00       Clear OVFLO
 0B37 85 00               STA OVFLO
 0B39 20 00 0D   BRW      JSR TENX      Multiply by ten.
 0B3C 20 30 0D            JSR NORM      Then normalize.
 0B3F C6 17               DEC DEXP      Decrement decimal exponent.
 0B41 A5 05               LDA BEXP      Test binary exponent.
 0B43 C9 20               CMP $20       Is it 32?
 0B45 F0 2C               BEQ BCD       Yes.
 0B47 90 F0               BCC BRW       It's less than 32 so multiply by 10.
 0B49 20 C5 0E            JSR DIVTEN    It's greater than 32 so divide.
 0B4C E6 17               INC DEXP      Increment decimal exponent.
 0B4E A5 05      BRU      LDA BEXP      Test binary exponent.
 0B50 C9 20               CMP $20       Compare with 32.
 0B52 F0 0F               BEQ BRV       Shift mantissa right until exponent
 0B54 46 01               LSR MSB       is 32.
 0B56 66 02               ROR NMSB
 0B58 66 03               ROR NLSB
 0B5A 66 04               ROR LSB
 0B5C 66 0B               ROR TEMP      Least-significant bit into TEMP.
 0B5E E6 05               INC BEXP      Increment exponent for each shift
 0B60 B8                  CLV           right.
 0B61 50 EB               BVC BRU
 0B63 A5 0B      BRV      LDA TEMP      Test to see if we need to round
 0B65 10 0C               BPL BCD       up. No.
 0B67 38                  SEC           Yes. Add one to mantissa.
 0B68 A2 04               LDX $04
 0B6A B5 00      BRS      LDA ACC,X
 0B6C 69 00               ADC $00
 0B6E 95 00               STA ACC,X
 0B70 CA                  DEX
 0B71 D0 F7               BNE BRS
 0B73 20 67 0D   BCD      JSR CONVD     Jump to 32 bit binary-to-BCD
                                        routine.
 0B76 A0 04      BRM      LDY $04       Rotate BCD accumulator right until
 0B78 A2 04      BRP      LDX $04       non-significant zeros are shifted
 0B7A 18                  CLC           out or DEXP is zero, whichever
 0B7B 76 20      BRQ      ROR BCDN,X    comes first.
 0B7D CA                  DEX
 0B7E 10 FB               BPL BRQ
 0B80 88                  DEY
 0B81 D0 F5               BNE BRP
 0B83 E6 17               INC DEXP      Increment exponent for each shift
 0B85 F0 06               BEQ BRO       right. Get out when DEXP = 0.
```

**Microprocessor Based Design** (McGraw-Hill).

Of course, each time the binary number is multiplied by ten or divided by ten the decimal exponent is adjusted. Thus, we are left with a BCD number in locations $0020 - $0024 (five locations for ten digits) and a decimal exponent in $0017. The rest of the routine is largely processing required to give a reasonable output format. Since we don't want to print a group of non-significant zeros, the BCD number is rotated right until all the zeros are shifted out *or* the decimal exponent is zero, whichever comes first.

Next the routine starts examining the BCD number from the left and skips any leading zeros. Thus, the first non-zero digit is the first digit printed. Of course, if the number is minus (a non-zero result in location $0007) a minus sign is printed. Next the decimal point is printed, and finally the exponent is printed in the form "E XX." Thus, the format chosen always has the decimal point to the right of the significant digits, 3148159. E-6 for example. If you want scientific notation for non-integer results you can modify the output routine. It's simply a matter of moving the decimal point. The flowchart and the comments should allow you to understand and modify the code.

```
0B87 A5 20                 LDA LBCDN      Has a non-zero digit been shifted
0B89 29 0F                 AND $0F        into the least-significant place?
0B8B F0 E9                 BEQ BRM        No. Shift another digit.
0B8D EA          BRO       NOP            Oops. These NOPs cover an
0B8E EA                    NOP            earlier mistake.
0B8F EA                    NOP
0B90 EA                    NOP
0B91 EA                    NOP
0B92 20 9B 0F              JSR CLDISP     This routine simply clears the
0B95 A5 07                 LDA MFLAG      AIM 65 20-character display.
0B97 F0 05                 BEQ BRN        If the sign of the number is minus,
0B99 A9 2D                 LDA $2D        output a minus sign first.
0B9B 20 A6 0F              JSR OUTCH      ASCII " - " = $2D. Output
                                          character.
0B9E A9 0B       BRN       LDA $0B        Set digit counter to eleven.
0BA0 85 0B                 STA TEMP
0BA2 A0 04       BRI       LDY $04        Rotate BCD accumulator left to
0BA4 18          BRH       CLC            output most-significant digits
0BA5 A2 FB                 LDX $FB        first. But first bypass zeros.
0BA7 36 25       BRG       ROL BCDN
0BA9 E8                    INX
0BAA D0 FB                 BNE BRG
0BAC 26 00                 ROL OVFLO      Rotate digit into OVFLO.
0BAE 88                     DEY
0BAF D0 F3                 BNE BRH
0BB1 C6 0B                 DEC TEMP       Decrement digit counter.
0BB3 A5 00                 LDA OVFLO      Is the rotated digit zero?
0BB5 F0 Eb                 BEQ BRI        Yes. Rotate again.
0BB7 18          BRX       CLC            Convert digit to ASCII and
0BB8 69 30                 ADC $30        output it.
0BBA 20 A6 0F              JSR OUTCH
0BBD A9 00                 LDA $00        Clear OVFLO for next digit.
0BBF 85 00                 STA OVFLO
0BC1 A0 04                 LDY $04        Output the remaining digits.
0BC3 18          BRL       CLC
0BC4 A2          $FB       LDX $FB
0BC6 36 25       BRJ       ROL BCDN,X     Rotate a digit at a time into
0BC8 E8                    INX            OVFLO, then output it. One digit
0BC9 D0 FB                 BNE BRJ        is four bits or one nibble.
0BCB 26 00                 ROL OVFLO
0BCD 88                    DEY
0BCE D0 F3                 BNE BRL
0BD0 A5 00                 LDA OVFLO      Get digit.
0BD2 C6 0B                 DEC TEMP       Decrement digit counter.
0BD4 D0 E1                 BNE BRX
0BD6 A5 17                 LDA DEXP       Is the decimal exponent zero?
0BD8 F0 48                 BEQ ARND       Yes. No need to output exponent.
0BDA A9 2E                 LDA $2E        Get ASCII decimal point.
0BDC 20 A6 0F              JSR OUTCH      Output it.
0BDF A9 45                 LDA $45        Get ASCII "E".
0BE1 20 A6 0F              JSR OUTCH
0BE4 A5 17                 LDA DEXP       Is the decimal exponent plus?
0BE6 10 0D                 BPL THERE      Yes.
0BE8 A9 2D                 LDA $2D        No. Output ASCII " - "
0BEA 20 A6 0F              JSR OUTCH
0BED A5 17                 LDA DEXP       It's minus, so complement it and
0BEF 49 FF                 EOR $FF        add one to form the twos
                                          complement.
0BF1 85 17                 STA DEXP
0BF3 E6 17                 INC DEXP
0BF5 A9 00       THERE     LDA $00        Clear OVFLO.
0BF7 85 00                 STA OVFLO
0BF9 F8                    SED            Convert exponent to BCD.
0BFA A0 08                 LDY $08
0BFC 26 17       BR1       ROL DEXP
0BFE A5 00                 LDA OVFLO
$0C00 65 00                ADC OVFLO
0C02 85 00                 STA OVFLO
0C04 88                    DEY
0C05 D0 F5                 BNE BR1
```

## MICRO MINI COMPUTER WORLD

74 Robinwood Ave.
Columbus, Ohio 43213
(614) 235-5813 or 235-6058

**PRESENTS**

### BUSINESS ENHANCEMENTS
### COMPUSERVICE BUSINESS SOFTWARE
**for**
### COMMODORE AND APPLE COMPUTER SYSTEMS

**B.E.C. #3000 Series**

| | | |
|---|---|---|
| General Ledger | (Apple II & CBM 2001/8032/2040) | $150.00 |
| Accounts Receivable | (Apple II & CBM 2001/8032/2040) | $150.00 |
| Payroll | (Apple II & CBM 2001/8032/2040) | $150.00 |
| Inventory | (Apple II & CBM 2001/8032/2040) | $150.00 |
| Mail List | (Apple II & CBM 2001/8032/2040) | $150.00 |
| Job Cost | (Apple II & CBM 2001/8032/2040) | $150.00 |
| Accounts Receivables with Order Entry | (CBM 2001/8032/2040) | $220.00 |
| ROM Control Chip required with CBM System | | $ 70.00 |

**B.E.C. #4000 Series NEW for 8032/8050**

| | | |
|---|---|---|
| General Ledger | (CBM 8032/8050) | $175.00 |
| Accounts Receivables with Order Entry | (CBM 8032/8050) | $220.00 |
| Accounts Payable | (CBM 8032/8050) | $175.00 |
| Payroll | (CBM 8032/8050) | $175.00 |
| Inventory | (CBM 8032/8050) | $150.00 |
| Mail List | (CBM 8032/8050) | $175.00 |
| Job Costing | (CBM 8032/8050) | $150.00 |
| ROM Control Chip required with CBM System | | $ 70.00 |

Special Offer, buy any two (2) of the above packages and receive a 10% discount off suggested retail. Offer expires 1 July, 1981.

Send for documentation package for $20, apply this amount towards your first purchase or return documentation in re-saleable condition for complete refund.

CALL or WRITE for MMCWI's FREE CATALOG of computer products.

B.E.C. Software is Distributed in the East by MICRO MINI COMPUTER WORLD INC.

**Dealer Inquiries Invited**

Introducing
## THE DOUGLAS POINT OF SALE SYSTEM

The DOUGLAS POINT OF SALE SYSTEM is a data processing system operating on the Commodore Business Machines 8032 or 2001 (with BASIC 4.0 ROM's installed), and the 8050 megabyte disk computing equipment. The standard business keyboard is required. The system integrates retail sales facilities and activities with the merchantile inventory to provide a complete "point-of-sale" data processing system for the retail trade establishment.

The DOUGLAS POINT OF SALE SYSTEM supports:

- Major requirements of the retail sale transaction:
  - Normal out-of-inventory sales.
  - Non-inventory sales such as installation fees.
  - Customer returns.
  - Down payments in cash or in used equipment.
  - Cash and credit transactions.
  - Provision for tax exempt customers.
  - Control of loaned inventory items.
  - Sales tax computations for sales and returns.
  - Printed receipts with company name and receipt number.
  - Capability to add personalized notes on each receipt.
- Integration of sales activity with inventory:
  - Automatic adjustment to inventory at time of sale.
  - Customer returns posted back to inventory at time of sale
  - Stock replenishment determined at time of sale.
  - Complete sales history captured during each sale.
- Automated processing of:
  - Mail lists of customer and distributor addresses.
  - Major item customer inventory.
  - Trade-in acceptance data for used inventory report.
  - The inventory receiving process and associated reports.
  - Service and maintenance contracts.
  - Daily sales activity report.
  - File purge and system backup.
  - Password security for unattended computer.
  - Pricing and addressing labels.

**DISTRIBUTED BY:**

**micro mini COMPUTER WORLD Inc.**
TOWN & COUNTRY SHOPPING CENTER

**SUGGESTED RETAIL $750.00**

**CALL OR SEND FOR MMCWI CATALOG**

**DEALER INQUIRIES INVITED**

**74 ROBINWOOD AVE. COLUMBUS,OHIO 43213**
**(614) 235-5813 (614) 235-6058**

## MICRO MINI COMPUTER WORLD

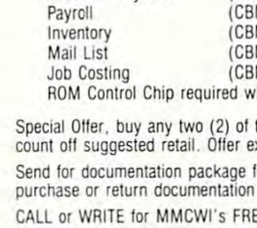
74 Robinwood Ave. Columbus, Ohio 43213
(614) 235-5813 or 235-6058

**PRESENTS**

## THE INTEGRATED COMPUTER TECHNOLOGIES

**Prioress-44 Internal Motherboard:**
The ICT P-44 is a 44 pin internal motherboard that facilitates expansion of your PET/CBM within the PET enclosure. The Prioress-44 is fully shielded on its underside by a massive ground plane. The connectors utilize any standard 44 pin edge card (many styles are available from Radio Shack). The following signals comprise the P-44 bus:

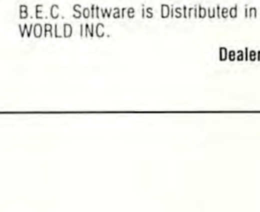
- +9v, −9v, +16v,GND,IRQ,RES,NMI,RDY,B02.
- BA0-BA15,BD0-BD7,BR/W,BW/R,SEL8,SEL9,SELA,SELB.
- DIAGNOSTIC SENSE,SYNC and 3 User definable.



The Prioress-44 is currently available for the new 2000 and 4000 series, and is under development for the 8000 series.

All ICT cards utilize the Prioress-44 bus.

Price: Prioress-44 with one connector . . . $79.00
Prioress-44 with six connectors . . . . 95.00
Each additional connector . . . . . . . . . 4.00
(specify when ordering)



**The ICT Programmable Character Generator:**
The ICT Programmable Character Generator is a 2K RAM replacement for the PET/CBM Character Generator ROM. The device allows the user to reprogram any or all of the 256 standard PET screen characters. The PCG also functions as 2K bytes of RAM in the $9000-$BFFF address range.

Uses of the ICT PCG:
a) Foreign character sets.
b) Math, Engineering and special notations.
c) Music notation.
d) Flow control and modeling.
e) Schematic and logic symbols.
f) Character oriented game symbols.
g) Architectural Drawings.
h)320Hx200V BIT GRAPHICS.
i) . . .many, many more.

The PCG has an empty socket for the original PET/CBM ROM. With the provided external switch, RAM or ROM may be selected.

ICT provides over 128K of software and data, allowing the user to immediately utilize the graphics system with extreme ease. Software is provided on 2040 format diskette and includes:

a) 7 complete 2K character sets (Russian, Katakana +).
b) Predefined graphics (including the Real-time rotating cube).
c) Development Tools including;
Charentry - used to program characters in an 8x8 matrix.
Draw - a program that allows drawing in a 320x200 area.
Plot - two versions, x,y plotting in the 320x200 matrix. A fast assembler version and a readable BASIC version.
Screen Dump - an assembler program to dump the EXACT screen contents to a Commodore 2022 printer.

Price: PCG with 2040 diskette and manual . . . . . . . . . . . . $240.00
Manual alone . . . . . . . . . . . . . . . . . . . . 7.50

**The ICT HexROM:**
A six socket programmable ROM board. Any three of the sockets may be programmed to become ROMs at $9000, $9800, $A000, $A800, $B000 and/or $B800. A simple BASIC POKE equates any socket to any of the above addresses.

Price: HexROM and manual . . . . . . . . . . . . $110.00
DumROM (6 sockets at fixed addresses) . . . . . . . . 69.00

**The ICT EPROMer:**
The EPROMer will READ/PROGRAM/VERIFY the following EPROMs:
2758, 2716, 2732 (24 pin EPROMs) and
2764, 27128 (28 pin EPROMs).
To a maximum of 36 pin I/O (5V).

The software (written in assembler) will support the above EPROM types and also allow the user to define any new EPROM configurations (5V Vcc, 25V Vpp).

Price: EPROMer, software and manual . . . . . . . . . . $180.00

ICT Products distributed by Micro Mini Computer world Inc.

***** Special introductory offer *****
**10% off all suggested retail prices**
**(offer expires July 1, 1981)** **DEALER INQUIRIES INVITED**

Integrated Computer Technologies

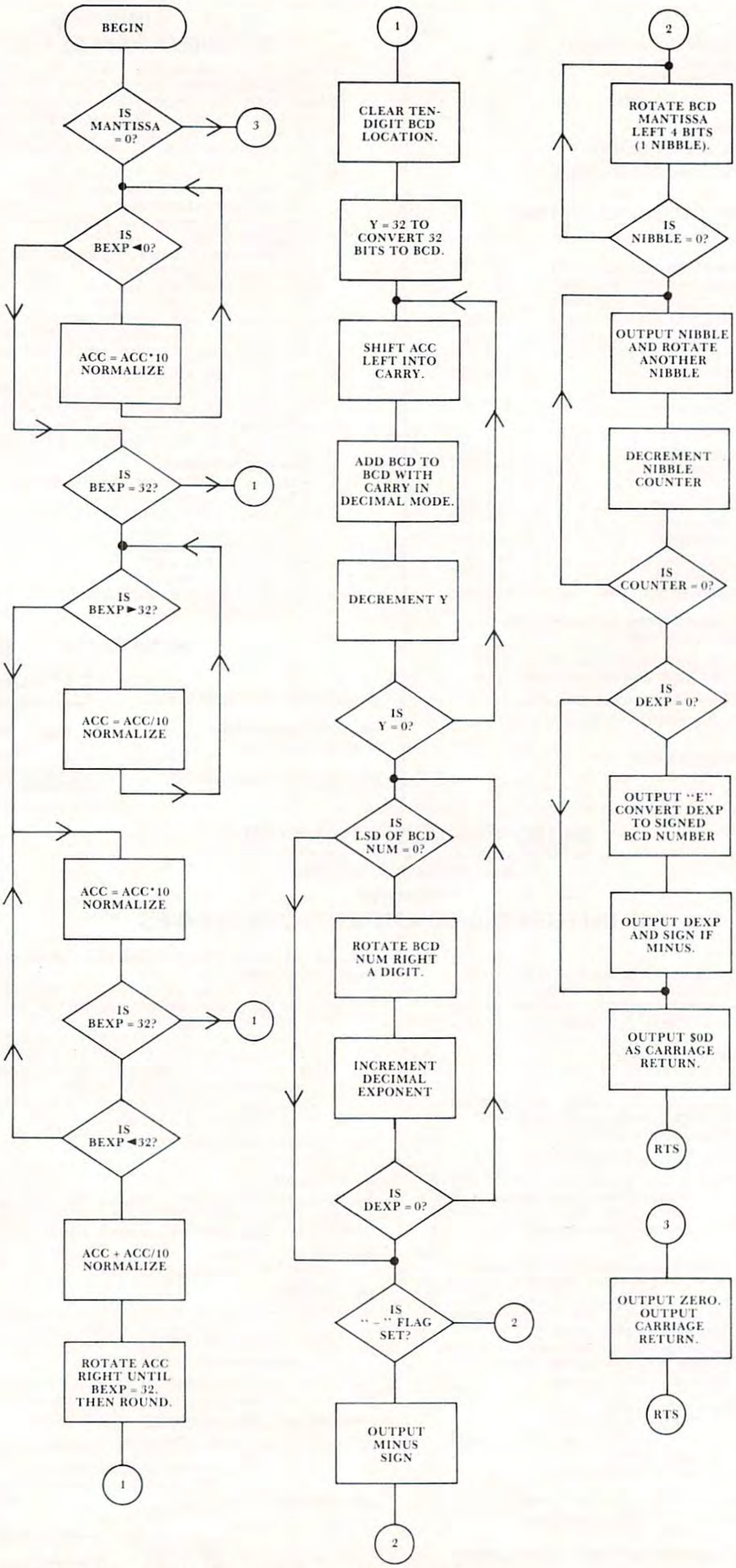



Figure 1. Flowchart of the Floating-Point Binary to BCD Routine.

```
0C07 D8                CLD
0C08 18                CLC
0C09 A5 00             LDA OVFLO     Get BCD exponent.
0C0B 29 F0             AND $F0       Mask low-order nibble (digit).
0C0D F0 09             BEQ BR2
0C0F 6A                ROR A         Rotate nibble to the right.
0C10 6A                ROR A
0C11 6A                ROR A
0C12 6A                ROR A
0C13 69 30             ADC $30       Convert to ASCII.
0C15 20 A6 0F          JSR OUTCH     Output the most-significant digit.
0C18 A5 00     BR2     LDA OVFLO     Get the least-significant digit.
0C1A 29 0F             AND $0F       Mask the high nibble.
0C1C 18                CLC
0C1D 69 30             ADC $30       Convert to ASCII.
0C1F 20 A6 0F          JSR OUTCH
0C22 A9 0D     ARND    LDA $0D       Get an ASCII carriage return.
0C24 20 A6 0F          JSR OUTCH
0C27 60                RTS           All finished.
```

Listing 4. Subroutine OUTCH For the AIM 65.

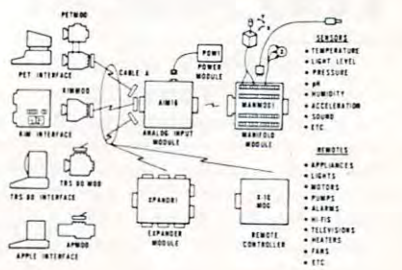
```
$0FA6 20 00 F0  OUTCH  JSR PRINT     AIM 65 monitor subroutine.
 0FA9 20 72 0F         JSR MODIFY    See previous article in COMPUTE!
 0FAC 20 60 0F         JSR DISPLAY   See previous article in COMPUTE!
 0FAF 60 RTS           RTS
```

Listing 5. A 32 Bit Binary-to-BCD Subroutine.

```
$0D67 A2 05     CONVD  LDX $05       Clear BCD accumulator.
 0D69 A9 00            LDA $00
 0D6B 95 20     BRM    STA BCDA,X    Zeros into BCD accumulator.
 0D6D CA               DEX
 0D6E 10 FB            BPL BRM
 0D70 F8               SED           Decimal mode for add.
 0D71 A0 20            LDY $20       Y has number of bits to be
 0D73 06 04     BRN    ASL LSB       converted. Rotate binary number
 0D75 26 03            ROL NLSB      into carry.
 0D77 26 02            ROL NMSB
 0D79 26 01             ROL MSB
 D7B A2 FB             LDX $FB       X will control a five byte
 0D7D B5 25     BRO    LDA BCDA,X    addition. Get least-significant
 0D7F 75 25            ADC BCDA,X    byte of the BCD accumulator,
 0D81 95 25            STA BCDA,X    add it to itself, then store.
 0D83 E8               INX           Repeat until all five bytes have
 0D84 D0 F7            BNE BRO       been added.
 0D86 88               DEY           Get another bit from the binary
 0D87 D0 EA            BNE BRN       number.
 0D89 D8               CLD           Back to binary mode.
 0D8A 60               RTS           And back to the program.
```

©

# Microcomputer Measurement And Control For PET,APPLE,KIM and AIM65






The world we live in is full of variables we want to measure. These include weight, temperature, pressure, humidity, speed and fluid level. These variables are continuous and their values may be represented by a voltage. This voltage is the analog of the physical variable. A device which converts a physical, mechanical or chemical quantity to a voltage is called a sensor.

Computers do not understand voltages: They understand bits. Bits are digital signals. A device which converts voltages to bits is an analog-to-digital converter. Our AIM 16 (Analog Input Module) is a 16 input analog-to-digital converter.

The goal of Connecticut microComputer in designing the uMAC SYSTEMS is to produce easy to use, low cost data acquisition and control modules for small computers. These acquisition and control modules will include digital input sensing (e.g. switches), analog input sensing (e.g. temperature, humidity), digital output control (e.g. lamps, motors, alarms), and analog output control (e.g. X-Y plotters, or oscilloscopes).

## Connectors

The AIM 16 requires connections to its input port (analog inputs) and its output port (computer interface). The ICON (Input CONnector) is a 20 pin, solder eyelet, edge connector for connecting inputs to each of the AIM16's 16 channels. The OCON (Output CONnector) is a 20 pin, solder eyelet edge connector for connecting the computer's input and output ports to the AIM16.

The MANMOD1 (MANifold MODule) replaces the ICON. It has screw terminals and barrier strips for all 16 inputs for connecting pots, joysticks, voltage sources, etc.

CABLE A24 (24 inch interconnect cable) has an interface connector on one end and an OCON equivalent on the other. This cable provides connections between the uMACSYSTEMS computer interfaces and the AIM 16 or XPANDR1 and between the XPANDR1 and up to eight AIM 16s.

## Analog Input Module

The AIM 16 is a 16 channel analog to digital converter designed to work with most microcomputers. The AIM 16 is connected to the host computer through the computer's 8 bit input port and 8 bit output port, or through one of the uMAC SYSTEMS special interfaces.

The input voltage range is 0 to 5.12 volts. The input voltage is converted to a count between 0 and 255 (00 and FF hex). Resolution is 20 millivolts per count. Accuracy is 0.5% ± 1 bit. Conversion time is less than 100 microseconds per channel. All 16 channels can be scanned in less than 1.5 milliseconds.

Power requirements are 12 volts DC at 60 ma.

## POW1

The POW1 is the power module for the AIM16. One POW1 supplies enough power for one AIM16, one MANMOD1, sixteen sensors, one XPANDR1 and one computer interface. The POW1 comes in an American version (POW1a) for 110 VAC and in a European version (POW1e) for 230 VAC.


## TEMPSENS

This module provides two temperature probes for use by the AIM16. This module should be used with the MANMOD1 for ease of hookup. The MANMOD1 will support up to 16 probes (eight TEMPSENS modules). Resolution for each probe is 1°F.

## Remote Controller-Clock and Calendar

**AN INEXPENSIVE CONTROL SOLUTION FOR**

**HOME SECURITY • ENERGY CONSERVATION • GREENHOUSES**
**• ENVIRONMENTAL CONTROL**
**• INDUSTRIAL CONTROL**
**• LABORATORIES**

## SUPER X-10 MOD SPECS

1. Remote controller
   Controls up to 256 different remote devices by sending signals over the house wiring to remote modules. Uses BSR remote modules available all over the USA (Sears, Radio Shack, etc.). Does not require BSR control module. Does not use sonic link.
2. Clock/calendar
   Time of day - hours, minutes, seconds
   Date - month, day - automatically corrects for 28,29,30 and 31 day months. Day of the week.
3. Digital input/outputs
   8 inputs - TTL levels or switch closures. Can be used as a trigger for a stored sequence.
   8 outputs - TTL levels
   Power supply included 110VAC only.

## XPANDR1

The XPANDR1 allows up to eight Input/Output modules to be connected to a computer at one time. The XPANDR1 is connected to the computer in place of the AIM16 or X10 MOD. Up to eight AIM16s or seven Aim 16s and one X10 MOD are then connected to each of the eight ports provided using a CABLE A24 for each module.

---

For your convenience the AIM16 and the X10 MOD come as part of a number of sets. The minimum configuration for a usable system is the AIM16 Starter Set 1 which includes one AIM16, one POW1, one ICON and one OCON. The AIM16 Starter Set 2 includes a MANMOD1 in place of the ICON. The minimum configuration for a usable system is the X10 MOD Starter Set which includes one X10 MOD, one ICON and one OCON. These sets require that you have a hardware knowledge of your computer and of computer interfacing.

For simple plug compatible systems we also offer computer interfaces and sets for many computers.

AIM16 .......... 179.00
SUPER X10 MOD (110 VAC only) .......... 249.00
POW1a (POWer module-110 VAC) .......... 14.95
POW1e (POWer module-230 VAC) .......... 24.95
ICON (Input CONnector) .......... 9.95
OCON (Output CONnector) .......... 9.95
MANMOD1 (MANifold MODule) .......... 59.95
CABLE A24 (24 inch interconnect cable) .......... 19.95
XPANDR1 (allows up to 8 Input or Output modules to be connected to a computer at one time) .......... 59.95
TEMPSENS2P1 (two temperature probes, -10°F to 160°F) .......... 69.95
LIGHTSENS1P1 (light level probe) .......... 89.95

The following sets include one AIM16, one POW1, one OCON and one ICON.

AIM16 Starter Set 1a (110 VAC) .......... 189.00
AIM16 Starter Set 1e (230 VAC) .......... 199.00

All prices and specifications subject to change without notice. Our 30-day money back guarantee applies.

The following sets include one AIM16, one POW1, one OCON and one MANMOD1.

AIM16 Starter Set 2a (110 VAC) .......... 239.00
AIM16 Starter Set 2e (230 VAC) .......... 249.00

The following modules plug into their respective computers and, when used with a CABLE A24, eliminate the need for custom wiring of the computer interface.

PETMOD (Commodore PET) .......... 49.95
KIMMOD (KIM,SYM) .......... 39.95
APMOD (APPLE II) .......... 59.95
TRS-80 MOD (Radio Shack TRS-80) .......... 59.95
AIM65 MOD (AIM 65) .......... 39.95

The following sets include one AIM16, one POW1, one MANMOD1, one CABLE A24 and one computer interface module

PETSET1a (Commodore PET - 110 VAC) .......... 295.00
PETSET1e (Commodore PET - 230 VAC) .......... 305.00
KIMSET1a (KIM,SYM,AIM65 - 110 VAC) .......... 285.00
KIMSET1e (KIM,SYM,AIM65 - 230 VAC) .......... 295.00
APSET1a(APPLE II - 110 VAC) .......... 295.00
APSET1e(APPLE II - 230 VAC) .......... 305.00
TRS-80 SET1a (Radio Shack TRS-80 - 110 VAC) .......... 295.00
TRS-80 SET1e(Radio Shack TRS-80 - 230 VAC) .......... 305.00
AIM65 SET1a(AIM65-110 VAC) .......... 285.00
AIM65 SET1e(AIM65-230 VAC) .......... 295.00

The following sets include one X10 MOD, one CABLE A24, one ICON and one computer interface module.

PETSET2(Commodore PET) .......... 295.00
KIMSET2(KIM,SYM) .......... 285.00
APSET2(APPLE II) .......... 295.00
TRS-80 SET2 (Radio Shack TRS-80) .......... 295.00
AIM65 SET2 (AIM65) .......... 285.00
SUPER X10 MOD/XPANDR1 SET2 (if you already have a SET1) .......... 295.00

# Printer And Communication Interfaces For The CBM/PET


## SADI For Serial Two-way Communication, Parallel Printers, and Serial Printers

SADI - The microprocessor based serial and parallel interface for the Commodore PET. SADI allows you to connect your PET to parallel and serial printers, CRT's, modems, acoustic couplers, hard copy terminals and other computers. The serial and parallel ports are independent allowing the PET to communicate with both peripheral devices simultaneously or one at a time. In addition, the RS-232 device can communicate with the parallel device.

Special Features for the PET interface include:
- Conversion to true ASCII both in and out
- Cursor controls and function characters specially printed
- Selectable reversal of upper and lower case
- Addressable - works with other devices

Special Features for the serial interface include:
- Baud rate selectable from 75 to 9600
- Half or full duplex
- 32 character buffer
- X-ON, X-OFF automatically sent
- Selectable carriage return delay

Special Features for the parallel interface include:
- Data strobe - either polarity
- Device ready - either polarity
- Centronics compatible

Complete with power supply, PET IEEE cable, RS-232 connector, parallel port connector and case. Assembled and tested.
SADIa (110VAC) $295
SADIe (230VAC) $325

## ADA1600 • For Parallel NEC and Centronics Standard Printers

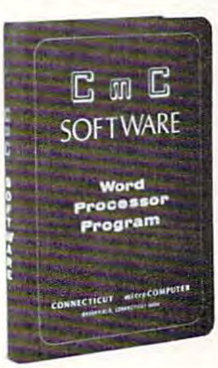
The ADA1600 is a low cost easy to use interface for the Commodore Computers. It allows the PET and CBM computers to use standard Centronics type printers (including the NEC 5530) for improved quality printing. The ADA1600 has a two foot cable which plugs into the PET IEEE port. Another IEEE card edge connector is provided for connecting disks and other peripherals to the PET. The ADA1600 is addressable and does not tie up the bus. The address is switch selectable. A four foot cable with a standard 36 pin Centronics connector is provided. A switch selects upper/lower case, upper/lower case reversed (needed for some Commodore machines) and upper case only for clearer program listings. Works with WORDPRO, BASIC and other software. No special programming is required. The case measures 3 1/2 x 5 3/4 inches. Comes complete, assembled and tested, with case and cables. Power is obtained from the printer or an external power supply may be used. Retail price for the ADA1600 is $129.

## ADA1450 • Serial Printer Adapters

The ADA1450 is a low cost, easy to use serial interface for the Commodore Computers. It allows the PET and CBM computers to use standard serial printers for improved quality printing. The ADA1450 has a two foot cable which plugs into the PET IEEE port. Another IEEE card edge connector is provided for connecting disks and other peripherals to the PET. The ADA1450 is addressable and does not tie up the bus. The address is switch selectable. A six foot RS-232 cable is provided with a DB25 connector. Pin 3 is data out. Pins 5,6 and 8 act as ready lines to the printer. Pins 4 and 20 act as ready lines from the printer. These lines can be switched for non-standard printers. Baud rate is selectable to 9600 baud. A switch selects upper/lower case, upper/lower case reversed (needed for some Commodore machines) and upper case only for clearer program listings. Works with WORDPRO, BASIC and other software. No special programming is required. The case measures 3 1/2 x 5 3/4 inches. Comes complete, assembled and tested, with case, cables, power supply and software on cassette for graphing functions, formatting data etc. The ADA1450 has a female DB25 connector at the end of the RS-232 cable for most standard printers. The ADA1450N has a male DB25 at the end of the RS-232 cable for the DIABLO serial printers. Retail price for the ADA1450 or 1450N is $149.

## ADA730 Parallel • For the Centronics 730 and 737 Printers

The ADA730 is a low cost easy to use interface for the Commodore Computers. It allows the PET and CBM computers to use Centronics type 730 and 737 printers. The ADA730 has a two foot cable which plugs into the PET IEEE port. Another IEEE card edge connector is provided for connecting disks and other peripherals to the PET. The ADA730 is addressable and does not tie up the bus. The address is switch selectable. A cable with a 36 pin card edge connector is provided. A switch selects upper/lower coase, upper/lower case reversed (needed for some Commodore machines) and upper case only for clearer program listings. Works with WORDPRO, BASIC and other software. No special programming is required. The case measures 3 1/2 x 5 3/4 inches. Comes complete, assembled and tested, with case and cables. Power is obtained from the printer or an external power supply may be used. Retail price for the ADA is $129.





## Word Processor Program •

PET Word Processor. On tape - $39.50, On disk - 49.50

For 8K Pets 29.50
For 16K and 32K Pets 39.50

Compose and print letters, flyers, ads, manuscripts, etc. Uses disk or tape. 30 page manual included.

## ADA400 RS-232 To Current Loop Adapter

RS-232 to current loop adapter (ADA 400). $29.50
Two circuits - 1 each direction. Run an RS-232 device off a computer's teletype port or vice versa. Optoisolated.

ADA400S - Solder Pads 24.50
ADA400B - Barrier Strips with screw terminals 29.50

| QUANTITY | DESCRIPTION | PRICE | TOTAL |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

SUBTOTAL
Handling and shipping — add per order $4.00
Foreign orders add 10% for AIR postage
Conn. residents add 7% sales tax
TOTAL ENCLOSED

NAME ________
COMPANY ________
ADDRESS ________
CITY ________
STATE ________ ZIP ________
VISA O M/C O Expiration date ________
Card number ________

Mention this magazine and deduct 2% from the TOTAL

**In the US order from: Connecticut MicroComputer, Inc.
34 Del Mar Drive Brookfield, CT 06804 (203) 775-4595
In Canada order from: Batteries Included, LTD
71 McCaul St. F6 Toronto, Canada M5T2X1 (416)596-1405**

**Connecticut microComputer, Inc.**
34 Del Mar Drive, Brookfield, CT 06804
203 775-4595 TWX: 710 456-0052

THE apple® GAZETTE

# Resolving Applesoft and Hires Graphics Memory Conflicts

Jeff Schmoyer

This article will attempt to divulge solutions to memory usage conflicts that can occur when an Applesoft program becomes large enough to start taking up residence in a Hires screen page. Of course the problems only appear when a program utilizing Hires graphics is executed. Throughout this article, numbers will be used in both decimal and hexadecimal (base 16). Hexadecimal numbers are represented with a dollar sign ($) preceding them, i.e. $800.

First, it is necessary to understand the memory layout of the region of RAM with which we are concerned. Applesoft programs may reside anywhere in memory from $800 to $BFFF in a 48K Apple II. If a disk is being used, the top boundary will be lower, generally $9600. The top boundary is not really important to this duscussion so it will be referred to as the top of memory. It makes no difference where it is.

The two Hires screens are in fixed positions in memory, the first located from $2000 to $3FFF, and the second from $4000 to $5FFF. Figure 1 is a map demonstrating what is known so far.

As may be seen in the drawing, if an Applesoft program is confined to the area from $800 to $1FFF, there is no conflict. This allows 6K of program space.

Now it's time to introduce another variable, variables! Not only does the program itself take up space, but as variables are allocated in the program, they too have to exist in memory.

String variables are no problem. They allocate space from the top of memory down. Plenty of unused space is available in this region.

Simple variables and numeric and string arrays on the other hand, start at the end of the program

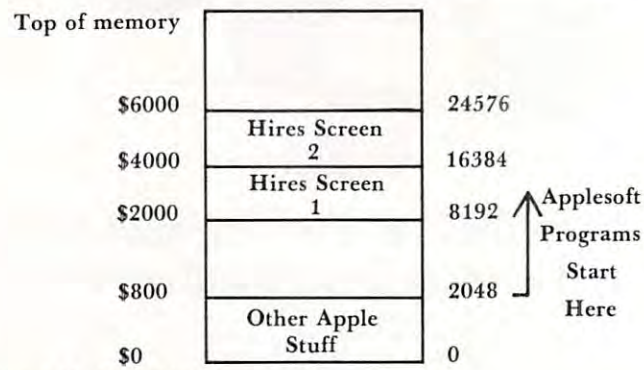



**Figure 1: Partial Memory Map**

and move up through memory. If there are enough of them, they will cruise right into the Hires page. If this happens, whenever the Hires screen is altered by an HGR or some other command, the variables in the screen space will be changed or erased!

At this point it can be seen that there is 6K of memory available for the program and simple variables and arrays altogether. Now the space is starting to get tight.

The first solution that comes to mind is simply to switch screens and use the second graphics page instead of Hires page 1. That will free up an additional 8K of memory yielding 4K total.

This is not really a bad way to go except that some Hires features are not fully supported for the second screen, such as the mixed text and graphics mode. For Hires page 2, the four lines of text at the bottom are always filled with garbage. (The lines are not actually full of garbage but that will have to be considered in a future article.) There is also the possibility of needing both Hires screens in the program for animation or some other purpose. So this solution may not be totally acceptable.

One acceptable solution deals solely with the variables. If it can be determined that the program itself does not infringe on the Hires territory, the variables may be dealt with separately. This determination may be made by loading the questionable program, entering HGR, entering TEXT, and then listing the program. (Make sure the program has been saved somewhere first.) If the end of the program is still intact, the program fits in the room available. If it is gone then move directly to the next solution. This one will not do it.

If the program fits and the variables do not, the variables may be easily moved to another region of memory. To do this, as the first line of the Applesoft program enter 0 LOMEN:16384. This line must be executed before any variables are allocated. In this

**ON-LINE** *systems*

## *IS PROUD TO ANNOUNCE A NEW ADDITION TO OUR FAMILY!*


# SuperScribe

*(Formerly SuperScript)*

**SuperScribe is the most powerful and easiest to use Word Processor available for your Apple II or II Plus computer. Besides leaping tall buildings in a single bound it:**

- Gives true upper/lower case text on your screen with no additional hardware whatsoever.
- Works with documents larger than the amount of memory in your Apple - transparently to you!
- Edit not only letters but also any text or binary file, or even basic programs!
- Automatically generates up to 4 separate indices for your document!
- Save typing time through a unique ability to designate specified keys as commonly used words, phrases or even commands!
- Globally search for or replace character strings.
- SuperScribe has a built-in instruction capability such that if you forget how to use a command and the manual is not close by - you may simply ask SuperScribe!
- Supports multiple disk drives!
- Will support alternate character sets.
- Produces form letters using address files easily!
- Supports the shift key modification if made to your Apple.
- Lets you work with your text on a screen at a time basis - reducing typos and allowing you to see your document as you edit it.
- Works with any printer!
- Supports the language card or any 16K expansion Ram card to keep more of your document readily available in memory.

**SuperScribe is 100% machine language and requires only a 48K Apple II or II Plus with a disk drive. It may be purchased through your local computer store or direct from us by sending $89.95 plus $2.50 to cover shipping to:**

**ON-LINE SYSTEMS** - 36575 Mudge Ranch Road - Coarsegold, CA 93614 - 209-683-6858

C.O.D., Master Charge or Visa accepted

way variables are stored above Hires page 1 and out of the way. If both screens usage are required, 0 LOMEN:24576 may be alternately entered to start variable allocation above screen 2. Be sure to check that the additional program line did not extend the program past the Hires boundary.

If the program itself is too large for the available space, it must be moved to a more roomy area of memory, in this case above the Hires pages. There are two page zero locations which control where an Applesoft program starts, $67 and $68. By altering these locations and reloading the program, it can be run from the new location. These alterations may be made from the direct mode or by a startup program. As long as Applesoft is not reinitialized the changes will remain in effect.

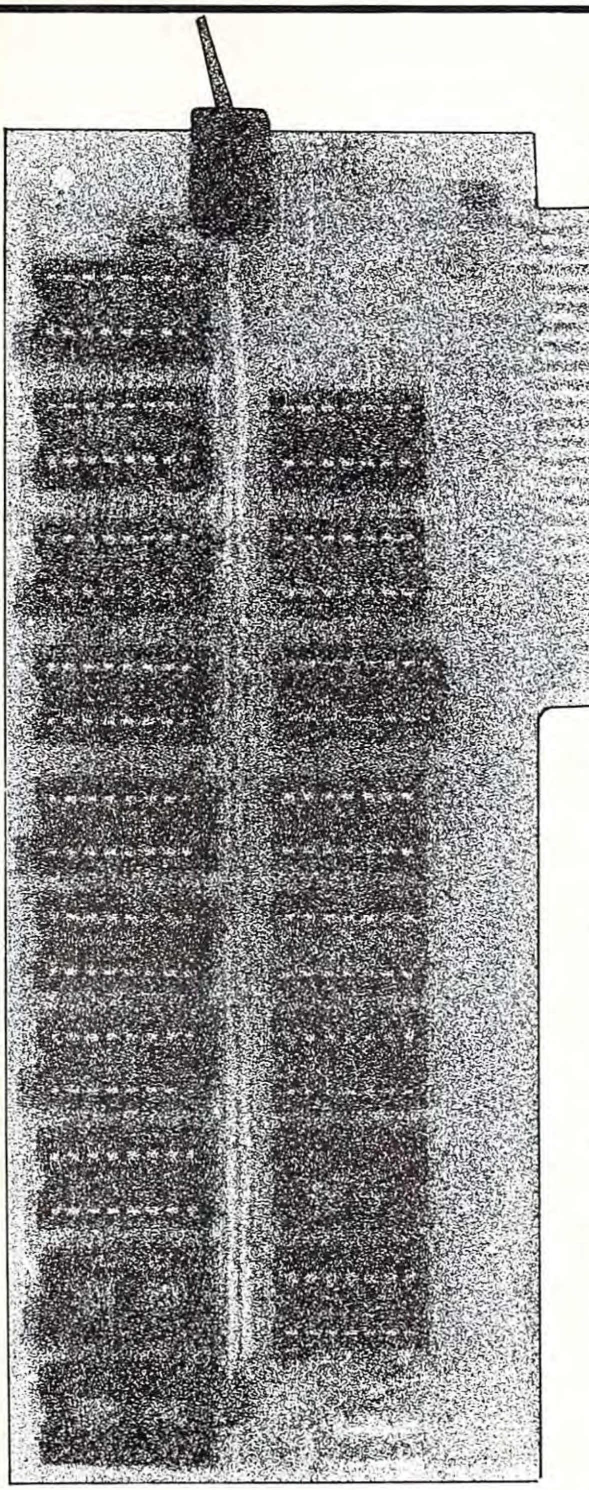
In reality only location $68 in page zero need be changed since $67 will be set to 1 in any case. The necessary poke is POKE 104,96. 104 translates to location $68, while the 96 in hex is $60. This is the high order byte of the new program starting address, $6001. Alternatively POKE 104,64 may be used to locate the program at $4001. This operation has set up the new address for the program to be loaded and run from. For the programs to execute correctly at the new location, one other poke is necessary. The location preceding the program must be set to zero. This would be $6000 for the program at $6001 or $4000 for the program at $4001. POKE 24576,0 or POKE 16384,0 respectively will accommodate this change for programs at $6001 and $4001.

After the program is moved, a 'dead zone' is left in memory from $800 to $1FFF. Neither the program nor any of its variables will use this space. Its best use would be for machine language routines and tables.

To reiterate, for a program to load and run above Hires page 1, POKE 104,64: POKE 16384,0 is necessary. For a program to load and run above Hires page 2, POKE 104,96: POKE 24576,0 is necessary.

Remember, the program will not actually be moved by this operation. Only programs loaded and run after this point will be above the Hires screen. Also, reinitialization of Applesoft will reset the pointers to $800. Setting LOMEM as with the previous technique is not necessary and should not be done.

To recap, three techniques to avoid memory conflicts with Applesoft and the Hires screens were outlined. The first is to use Hires page 2 instead of Hires page 1. The second is to move the simple variables and arrays out of the way with the LOMEM command. The third is to change the program start pointers to reset the program load and run point above the Hires pages. There are other ways to accomodate the screens but these few should suffice in most cases. ©



# Fill The Screen With Your Message: Generating Large Multi-Colored Characters Using Apple Low-Resolution Graphics

Francis A. Harvey
Rosann W. Collins
Theodore C. Hines
School of Education
University of North Carolina at Greensboro
Greensboro, North Carolina 27412

Programs written by beginning programmers can often be distinguished from more elaborate "commercial" programs by the fact that the commercial programs make such extensive use of color and graphics. Computers such as the Apple and Atari have very good graphics capability, but many users lack the time or programming background, or both, to take full advantage of these capabilities. As a result their programs, while they may be carefully designed and interesting, lack the pizzazz that children expect from computers as a result of their experiences with commercial programs and computer games at home and in game rooms.

As part of a series of utilities of this kind, we have developed a set of subroutines in Applesoft which will display the characters in any string on the screen as large, colorful letters. With these subroutines program instructions, prompts, positive reinforcement, and negative responses to user input can look just like those in "real" computer games. Very little modification of an existing program is required to convert screen output to this form.

Each character is defined (with a combination of PLOT, VLIN, and HLIN commands) on a matrix which uses seven blocks in the vertical dimension and which varies in width depending on the shape of the character. With the character set defined in this way, each line of text can have between six and nine characters, and a total of four lines of text can be displayed. Each letter is approximately one-fifth as

```
10  REM   ---- LETTER  MATCH  ---
20  REM   ---- 10/20/80 VERSION--
30  REM --------   BY   --------
40  REM ---- FRANCIS A. HARVEY -
50  REM ---- ROSANN W. COLLINS -
60  REM -- & THEODORE C. HINES -
65  REM --- COPYRIGHT 1980 -----
70  HOME : GR : GOSUB 5020
80  REM -------TITLE PAGE------
90 Y = 3:A$ = "MATCH": GOSUB 6010
100 Y = Y + 12:A$ = "THE": GOSUB 6010
110 Y = Y + 12:A$ = "LETTERS": GOSUB 6010
120  PRINT : PRINT : PRINT
130  FOR I = 1 TO 1000: NEXT I
141  PRINT "BY        FRANCIS A. HARVEY"
142  PRINT "          ROSANN W. COLLINS"
143  PRINT "          THEODORE C. HINES"
145  FOR I = 1 TO 2500: NEXT I
146  PRINT : PRINT : PRINT
160 A$ = "COPYRIGHT OCTOBER 1980": GOSUB 4020
170  FOR I = 1 TO 4000: NEXT
180  PRINT : PRINT : PRINT : REM --CLEAR TEXT
190 E = 5: REM --- FOR DEMO PURPOSES
200  REM :E IS NUMBER OF ATTEMPTS
210  REM --- USER INSTRUCTIONS---
220  GR : GOSUB 5020
230 X = 0:Y = 0: REM --RESETS LETTER POSITION
240 A$ = "I TYPE": GOSUB 6010
250 Y = Y + 12
260 A$ = "A": GOSUB 6010
270 Y = Y + 12:A$ = "LETTER.": GOSUB 6010
280  FOR I = 1 TO 5000: NEXT I: GR
290  GOSUB 5020
300 Y = 3
310 A$ = "YOU TYPE": GOSUB 6010
320 Y = Y + 9
330 A$ = "THE SAME"
340  GOSUB 6010
345 Y = Y + 9:A$ = "LETTER.": GOSUB 6000
350  FOR I = 1 TO 4000: NEXT I
360 Y = Y + 10
370 A$ = "READY?": GOSUB 6000
380  HOME
390  INPUT "STRIKE 'RETURN' WHEN READY.";A$
400  REM
410  REM *********************
420  REM ---BEGIN MAIN PROGRAM---
430 L =  RND (1) * 26 + 1
440 C1 = C1 + 1: REM --COUNTS LETTERS TRIED
450 L =  INT (L) + 64
460 Y = 3
470 A$ =  CHR$ (L): GR : GOSUB 5020: GOSUB 6010
480  HOME
490  PRINT "TYPE THE SAME LETTER."
500  GET B$
510  REM ---DISABLE RETURN KEY
520  IF  ASC (B$) = 13 THEN 500
530  REM ---DISABLE SPACE BAR
540  IF  ASC (B$) = 32 THEN 500
550 Y = Y + 8
560 A$ = B$: GOSUB 6010
570  FOR K = 1 TO 500: NEXT
580  IF B$ =  CHR$ (L) THEN  GOSUB 2010
590  IF B$ <  >  CHR$ (L) THEN  GOSUB 1010:Y = 3: GOTO 500
600  FOR I = 1 TO 2000: NEXT I
610  IF C1 < E THEN 420
620  GR : GOSUB 5000:Y = 3
630  HOME
640 A$ = "THAT'S": GOSUB 6010
650 Y = Y + 12
660 A$ = "ALL": GOSUB 6010
670 Y = Y + 12
680 A$ = "FOR NOW.": GOSUB 6010
```

# QUALITY DISK SOFTWARE

## BACKED BY ON-GOING APPLICATIONS SUPPORT

from SPECTRUM SOFTWARE

**APPLE II Ⓐ** **TRS•80 Ⓣ**

### HOME FINANCE PAK I: Entire Series $49.95 Ⓐ Ⓣ

**CHECK REGISTER AND BUDGET:** This comprehensive CHECKING ACCOUNT MANAGEMENT SYSTEM not only keeps complete records, it also gives you the analysis and control tools you need to actively manage your account. The system provides routines for BUDGETING INCOME AND EXPENSE, AUTOMATIC CHECK SEARCH, and BANK STATEMENT RECONCILING. CRT or printer reports are produced for ACTUAL EXPENSE vs BUDGET, CHECK SEARCH DISPLAY RECONCILIATION REPORT and CHECK REGISTER DISPLAY by month. Check entry is prompted by user-defined menus of standard purposes and recipient codes, speeding data entry and reducing disk storage and retrieval time. Six fields of data are stored for each check: amount, check no., date, purpose, recipient and TAX DEDUCTIBLE REMINDER. CHECK SEARCH routines allow searching on any of these data fields. Routines are also provided for CHECK SORT by date and check no., DATA EDITING and Report Formats. Up to 100 checks/mo. storage . . . . . . . . . . $39.95

**SAVINGS:** Account management system for up to 20 separate Savings accounts. Organizes, files and displays deposits, withdrawals and interest earned for each account. Complete records shown via CRT or printer . . . . . . . . . . $14.95

**CREDIT CARD:** Get Control of your credit cards with this program. Organizes, stores and displays purchases, payments and service charges for up to 20 separate cards. Use for credit cards or bank loans. CRT or printer reports . . . . . . . . . . $14.95

### UNIVERSAL COMPUTING MACHINE: $49.95 Ⓐ

A user programmable computing system structured around a 50 row x 50 column table. User defines row and column names and equations forming a unique computing machine. Table elements can be multiplied, divided, subtracted or added to any other element. User can define repeated functions common to row or column greatly simplifying table setup. Hundreds of unique computing machines can be defined, used and stored, and recalled, with or without old data, for later use. Excellent for sales forecasts, engineering design analysis, budgets, inventory lists, income statements, production planning, project cost estimates-in short for any planning, analysis or reporting problem that can by solved with a table. Unique curser commands allow you to move to any element, change its value and immediately see the effect on other table values. Entire table can be printed by machine pages (user-defined 3-5 columns) on a 40 column printer.

### COLOR CALENDAR: $29.95 Ⓐ

Got a busy calendar? Organize it with Color Calendar. Whether it's birthdays, appointments, business meetings or a regular office schedule, this program is the perfect way to schedule your activities.
The calendar display is a beautiful HI-RES color graphics calendar of the selected month with each scheduled day highlighted in color. Using the daily schedule, you can review any day of the month and schedule an event or activity in any one of 20 time slots from 8:00 A.M. to 5:30 P.M. Your description can be up to 20 characters long. The system will also print out hard copies on your minimum 40-column printer.

### BUSINESS SOFTWARE: Entire Series $159.95 Ⓐ Ⓣ

**MICROACCOUNTANT:** The ideal accounting system for small businesses. Based on classic T-accounts and double-entry booking, this efficient program provides a ledger journal for recording posting and reviewing up to 1,000 transactions per month to any one of 300 accounts. The program produces CRT and printer reports covering:

| | |
|---|---|
| Transaction Journal | Balance Sheet |
| Accounts Ledgers | Income and Expense Statement |

Includes a short primer on Financial Accounting. Requires 48K Ram . . . . . . . . . . $49.95

**UNIVERSAL BUSINESS MACHINE:** This program is designed to SIMPLIFY and SAVE TIME for the serious businessman who must periodically Analyze, Plan and Estimate. The program was created using our Universal Computing Machine and it is programmed to provide the following planning and forecasting tools.

| | |
|---|---|
| CASH FLOW ANALYSIS | SALES FORECASTER |
| PROFORMA PROFIT & LOSS | SOURCE AND USE OF FUNDS |
| PROFORMA BALANCE SHEET | JOB COST ESTIMATOR |
| REAL ESTATE INVESTMENT | INVENTORY ANALYSIS |

Price, including a copy of the Universal Computing Machine . . . . $89.95

**BUSINESS CHECK REGISTER AND BUDGET:** Our Check Register and Budget programs expanded to include up to 50 budgetable items and up to 400 checks per month. Includes bank statement reconciling and automatic check search (48K) . . . . . . . . . . $49.95

### ELECTRONICS SERIES: Entire Series $259.95

**LOGIC SIMULATOR:** SAVE TIME AND MONEY. Simulate your digital logic circuits before you build them. CMOS, TTL, or whatever, if it's digital logic, this program can handle it. The program is an interactive, menu driven, full-fledged logic simulator capable of simulating the bit-time response of a logic network to user-specified input patterns. It will handle up to 1000 gates, including NANDS, NORS, INVERTERS, FLIP-FLOPS, SHIFT REGISTERS, COUNTERS and user-defined MACROS. UP to 40 user-defined random, or binary input patterns. Simulation results displayed on CRT or printer. Accepts network descriptions from keyboard or from LOGIC DESIGNER for simulation . . . . . . . . . . $159.95 Ⓐ Ⓣ

**LOGIC DESIGNER:** Interactive HI-RES Graphics program for designing digital logic system. A menu driven series of keyboard commands allow you to draw directly on the screen up to 15 different gate types, including 10 gate shape patterns supplied with the program and 5 reserved for user specification. Standard patterns supplied are NAND, NOR, INVERTER, EX-OR, T-FLOP, JK-FLOP, D-FLOP, RS-FLOP, 4 BIT COUNTER and N-BIT SHIFT REGISTER. User interconnects gates just as you would normally draw using line graphics commands. Network descriptions for LOGIC SIMULATOR generated simultaneously with the CRT diagram being drawn. $159.95 Ⓐ

**MANUAL AND DEMO DISK:** Instruction Manual and demo disk illustrating capabilities of both programs . . . . . . . . . . $29.95

### MATHEMATICS SERIES: Entire Series $49.95 Ⓐ

**STATISTICAL ANALYSIS I:** This menu driven program performs LINEAR REGRESSION analysis, determines the mean, standard deviation and plots the frequency distribution of user-supplied data sets. Printer, Disk, I/O routines . . . . . . . . . . $19.95

**NUMERICAL ANALYSIS:** HI-RES 2-Dimensional plot of any function. Automatic scaling. At your option, the program will plot the function, plot the INTEGRAL, plot the DERIVATIVE, determine the ROOTS, MAXIMA, MINIMA, INTEGRAL VALUE . . . . . . . . . . $19.95

**MATRIX:** A general purpose, menu driven program for determining the INVERSE and DETERMINANT of any matrix, as well as the SOLUTION to any set of SIMULTANEOUS LINEAR EQUATIONS. . . . . . . . . . . $19.95

**3-D SURFACE PLOTTER:** Explore the ELEGANCE and BEAUTY of MATHEMATICS by creating HI-RES PLOTS of 3-dimensional surfaces from any 3-variable equation. Disk save and recall routines for plots. Menu driven to vary surface parameters. Hidden line or transparent plotting . . . $19.95

### ACTION ADVENTURE GAMES: Entire Series $29.95 Ⓐ

**RED BARON:** Can you outfly the RED BARON? This fast action game simulates a machine-gun DOGFIGHT between your WORLD WAR I BI-PLANE and the baron's. You can LOOP, DIVE, BANK or CLIMB-and so can the BARON. In HI-RES graphics. . . . . . . . . . . $14.95

**BATTLE OF MIDWAY:** You are in command of the U.S.S. HORNETS' DIVE-BOMBER squadron. Your targets are the Aircraft carriers, Akagi, Soryu and Kaga. You must fly your way through ZEROS and AA FIRE to make your DIVE-BOMB run. In HI-RES graphics . . . . . . . . . . $14.95

**FREE CATALOG**-All programs are supplied on disk and run on Apple II w/Disk & Applesoft ROM Card & TRS-80 Level II and require 32K RAM unless otherwise noted. Detailed instructions included. Orders shipped within 5 days. Card users include card number. Add $1.50 postage and handling with each order. California residents add 6½% sales tax. Foreign orders add $5.00 postage and handling.

SPECTRUM SOFTWARE
142 Carlow, P.O. Box 2084
Sunnyvale, CA 94087

**FOR PHONE ORDERS: (408) 738-4387**
**DEALER INQUIRIES INVITED.**

VISA master charge

high as the screen and about one-eighth of a screen wide.

Color can be set within the program or randomly selected each time a line of characters is displayed. The upper-left corner of the matrix is defined as (X,Y), and each character is "drawn" from this reference point.

Each line of characters is passed to the subroutine as the string A$. An initial value of X, the horizontal beginning point of each character, is calculated which will center the characters on the line, and the characters are drawn one at a time.

The subroutines that draw each character automatically increment the value of X the appropriate number of spaces to the right. Messages longer than one line (e.g., "You are sharp!") can be subdivided; the value of the string A$ is set to the contents of each line, and Y is incremented by at least nine before calling the subroutine which centers and plots the characters.

The sample program listed demonstrates two ways in which these techniques can be used. The program, LETTERMATCH, was developed to familiarize primary school students with the letters of the alphabet and the computer keyboard. A randomly selected letter is displayed on the screen, and the user is asked to type the same letter. The GET command is used for input and all non-letter keys, the RETURN key, and space bar are disabled.

If the student enters the wrong letter, the computer responds with a large "TRY IT AGAIN," then clears the screen of the student's response and redisplays the original letter. The student responds until the correct letter is selected. When the student does enter the correct letter, the computer responds (again, in large, multi-colored letters) with one of five randomly selected positive responses, such as "RIGHT!" or "YOU ARE SHARP!". Each student is asked to identify five letters correctly.

```
690  FOR I = 1 TO 2000: NEXT I
700  GR : GOSUB 5000: REM —CLEARS SCREEN
710 Y = 3
720 A$ = "NEXT": GOSUB 6000
730 Y = Y + 12:A$ = "PERSON,": GOSUB 6000
740 Y = Y + 12:A$ = "PLEASE.": GOSUB 6000
745  FOR I = 1 TO 2000: NEXT I
750  PRINT
760  PRINT "TYPE ";: FLASH : PRINT "S";: NORMAL : PRINT " TO STOP."
770  PRINT "STRIKE ANY KEY TO GO ON."
780  GET Z$: IF Z$ <  > "S" THEN C1 = 0: GR : GOTO 380
790  GR : GOSUB 5000:Y = 3
800  HOME
810 A$ = "OK!": GOSUB 6000
820 Y = Y + 12:A$ = "GOODBYE"
830  GOSUB 6000
840 Y = Y + 12:A$ = "FOR NOW.": GOSUB 6000
899  END
1000  REM
1010  REM ——SUBROUTINE FOR WRONG ANSWERS
1020 Y = Y + 8
1030 A$ = "TRY IT": GOSUB 6010
1040 Y = Y + 8:A$ = "AGAIN.": GOSUB 6010
1050  FOR I = 1 TO 1500: NEXT I
1060 Y = 11: GOSUB 3000
1070  RETURN
2000  REM
2010  REM ——SUBROUTINE FOR RIGHT ANSWERS
2020 Y = Y + 12
2030 M =  INT ( RND (1) * 5) + 1
2040  ON M GOTO 2050,2060,2070,2090,2100
2050 A$ = "RIGHT!": GOSUB 6000: RETURN
2060 A$ = "OK!2:GOSUB 6010: RETURN
2070 A$ = "YOU ARE": GOSUB 6010
2080 Y = Y + 8:A$ = "SHARP!": GOSUB 6010: RETURN
2090 A$ = "GREAT!2:GOSUB 6010: RETURN
2100 A$ = "SUPER!": GOSUB 6010: RETURN
2110  RETURN
3000  REM —BLANKS REST OF SCREEN
3010  COLOR= 0
3020  FOR T = Y TO 39
3030  HLIN 0,39 AT T
3040  NEXT
3050  GOSUB 5020
3060  RETURN
4000  REM
4010  REM *********************
4020  REM ——CENTERS AND PRINTS
4030  REM ——REGULAR TEXT——
4040 Z = (40 -  LEN (A$)) / 2
4050  HTAB Z: PRINT A$
4060  RETURN
5000  REM
5010  REM *********************
5020  REM —PICKS RANDOM COLOR——
5030  COLOR=  INT ( RND (1) * 15) + 1
5040  RETURN
6000  REM  **********************
6010  REM -LARGE PRINT SUBROUTINE
6020  REM ——A$ IS STRING TO———
6030  REM  ——   BE PRINTED ————
6040  REM ————CENTERS TEXT—
6050 X =  ABS (20 -  LEN (A$) * 2.5)
6060  FOR W = 1 TO  LEN (A$)
6070  IF  ASC ( MID$ (A$,W,1)) = 32 THEN X = X + 2: GOTO 6160
6080  IF  MID$ (A$,W,1) = "?" THEN  GOSUB 8010: GOTO 6160
6090  IF  MID$ (A$,W,1) = "!" THEN  GOSUB 8080: GOTO 6160
6100  IF  MID$ (A$,W,1) = "," THEN  GOSUB 8130: GOTO 6160
6110  IF  MID$ (A$,W,1) = "." THEN  GOSUB 8180: GOTO 6160
6120  IF  MID$ (A$,W,1) = "/" THEN  GOSUB 8230: GOTO 6160
6125  IF  MID$ (A$,W,1) = ";" THEN  GOSUB 8270: GOTO 6160
6130 P =  ASC ( MID$ (A$,W,1)) - 64
6140  ON P GOSUB 6200,6270,6350,6420,6490,6560,6620,6700,6750,6810,6860,69
     00,6950,7010,7070,7130,7190,7270,7360,7440,7480,7530,7590,7650,7700,7
     750
6150 X = X + 6
6160  NEXT
6170  GOSUB 5020
6180  RETURN
6200  REM ————PRINTS LETTER A
6210  PLOT X + 2,Y
6220  PLOT X + 1,Y + 1: PLOT X + 3,Y + 1
```

We have found that LETTERMATCH provides an excellent introduction to computers for quite young children. It can be modified (for example, by considerably shortening the delay loops) for use to introduce other users to computers and to sharpen keyboarding skills of users of any age.

LETTERMATCH occupies about 6400 bytes of memory. The subroutines which format and plot the characters (beginning in line 5000) occupy about 3700 bytes. Thus, any Applesoft program which leaves 3700 bytes of memory free when loaded can be modified to give large, multi-colored responses by merging these subroutines with the existing program (using the Apple RENUMBER program), making minor changes in the main program (adding GR, selecting colors, etc.), then modifying each PRINT statement to use the subroutine. For example, the line "200 PRINT "VERY GOOD!" would be changed to "200 A$ = "VERY GOOD!": GOSUB 6000."

The character set as developed includes the upper case letters A to Z and the question mark, exclamation point, comma, period, single quotation marks, and semicolon. The set could easily be expanded to include lower case letters, numerals, and other punctuation. The program randomly selects the color of each line of characters.

Copies of LETTERMATCH on diskette or cassette are available from the authors at the above address for the cost of duplication. While we reserve all commercial rights to these programs, we offer them free to any user for any noncommercial educational purpose. Other utilities of this kind which we have developed include routines for adding sound effects and music to programs, additional graphics (such as screen borders), and others. These will appear in later issues of **COMPUTE!** We hope that teachers and other computer users will find these procedures a useful addition to their program collection.

```
6230  VLIN Y + 2,Y + 6 AT X + 4
6240  HLIN X,X + 4 AT Y + 4
6250  VLIN Y + 2,Y + 6 AT X
6260  RETURN
6270  REM ----PRINTS B
6280  VLIN Y,Y + 6 AT X
6290  HLIN X,X + 2 AT Y
6300  HLIN X,X + 2 AT Y + 3
6310  HLIN X,X + 2 AT Y + 6
6320  VLIN Y + 1,Y + 2 AT X + 3
6330  VLIN Y + 4,Y + 5 AT X + 3
6340  IF X > 0 THEN X = X - 1: RETURN
6350  REM -----PRINTS C
6360  VLIN Y,Y + 6 AT X
6370  HLIN X,X + 3 AT Y + 6
6380  HLIN X,X + 3 AT Y
6390  PLOT X + 3,Y + 1
6400  PLOT X + 3,Y + 5
6410 X = X - 1: RETURN
6420  REM -----PRINTS D
6430  VLIN Y,Y + 6 AT X
6440  VLIN Y + 1,Y + 5 AT X + 3
6450  HLIN X,X + 2 AT Y
6460  HLIN X,X + 2 AT Y + 6
6470 X = X - 1: RETURN
6480  PRINT "1799": END
6490  REM --------PRINTS E
6500  VLIN Y,Y + 6 AT X
6510  HLIN X,X + 3 AT Y
6520  HLIN X,X + 2 AT Y + 3
6530  HLIN X,X + 3 AT Y + 6
6540 X = X - 1
6550  RETURN
6560  REM -------PRINTS F
6570  VLIN Y,Y + 6 AT X
6580  HLIN X,X + 3 AT Y
6590  HLIN X,X + 2 AT Y + 3
6600 X = X - 1
6610  RETURN
6620  REM ----------PRINTS G
6630  VLIN Y,Y + 6 AT X
6640  HLIN X,X + 3 AT Y + 6
6650  HLIN X,X + 3 AT Y
6660  PLOT X + 3,Y + 5: PLOT X + 3,Y + 4
6670  PLOT X + 2,Y + 4
6680 X = X - 1
6690  RETURN
6700  REM ----------PRINTS H
6710  VLIN Y,Y + 6 AT X
6720  VLIN Y,Y + 6 AT X + 3
6730  HLIN X,X + 3 AT Y + 3
6740 X = X - 1: RETURN
6750  REM ------------PRINT I
6760  HLIN X,X + 2 AT Y
6770  HLIN X,X + 2 AT Y + 6
6780  VLIN Y,Y + 6 AT X + 1
6790 X = X - 2
6800  RETURN
6810  REM -------PRINTS J
6820  HLIN X,X + 4 AT Y
6830  VLIN Y,Y + 5 AT X + 3
6840  PLOT X,Y + 5: HLIN X + 1,X + 2 AT Y + 6
6850  RETURN
6860  REM -----------PRINTS K
6870  VLIN Y,Y + 6 AT X
6880  PLOT X + 3,Y + 1: PLOT X + 2,Y + 2: PLOT X + 1,Y + 3: PLOT X + 1,Y +
    4: PLOT X + 2,Y + 5: PLOT X + 3,Y + 6
6890 X = X - 1: RETURN
6900  REM ------------PRINTS L
6910  VLIN Y,Y + 6 AT X
6920  HLIN X,X + 3 AT Y + 6
6930 X = X - 1: RETURN
6940  RETURN
6950  REM ------------PRINTS M
6960  VLIN Y,Y + 6 AT X
6970  VLIN Y,Y + 6 AT X + 4
6980  PLOT X + 1,Y + 1: PLOT X + 3,Y + 1
6990  PLOT X + 2,Y + 2
7000  RETURN
7010  REM ------------PRINTS N
7020  VLIN Y,Y + 6 AT X
7030  VLIN Y,Y + 6 AT X + 4
7040  PLOT X + 1,Y + 1: PLOT X + 2,Y + 2: PLOT X + 2,Y + 3: PLOT X + 3,Y +
    4
```

```
7050  PLOT X + 3,Y + 5
7060  RETURN
7070  REM ---------------PRINTS O
7080  VLIN Y,Y + 6 AT X
7090  HLIN X,X + 3 AT Y
7100  HLIN X,X + 3 AT Y + 6
7110  VLIN Y,Y + 6 AT X + 3
7120 X = X - 1: RETURN
7130  REM ---------------PRINTS P
7140  VLIN Y,Y + 6 AT X
7150  HLIN X,X + 3 AT Y
7160  HLIN X,X + 3 AT Y + 3
7170  VLIN Y,Y + 3 AT X + 3
7180 X = X - 1: RETURN
7190  REM ---------------PRINTS Q
7200  VLIN Y,Y + 6 AT X
7210  VLIN Y,Y + 6 AT X + 4
7220  HLIN X,X + 4 AT Y
7230  HLIN X,X + 4 AT Y + 6
7240  PLOT X + 3,Y + 5: PLOT X + 2,Y + 4
7250  PLOT X + 2,Y + 4
7260  RETURN
7270  REM ---------------PRINTS R
7280  VLIN Y,Y + 6 AT X
7290  HLIN X,X + 3 AT Y
7300  HLIN X,X + 3 AT Y + 3
7310  VLIN Y,Y + 3 AT X + 3
7320  PLOT X + 1,Y + 4: PLOT X + 2,Y + 5
7330  PLOT X + 3,Y + 6
7340  IF X > 1 THEN X = X - 1
7350  RETURN
7360  REM ----------------PRINTS S
7370  HLIN X,X + 3 AT Y
7380  HLIN X,X + 3 AT Y + 6
7390  HLIN X,X + 3 AT Y + 3
7400  VLIN Y,Y + 3 AT X
7410  VLIN Y + 3,Y + 6 AT X + 3
7420 X = X - 1
7430  RETURN
7440  REM ---------------PRINTS T
7450  VLIN Y,Y + 6 AT X + 2
7460  HLIN X,X + 4 AT Y
7470  RETURN
7480  REM ----------------PRINTS U
7490  VLIN Y,Y + 6 AT X
7500  VLIN Y,Y + 6 AT X + 3
7510  HLIN X,X + 3 AT Y + 6
7520 X = X - 1: RETURN
7530  REM ----------------PRINTS V
7540  VLIN Y,Y + 3 AT X
7550  VLIN Y,Y + 3 AT X + 4
7560  VLIN Y + 4,Y + 5 AT X + 1: VLIN Y + 4,Y + 5 AT X + 3
7570  PLOT X + 2,Y + 6
7580  RETURN
7590  REM ---------------PRINTS W
7600  VLIN Y,Y + 5 AT X
7610  VLIN Y,Y + 5 AT X + 4
7620  PLOT X + 1,Y + 6: PLOT X + 3,Y + 6
7630  VLIN Y + 4,Y + 5 AT X + 2
7640  RETURN
7650  REM ---------------PRINTS X
7660  VLIN Y,Y + 1 AT X: VLIN Y,Y + 1 AT X + 4
7670  VLIN Y + 5,Y + 6 AT X: VLIN Y + 5,Y + 6 AT X + 4
7680  PLOT X + 1,Y + 2: PLOT X + 3,Y + 2: PLOT X + 1,Y + 4: PLOT X +
     4: PLOT X + 2,Y + 3
7690  RETURN
7700  REM ---------------PRINTS Y
7710  VLIN Y,Y + 1 AT X: VLIN Y,Y + 1 AT X + 4
7720  PLOT X + 1,Y + 2: PLOT X + 3,Y + 2
7730  VLIN Y + 3,Y + 6 AT X + 2
7740  RETURN
7750  REM ----------------PRINTS Z
7760  HLIN X,X + 4 AT Y
7770  HLIN X,X + 4 AT Y + 6
7780  PLOT X + 4,Y + 1: PLOT X + 3,Y + 2: PLOT X + 2,Y + 3: PLOT X +
    4: VLIN Y + 5,Y + 6 AT X
8000  REM --PUNCTUATION ROUTINES-
8010  REM ---- QUESTION MARK---
8020  PLOT X,Y + 1: HLIN X + 1,X + 3 AT Y
8030  VLIN Y + 1,Y + 2 AT X + 4
8040  HLIN X + 2,X + 3 AT Y + 3
8050  PLOT X + 2,Y + 4: PLOT X + 2,Y + 6
8060 X = X + 6
```



```
8070  RETURN
8080  REM --- EXCLAMATION POINT
8090  VLIN Y,Y + 4 AT X + 1
8100  PLOT X + 1,Y + 6
8110 X = X + 3
8120  RETURN
8130  REM ----- COMMA -----
8140  VLIN Y + 5,Y + 6 AT X
8150  VLIN Y + 5,Y + 7 AT X + 1
8160 X = X + 4
8170  RETURN
8180  REM ----- PERIOD -----
8190  PLOT X,Y + 6
8200 X = X + 3
8210  RETURN
8220  RETURN
8230  REM ---SINGLE QUOTES---
8240  VLIN Y,Y + 1 AT X
8250 X = X + 2: RETURN
8260  END
8270  REM -----SEMICOLON-----
8280  VLIN Y + 2,Y + 3 AT X
8290  VLIN Y + 2,Y + 3 AT X + 1
8300  VLIN Y + 5,Y + 6 AT X
8310  VLIN Y + 5,Y + 7 AT X + 1
8320 X = X + 4
8330  RETURN
8340  END
```

©

# Decrementing The For... Next & Endless Loops

Derek Kelly

Scrolling text *backward* in a RUNning program can be programmed almost as easily as scrolling text forward. Scrolling text forward can be easily accomplished, as can be seen in the program in figure 1:

```
10 For I = 1 to 20
20 Print A$(I)
30 Get G$
40 Next I.
```

The GET command in Applesoft BASIC allows a pause-until-any-key-is-hit way for a user to control the progress of a program. Here, after every string printed (A$(I), a pause is inserted. After each pause, the printing of strings will proceed in a sequential manner, incrementing the counter for 'I' by 1 each time. This is forward scrolling.

But what if you want to go backward? Suppose you want the previous string by a count of 1, 2, to the first printed string? Can you go backward? The answer is Yes, and it's simple to accomplish. But like the various problems associated with For... Next loops generally, if not properly constructed, e.g., with no GOTO's that branch away totally from the loop, is that For... Next loops can involve endless loops, whether such loops run forward or backward.

In any well-constructed rogram module, there should be only one point of entrance to that module, and only one real exit. Some programmers who use GOTO, and related statements, have been known to construct programs where *under normal use* the program will "hang" somewhere, a somewhere often caused by a GOTO or a number of GOTO's which send the program around in circles. For example, consider programs A and B below. A has one entrance and exit, B has many:

```
10 GOSUB 100
 .
 .
100 For I = 1 to 230: Print "!";: GOSUB 5
110 GOSUB 200: GOSUB 300
120 NEXT I
130 RETURN
```

**Program A**

```
10 For I = 1 to 33
20 If I = Int(I/11) Then GOTO 101
30 GET G$: If G$ = "?" Then GOTO
   163
 .
 .
 .
163 Print " = ": GOTO 194
101 For J = 1 To 3: If I = J
    Then GOTO 121: Next J
102 GOTO 30
```

**Program B**

Program A is called by a GOSUB from line 10. Program A does one set of functions in the program as a whole. When called, it does what it's supposed to, and then returns. Program B, on the other hand, GOTO's all over the place, and one can't predict that any one GOTO will even find its way back to the place that started the GOingTO process. Under normal conditions, and presupposing that the GOSUBs called by Program A are well-constructed, then we can reasonably predict that program A will not cause an endless loop, while Program B most certainly *may* contain an endless loop or two that will "hang" your computer.

The two short programs above affect the forward running of For... Next loops. If a user desires to decrement rather than increment a For... Next loop so as to reverse the order of a printout, and be able to review previously printed items, then other endless loop possibilities arise.

Take Program A as an example. I might want to add a pause to the program, so I add III Get G$: If G$ = "?" Then GOTO 130. This GOTO is OK because it takes place *within* the routine.

Now I want to add a provision to reverse the loop. Since the "NEXT I" statement increments a counter by 1 (or whatever), to reverse print, I must subtract 2 from the I counter to get the next previous record. So my program now looks like PROGRAM C:

```
100 For I = 1 to 230: Print "!";: GOSUB 5
110 GOSUB200 : GOSUB300)
111 Get G$: If G$ = "-" Then I = I-2
120 NEXT I
130 RETURN
```

**Program C**

As soon as Program C is RUN, you will see that it decrements to 1 then to -1 and so on. You don't want negative numbers, so another line of code will have to be added. In addition, when the counter gets to '-1', the computer — at least *my* APPLE — hangs and prints out a steady, unstoppable — except by RESET, stream of 1's... 1.1.1.1.1.1.1.1.1.1.1....
This problem can be avoided by changing line 111 to read:

111 Get G$: IF G$ = "-" Then I = I-2: If

I< =0 Then I=0 If I=0 or less than zero, then the first item will be printed, as I will =0 *before* the execution of the NEXT which will increment I by 1 (or more) to 1.

The simple program C, if improved to include the line 111 above, will be able to process the forward and backward scrolling in any program in which it is called.

In general, there is no formula that can guarantee that a program has no endless loops. It is always possible for some bug such as an endless loop to exist under certain conditions in any program. Since you can't be sure about the absence of bugs, you have to make do with the presence of controls to limit the possibility of harmful bugs. ©

# HELP WANTED

## Apple Authors

COMPUTE! is looking for applications articles aimed at beginners and intermediate programmers. We're specifically interested in programming hints, tutorials, articles written to help users get more out of their machine. We're not afraid of re-inventing the wheel (those useful programming tricks you learned two years ago are just as relevent to a beginner now), but we do insist on high quality, pertinant information. We pay on a published page basis, and that includes program listings as well. Here are our brief guidelines: Text should be typed in upper/lower case, double spaced, with your name and name of the article appearing at the top of each page. Program listings should be single spaced, with a new ribbon on your printer.

## Apple User Groups

Please take the time to announce this at your next meeting. COMPUTE! is serious about supporting the Apple with the same high quality information we provide for other 6502 machines. To do that, we need your support as well.

**Mailing address: COMPUTE!**
Post Office Box 5406
Greensboro, NC 27403 USA
**Telephone: (919) 275-9809**

## Got A Question You'd Like Answered?

Write: Ask The Readers,
c/o **COMPUTE!**,
P.O. Box 5406,
Greensboro, NC 27403

THE ATARI® GAZETTE

# Super Cube

Steve Steinberg
Washington, DC

If you own an Atari and like to fiddle with graphics you're going to love this one. It's a program that allows you to draw cubes in different sizes and colors and also to use the cube itself as a cursor to make three dimensional bars, columns, and drawings reminiscent of the works of Escher. It can create some of the most impressive graphics displays you've yet seen on your monitor, and, to crow just a bit, it *can't* be done on an Apple.

Now, after all that buildup, a bit of mea culpa. While this program proves that a relative novice (never touched a computer until I bought my Atari last year) can make things happen on a home computer, the program is really nothing more than a synthesis of other people's work.

It all began when I started fooling around with the brilliant little program by David D. Thornburg that appeared in the May/June 1980 issue of **COMPUTE!** Mr. Thornburg's program enables you to draw a shaded three dimensional cube and change the colors by use of the keyboard. While playing with that one, Ray Daly of the Program Store in Washington, D.C., was kind enough to give me a program he wrote that puts a cursor on the screen and allows you to draw with a joystick. My question: why not change Ray's PLOT X,Y to GOSUB to a subroutine that will draw with a cube instead of with a point? My solution, with a few minor embellishments: to take David Thornburg's DRAW sequence and do just that.

Sound simple? It was, almost unbelievably so. Since then, I've been having a ball creating skyscrapers, harbors, cubist drawings and the like, and while I plead guilty to the charge of total plagiarism, it would be a shame not to share my fun with other Atari owners. Sorry, I can't tell you exactly how Ray's program works. I never really had to know. I can tell you that lines 600 to 670 set up the colors for the cube (sorry about that too, you can only get about 90 different color combinations!). Lines 720 to 830 are the cube drawing itself, with the input variable SQ setting the size. As far as the joystick commands are concerned, check them out in Atari's Basic Reference Manual; it's fairly easy to figure out what they do.

Now, for operating the program. The first thing the computer will ask you for is a dimension for the cube. Actually, you can enter numbers as large as 40 to 45, but this will cause problems if the cube or your drawing run off the screen, so until you've had a little practice, stick to dimension sizes between 1 and 10. Once you've entered your cube dimension, press RETURN and your Atari will go to the Graphics 7 mode.

The cursor is in the upper left hand corner of the screen. It may be a little difficult to see if you have a dark initial color, but you can change the display colors at any time by pushing the joystick button when the cursor is anywhere at the far left of the screen.

Next, using your joystick, move the cursor to wherever you want to begin your work of art and push the joystick button. Voila, a cube!

Whenever you let go of the joystick button, the cursor returns to the upper left hand corner of the screen. This keeps it out of the way of your drawing, since the cursor draws an erase line as it moves along. If you want to draw a bar or a column, or draw diagonally, hold the joystick button down while moving the cursor.

When you want to draw with a different sized cube, move the cursor to the bottom of the screen and press the joystick button. In the text window at the bottom of the screen, you'll be asked for a new dimension for the cube. Enter it, hit RETURN, and the cursor goes back to the upper left hand corner.

Certain three dimensional effects require that you go over what you have drawn. For example, to make a square shaped "O": Give yourself a cube dimension of, say, 6. Start somewhere near the center of the screen. Go up, straight up, about ten increments, across to the right another ten, down ten, and then draw carefully to the left, stopping when the left hand side of the drawing is just joined. Now, go back across to the right, stopping at the end of your original bottom bar line, and draw back up to the top.

Again, any time you want to change colors (it's more fun when whatever you're drawing is complete) either release the joystick button so that the cursor goes back to the upper left hand corner or just move the cursor to the far left of the screen and press the joystick button. Every now and then when you change colors, a cube will appear at the far left. If anyone figures out why this happens, please let me know.

# LE SHHHSTICK

Finally! The ultimate goal in microcomputer hardware, achieved.

A joystick with such remarkable realizm and control that even the thought of other joysticks virtually vanish.

Le Stick was born from a design used in airforce test programs. Inside are motion detectors which will manuever your sights in any direction you want with simple one handed movements. The large red push button on top provides a quick and accurate firing mechanism for better response time.

From star raiders to graphic draw programs — Le Stick eliminates all the frustrations you experience with the conventional two handed joysticks or keyboards.

The joystick of the future. See and test it at your Datasoft™ dealer today, or contact us for more information.

**Atari Le Stick $39.95**

**Datasoft Inc.**™
COMPUTER SOFTWARE
16606 Schoenborn Street • Sepulveda, California 91343
(213) 894-9154

**Atari Character Generator**
This new program for the Atari Computer enables you to create high resolution graphics in the text mode. Redefines a character set virtually any way you want. Ready for whatever language or graphics set you can devise — from ancient Japanese to Modern Russian. **(Diskette) $19.95 / (Cassette) $15.95.**

**Atari Mailing List**
Business application software for the Atari Computer. A versatile disc based system. An efficient micro format equal to those used for many large computers. Sorts by Zip Codes, Names, and stores special data for your personal use. **(Diskette) $24.95.**

This program is on the border line between 8K and 16K, so if you have only an 8K computer you will probably want to pack some of the program lines together, and eliminate the first two print statements. I've checked it out with only 8K in my Atari, and I can assure you it will work.

The program also provides a dramatic demonstration of how much memory graphics require. Try this: load the program, type RUN, then BREAK, and then PRINT FRE(0) and RETURN, which will tell you how much memory you have left. Now RUN the program again, enter an initial cube dimension, and when the program goes to Graphics Mode 7, type BREAK and then PRINT FRE(0), and RETURN again. Where did it all go?

One final note: As I've mentioned, I'm a novice at all this and Super Cube could certainly use some additional improvement. For example, how about using a second joystick to draw pyramids?

Or controlling the colors with the keyboard instead of making them random? I'd be pleased to hear from anyone who has any further ideas to offer. In the meantime, Picasso! Move over! A new generation of cubists is about to begin work!

*Editor's Note: As you update Steve's program, send in your enhancements and we'll keep you posted. RCL*

```
10 ? "}":POSITION 10,5:? "3-D DRAWING":?
 :? :? "PRESS JOYSTICK BUTTON TO DRAW"
20 PRINT :PRINT :PRINT "TO CHANGE CUBE D
IMENSION MOVE  CURSOR TO BOTTOM OF SCREE
N AND PRESS BUTTON"
30 ? :? :? "ENTER CUBE DIMENSION, THEN H
IT RETURN:(Note: Dimensions larger than
10 may  cause Errors)":INPUT SQ
40 GRAPHICS 7
50 PRINT "TO CHANGE COLORS PRESS BUTTON
WHEN    CURSOR IS AT EXTREME LEFT OF SCR
EEN"
70 GOSUB 600
80 IF STRIG(0)<>0 THEN GOSUB 500:GOTO 14
0
90 IF Y>75 THEN PRINT "ENTER NEW DIMENSI
ON FOR CUBE";:INPUT SQ:X=0:Y=0
120 POKE 77,0
130 IF STRIG(0)=0 THEN GOSUB 700
140 GOSUB 1000
150 X=X+XDIF:Y=Y+YDIF
200 IF X>143 THEN X=143:GOTO 300
210 IF Y>80 THEN Y=80:GOTO 300
300 IF X<0 THEN X=0:GOTO 400
310 IF Y<7 THEN Y=7
400 GOTO 80
500 COLOR 1:PLOT X,Y:FOR I=1 TO 5:NEXT I
:COLOR 4:PLOT X,Y:RETURN
600 A=INT(RND(1)*15)+1
610 B=INT(RND(1)*14)+2
620 IF B<4 THEN B=10
630 SETCOLOR 1,A,B
640 SETCOLOR 2,A,B-2
650 SETCOLOR 0,A,B-4
660 IF X<10 THEN GOTO 120
670 RETURN
700 IF X=0 THEN IF STRIG(0)=0 THEN GOSUB
 600
710 TRAP 80
720 COLOR 1
730 FOR I=0 TO SQ
740 PLOT X,Y+I:DRAWTO X+SQ,Y+I
750 NEXT I
760 COLOR 2
770 FOR I=1 TO INT(3*SQ)/5
780 PLOT X+I,Y-I:DRAWTO X+I+SQ,Y-I
790 NEXT I
800 COLOR 3
810 FOR I=1 TO INT(3*SQ)/5
820 PLOT X+SQ+I,Y-I:DRAWTO X+SQ+I,Y+SQ-I
+1
830 NEXT I
840 IF STRIG(0)<>0 THEN X=0:Y=0
850 RETURN
1000 WHAT=STICK(0):XDIF=0:YDIF=0
1100 IF WHAT=15 THEN RETURN
1110 IF WHAT=14 THEN YDIF=-1:RETURN
1120 IF WHAT=13 THEN YDIF=1:RETURN
1130 IF WHAT=11 THEN XDIF=-1:RETURN
1140 IF WHAT=10 THEN XDIF=-1:YDIF=-1:RET
URN
1150 IF WHAT=9 THEN YDIF=1:XDIF=-1:RETUR
N
1160 IF WHAT=7 THEN XDIF=1:RETURN
1170 IF WHAT=6 THEN YDIF=-1:XDIF=1:RETUR
N
1180 IF WHAT=5 THEN XDIF=1:YDIF=1:RETURN
```

©

# Printing Characters In Mixed Atari Graphics Modes

Craig Patchett
2 Swan Terrace
Greenwich, CT 06830

For those of you who have been anxiously awaiting this appendum, I apologize. Time conflicts and the discovery of redefinable character sets have prohibited me (until now, of course) from writing it. For those of you who haven't, get hold of the Sept/Oct 1980 issue of COMPUTE! and read up on ''Designing Your Own Atari Graphics Modes.'' This article won't be of much use to you until you do.

The problem, if you recall, is with printing characters on mode lines that are out of the usual range of that mode. For example, if we design a graphics mode such that the thirtieth line is mode two, we would get an error message if we attempted to print on that line. This is because the Atari thinks it is in the regular mode two, which only allows twelve lines of characters. We must therefore find another way to put the characters on the screen.

As you may already realize, the screen is just a type of window looking into a part of memory. If you change that memory, what you see on the screen also changes. The solution, therefore, is just to POKE the characters into the memory locations that correspond to the positions on the screen where we want them to appear.

### Where is the screen in memory?

We already know where the display list is in memory; we used the variable BEGIN to point to it last time:

```
BEGIN = PEEK(560) + PEEK(561)*256 + 4
```

But, you may well ask, what does this have to do with the screen memory, or display memory as we will call it here? It just so happens that the first two memory locations in the display list point to the beginning of display memory in the following fashion:

```
DISMEN = PEEK(BEGIN) + PEEK(BEGIN + 1)*256
```

If you recall, we never used the first two memory locations in the display list last time; now you know why.

### How do we calculate the exact memory locations to POKE into?

Each mode line used up a certain amount of memory. As you might guess, different modes use different amounts of memory per line. To be more exact:

| MODE | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| MEM/LINE | 40 | 20 | 20 | 10 | 10 | 20 | 20 | 40 | 40 |

So all we have to do is figure out how much memory is used before the mode line that we want to print on, and add that to DISMEM to determine where we want to start POKEing. As an example of how to do this, let's suppose we have a graphics mode with four lines of mode 1, fifty lines of mode seven, three lines of mode four, and three lines of mode two (4X8 + 50X2 + 3X4 + 3X16 = 32 + 100 + 12 + 48 = 192); and we want to print on the second line of mode two. Checking the table above, we go:

| | | | |
|---|---|---|---|
| 4 lines of mode 1 = 4X20 = | 80 | |
| 50 lines of mode 7 = 50X40 = | 2000 | |
| 3 lines of mode 4 = 3X10 = | 30 | |
| 1 line of mode 2 = 1X20 = | 20 | (remember, we only count the lines *above* the one we want to print on) |
| For a grand total of: | 2130 | |

Therefore, memory location DISMEM + 2130 represents the first character in the second line of mode 2 *for this particular mode.* Memory location DISMEM + 2131 represents the second character, and so on up to DISMEM + 2149 for the twentieth character.

We know that POKEing the appropriate value into the appropriate location will cause the desired character to appear at the desired screen location. Since we already know how to determine the appropriate memory location, we now ask:

### How do I calculate the appropriate value for a character?

It turns out that the value to poke for a given character corresponds to the order in which the character descriptions are stored in ROM (see "Designing Your Own Atari Character Sets" in the March 1981 issue of COMPUTE!). As a quick memory refresher:

| ATASCII VALUE | VALUE TO POKE |
|---|---|
| 0-31 | 64-95 |
| 32-95 | 0-63 |
| 96-127 | 96-127 |

For reverse characters, just add 128 to the value of the normal character.

### My brain is in hibernation; how do I convert a character string to its appropriate values?

I'll leave you with the folowing self-explanatory subroutine that will take the (predefined) character string PRNTME$ and the starting memory location STARTHERE (also predefined and equal to DISMEM + offset) and POKE PRNTME$ into the appropriate (love that word!) memory locations. Enjoy!

```
30000 REM /*This loop will act on each c
haracter in PRINTME$*/
30010 FOR ME=1 TO LEN(PRNTME$)
30020 REM /*Find ATASCII value of charac
ter*/
30030 VALUE=ASC(PRNTME$(ME,ME))
30040 REM /*Subtract 128 temporarily if
it's a reverse character*/
30050 VALUE=VALUE-128*(VALUE>127):REM /*
see note below*/
30060 REM /*Make the appropriate value a
djustments*/
30070 VALUE=VALUE+64*(VALUE<32)-32*(VALU
E>31 AND VALUE<96)
30080 REM /*Convert back to reverse if n
ecessary*/
30090 VALUE=VALUE+128*(ASC(PRNTME$(ME,ME
))>127)
30100 POKE STARTHERE+ME-1,VALUE:REM /*re
member, ME starts at 0, not 1*/
30110 ? VALUE
30120 REM /*Go to next character*/
30130 NEXT ME
30140 REM /*All done, say goodbye*/
30150 RETURN
```

Note that (condition) equals 1 if the condition is true, Ø if it's not. Thus, X = 126:PRINT (X = 126) :PRINT (X = 127) will print a 1 followed by a Ø.

### SPECIAL CORRECTION NOTE

The original article, *"Designing Your Own Atari Graphics Modes,"* was incorrect in that the graphics mode you use initially should be a full screen mode. In other words, if graphics mode six is the mode that uses the most memory out of the modes you will be mixing, you should use GRAPHICS 22, not GRAPHICS 6 (just add 16 to the mode number). In the example used in the article, line 10 should read:

**10 GRAPHICS 17 and not 10 GRAPHICS 1**

My apologies for any problems caused by this mistake. ©

# Invaders From Outer Space

## An Atari Program Using One Joystick

David H. Markley
Reynoldsburg, Ohio

This program is a simplified version of an earlier program I wrote called ''Alien Landers'' and is designed to run in 8K of memory. The object of the game described here, is to protect your galactic base from an invasion of allien saucers of unknown origin. Your weapon against the alien saucer is a sophisticated photon cannon which you control by a joystick connected to controller port #1. The saucer on the other hand, is equipped with a mysterious paralyzing death ray which when energized, will immobilize and possibly vaporize your photon cannon. The game is played by defending your base against the saucers and achieving the highest score possible before your 3 photon cannons are destroyed. The program keeps track of the highest score obtained during any individual game and displays the final game score and the high game score at the end of each battle. After starting the game using the RUN command, additional battles are initiated by holding the start button down until the new game begins.

This program utilizes many of the fine features of the ATARI personal computer, such as the exciting sound effects and the advanced player/missile graphics. The player/missile graphics are used to enable the program to provide good animation without the overhead of saving and restoring the background as the player or missile image moves through the playfield. Another ATARI feature used here is the player/missile collision registers. These registers are associated with the ATARI's graphics and indicate when a player or missile overlaps with another player, missile, or playfield image.

The program from line 30 thru line 600 provides background setup and player/missile initialization. Initialization begins by allocating space for the player/missile image buffers. The buffers are placed in a free area of memory just below the graphics memory with the base address located 2048 bytes (8 pages) from the top of memory. This leaves the top 1K of memory for the graphics 2 display list and mapping data.

The functional player/missile (P/M) graphics area begins with the missile image buffer. Since the program uses the double resolution mode of P/M graphics, the missile image buffer begins 384 bytes from the base address and is followed by four player image buffers. Each buffer occupies 128 bytes of memory. Invaders from Outer Space uses only two of the four available players which are in the form of a saucer (player 1) and a photon cannon (player 2). The images of the saucer and cannon are formed by placing a bit pattern of the shapes into their corresponding player image buffers.

Lines 600 and 700 initialize the game's counters and registers. The high game score counter initialized by line 600 is used to keep track of the highest score obtained for all games played during any program run. This counter is only initialized one time. Registers and counters which are initialized every time a new game is played are all located at line 700.

With the playfield and players enabled and the games counters and registers initialized, the program is ready to begin antimating the graphics by entering the game's antimation sequencer loop (lines 1000 thru 2000). This section of the program is used to control the movement of the saucer, cannon, and photon. It also determines when to fire the saucer's death ray and checks for P/M collisions.

The sequencer loop begins by determining the saucer's horizontal position in relation to the play field. Normally the saucer will travel across the field in a left to right direction in steps of 5 horizontal increments. If the saucer is within the range of fire of the cannon however, and is currently under attack (M>0) the saucer will take evasive action by moving randomly within a short distance of either side of its current position. With the position of the saucer determined, the sequencer's next task is to produce the saucer's sound effect. This effect is created by stepping through a series of six frequencies in which the frequency is changed one step for each loop cycle. Once the saucer has been placed in its new position, a test is made to see if it is in attack range of the cannon. If the test (line 1060) indicates that the saucer is within 15 increments of the cannon, the program will go to the death ray handler (line 3100).

The next sequencer task to be handled is the control of the cannon and its associated photon missile (lines 1100 thru 1220). The program begins by checking to see if the photon is ready to be fired. If it is ready to fire or has reached the top of the playfield (M<J+8), its sound is turned off and the joystick trigger is examined. If the trigger is not pressed, the program will continue through the sequencer loop and will examine the trigger each time it passes through the loop. When the trigger is finally pressed, the M pointer will be set to an address of the missile image buffer which corresponds to a vertical position directly above the cannon's muzzle.

The horizontal position of the photon is handled by calculating the horizontal position of the cannon's center and placing it into the cannon's horizontal positioning missile register. Line 1110 controls the movement of a photon which is enroute to its target. To move an object vertically using the P/M graphics, the image is first removed from its current position within the image buffer and then rewritten into its new location. The P/M graphics function in such a way that as the base address of the image data is moved to a lower numbered address in the buffer, the image will appear to move upward on the screen. After each photon movement, the player/missile collision register is checked to see if the saucer has been hit. When a hit is detected the program exits the sequencer loop and goes to the routine at line 3000 to handle the collision. If the saucer is not hit the sequencer loop enters the code which handles cannon movement. When the joystick is moved to the left or right the cannon is shifted 5 increments to the left or right of its current position each pass through the loop. With the completion of this code, the sequence loop is now ready for the next pass.

The routine starting at line 3000 is used to control hits on the saucer. It begins by removing the photon from the playfield by clearing the first 29 bytes of the missile image buffer. This area represents the area on the playfield in which the photon can collide with the saucer. When the buffer has been cleared, the missile position pointer M is reset to zero as an indication that it is ready to be fired again. To help produce the effect of more than one saucer, the saucer's color is randomly changed each time it is hit. Finally the score is updated and the background flag (BK) is set to indicate an above ground explosion. The program then goes to the explosion handler routine (line 4000).

The routine starting at location 3100 is used to generate the saucer's death ray. It begins by positioning the ray at the center of the saucer. This is done by calculating the center of the saucer and placing this position into the ray's horizontal position missile register. The sound and visual effects are handled by lines 3120 thru 3140. The ray is first drawn by setting bits 0 and 1 for every third byte in the missile image buffer. It is then erased by again accessing every third byte and clearing bits 0 and 1 to zero. As the ray is removed, the frequency of its associated sound generator is decreased. Upon completing the death ray, the program examines the P/M collision register corresponding to the ray missile and the cannon player. If the collision took place, an explosion effect is produced by moving the cannon to a position off the playfield and setting the explosion back ground flag for the bottom text window before calling the explosion handler at line 4000.

The explosion handler is used to produce the explosion effect for both the saucer and the cannon. To begin, the explosion handler resets the P/M collision registers. The collision registers are designed to be latching and must be reset each time a collision has been registered. After terminating the sound of the photon which may have caused the explosion, the program begins to produce the visual and audio effects of the explosion. This is done by stepping the variable X from 14 to 0 and using its value to control the intensity of the color and sound. The background which is determined by BK was selected by the calling program before the explosion handler was entered. When the explosion is complete, the playfield colors are returned to their normal hue.

When the explosion effect is complete, the program is ready to display the cannon number or score. Since the program displays all messages at the top of the screen using graphics 2, the saucer must be removed from the screen by setting its P/M color registers to zero. The saucer's original color is stored in X and will be again restored at the completion of the message display cycle. After a short delay while the message is being displayed, a test is made to determine if the last cannon has been hit. If not, the message is removed and the background is restored and the game continues. When the game is finally completed, the score is compared to the high game score. If the score resulting from the last game is higher, the high game score is updated. The program indicates the end of the game by alternately displaying the last game and high game scores. After each display of the high game score the start switch is examined to see if a new game is requested.

```
10 REM INVADERS FROM OUTER SPACE
15 REM COPYRIGHT (c) 1980
20 REM BY DAVID H. MARKLEY
```

**Print title**

```
25 GRAPHICS 2
30 PRINT #6;"     INVADERS"
35 PRINT #6;"       FROM"
40 PRINT #6;"    OUTER SPACE"
```

**Start player/missile data list 2K from top of memory. Place in P/M base register and calculate beginning address of the player data lists.**

```
50 I=PEEK(106)-8:POKE 54279,I
60 J=I*256+384
```

**Clear P/M buffer area**

```
70 FOR X=J TO J+384
80 POKE X,0
90 NEXT X
```

**Generate play field**

```
100 GRAPHICS 2
110 SETCOLOR 2,11,6
120 COLOR 3
130 POKE 84,0:POKE 85,4:? #6;" . "," . "
140 POKE 84,5:POKE 85,2:? #6;"*    . "
150 POKE 84,3:POKE 85,8:? #6;" . "," . "
160 POKE 84,7:POKE 85,14:? #6;" . "
```

**Enable player/missile DMA for double line resolution.**

```
200 POKE 559,46
```

**Place saucer image data into first player's image buffer area and set its color to green (200). Set the saucer's initial horizontal position to 70.**