THE MICROCOMPUTER MAGAZINE

commodore

APRIL/MAY, 1982
$2.50

Software is the solution... for making your business more profitable!

And, the most comprehensive business and professional software is Info-Solutions from Info-Designs, Inc.

Info-Designs, Inc. is a rapidly growing company with a "commitment to provide the highest quality microcomputer software anywhere!"

Our Professional Practice Software Solutions provide a comprehensive management system for attorneys, engineers, architects, consultants, accountants, and other professionals who derive income from billing their professional services and reimbursable expenses.

Our management Accounting Software Solutions provide a totally integrated accounting system for the small to medium size business—retailer, wholesaler, distributor, manufacturer, and service company.

OFFICIALLY APPROVED BY COMMODORE

Now, on one compact Commodore CBM microcomputer system, you can achieve better management control of unbilled time, billings, accounts receivable, accounts payable, checkwriting, payroll, general ledger, inventory management and on-line order entry/point-of-sale.

See your local Commodore dealer today for the **Information-Solution** to your business automation needs.

InfoDesigns

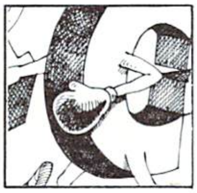
INFORMATION SOLUTIONS USING MICROCOMPUTER TECHNOLOGY

6905 Telegraph Rd. ● Birmingham, MI 48010 ● (313) 540-4010

THE MICROCOMPUTER MAGAZINE

commodore

FEATURES



DEPARTMENTS

Commodore News

Education

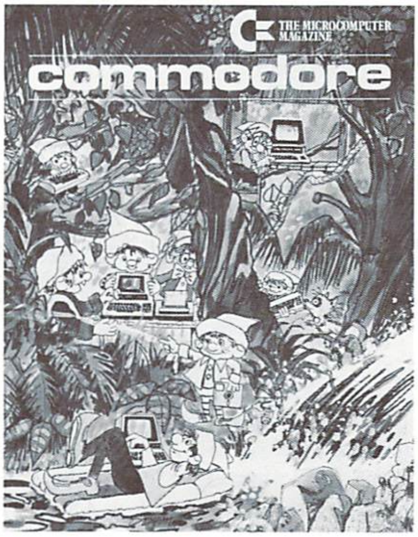
Business

VIC 20

Programmer's Tips

Excerpts From a Technical Notebook

Product Review



commodore

Editorial Manager
David A. Kaminer
Editor
Paul Fleming
Staff Editor
Jody Miller
Staff Writers
Rick Cotton
Paul Goheen
Mike Heck
Pat McAllister
Dave Middleton
Sue Mittnacht
Dave Moyssiadis
John O'Brien
Dave Scott
Michael Tomczyk
Contributing Writers
Jim Butterfield
Elizabeth Deal
Fred Chesire
Karl Hildon
John Krebs
Preston Marshall
Louise Schultz
Jeremy Wells
Bob West
Hubert Youmans
Jack Weaver
Technical Staff
Paul Goheen
Pat McAllister
Dave Middleton
Dave Scott
Joseph Siciliano
John Stockman
Michael Tomczyk
Circulation Manager
John O'Brien
Circulation Assistant
Susan Markle
Advertising Manager
Paul Fleming

Cover art by Don Dyen

Commodore Magazine is issued bimonthly. Published by the Computer Systems Division, Commodore Business Machines, Inc., 681 Moore Rd., King of Prussia, PA 19406 (215) 337-7100. The U.S. Subscriber rate: $15.00 per year. In Canada and Mexico the Subscriber rate is $25.00. No material may be reprinted without permission. Volume 3. Number 2.

Q&A HOTLINE

Q. I have a 2040 Dual Disk Drive with 2.1 upgrade DOS (equivalent to 4040), and a 2001 PET (upgrade 3.0 BASIC). If the PET is RESET (i.e., switched on/off/on) while a floppy disk is in the drive, is there any risk of damage to the disk data? What is happening as the disk drive spins and whirs when I reset my PET? Is it a ''defined'' event that is safe for my disks or should I remove them first?

Also, is there more than one version of DOS 2.1? Does my upgraded 2040 differ in any way with the 4040 version of DOS 2.1?

G. Stone
Laurel, MD

A. *It is okay to leave the disks in the drive as long as the computer is the only thing switched off and on. The computer at power-up time sends a reset signal over the IEEE bus to all peripherals. The peripherals jump to their own pre-defined initialization routines, and reset internal registers, etc. In the case of the disk drive, the heads are not write-enabled, so there is no chance of damage. Of course, if the disk drive were powered down there is a chance of damaging the diskette, because the logic would no longer be defined when the voltage fails.*

There is only one version of level 2.1 DOS, which is the DOS in the 4040 disk drive. Your new ROMs have the same part numbers as the ROMs installed in the 4040 drives; the front panel label on your drive is the only remaining 2040 part. The DOS levels installed in the drives are:

Model 2040 — DOS 1.0
(Discontinued with introduction of 4040)
Model 4040 — DOS 2.1
Model 8050 — DOS 2.5

Q. Is it possible to get the new 901472-04 ROM for the 2022 Printer? If not, is a higher revision available?

M. Raymond
Neufchatel, Quebec

Q. I would like to congratulate you on the notable improvement of your magazine over the last few issues. In reference to the ROM genealogy article in the December, 1981 issue, how does one obtain a 901472-07 final fix ROM for the 2022 Printer? I purchased my 2022 in December, 1979 and was able to obtain a 04 ROM about a year later. I have been unable to find an 07 ROM.

J. Melaugh
Tulsa, Oklahoma

A. *The latest ROM for the 2022 Printer is part number 901472-07. Ask your authorized Commodore dealer for more information. Incidentally, the bidirectional upgrade ROM for the 4022 Printer (part number is 901631-02) is also available now.*

Q. How does Commodore's IEEE differ from Hewlett-Packard's IEEE?

J. Biscunyak
Manahawkin, NJ

A. *There are several differences in both hardware and software. Commodore features a Remote Enable line that is tied permanently to ground. The Interface Clear line is not used and the Commodore Service request line is not implemented in the firmware. Also, the HP does not include a 64 millisecond time-out on their bus. For more information, refer to* The PET and the IEEE-4888, *which is published by Osborne/McGraw Hill, and should be available at your local Commodore dealer.*

Q. How can a window be set on the 8032?

L. Corcoran
Sunnyvale, CA

A. *Under program control, move the cursor to the upper-left corner of your desired window and PRINT CHR$ (15). Then move the cursor to the lower-right corner of the window and PRINT CHR$ (143). This procedure is elaborated in the PET/CBM Pesonal Computer Guide/Second edition.*

Continued on page 4.

IF YOU'RE WAITING FOR THE PRICE OF WORD PROCESSORS TO FALL WITHIN REASON,

Everyone expected it would happen sooner or later...with WordPro PLUS™ it already has! Now all the marvelous benefits of expensive and advanced word processing systems are available on Commodore computers, America's largest selling computer line. WordPro PLUS, when combined with the new 80 column CBM 8032, creates a word processing system comparable to virtually any other top quality word processor available—but at savings of thousands of dollars!

New, low cost computer technology is now available at a fraction of what you would expect to pay. This technology allowed Commodore to introduce the new and revolutionary CBM 8032 Computer.

WordPro PLUS turns this new CBM 8032 Computer into a sophisticated, time saving word processing tool. With WordPro PLUS, documents are displayed on the computer's screen. Editing and last minute revisions are simple and easy. No more lengthy re-typing sessions. Letters and documents are easily re-called from memory storage for editing or printing with final drafts printed perfectly at over five hundred words per minute!

Our nationwide team of professional dealers will show you how your office will benefit by using WordPro PLUS. At a price far less than you realize.

Invest in your office's future...
Invest in WordPro PLUS...
Call us today for the name of the WordPro PLUS dealer nearest you.

Professional Software Inc.
166 Crescent Road
Needham, MA 02194
(617) 444-5224
TELEX: 95 1579

Professional Software, Inc. WordPro was designed and written by Steve Punter.

Q&A HOTLINE

Q. When using the modem with the SuperPET, why do characters seem to be echoed back onto the screen?

B. Salter
Madison, WI

A. *The built-in communications program on the SuperPET prints the characters on the screen as they are keyed in from the keyboard. If the host computer is set to echo, then the received character will be a duplicate, causing the screen to display each character twice. Almost every host computer can be stopped from echoing. The specific command depends on the host computer being used.*

Q. We are using seven PETs, but have only two disk drives. Would it be possible to connect more than one PET to a disk drive?

V. Dunleavy
Waco, TX

A. *Although it is not recommended, it is possible to connect more than one computer to a disk drive. Be careful to have only one computer access the disk drive at any one moment, or else the system may crash. One way to prevent this problem is to use the MUPET, which is sold by Canadian Micro Distributors, Milton, Ontario, (416) 878-7277. This device allows connection of up to eight computers to one disk drive. Another alternative is the REGENT, which is sold by Skyles Electric Works, Mountain View, CA, (415) 965-1735. This device allows connection of up to 16 computers to one drive. It works especially well in a school environment.*

Q. I have a PET 20001-8K that was manufactured in 1978 (40-column black and white CRT, graphic calculator-style keyboard, built-in cassette, and original configuration ROMs). How can I upgrade my ROMs to level 2.0 or 3.0, and what do I have to go through to change my RAMs? The 6550s, 6540s, and 6520s are in sockets while all other connections are direct to the board.

R. Ludwick
Chadron, Nebraska

A. *The main logic board in your PET has 28-pin ROMs. You can install an upgrade set of ROMs which are sometimes called "new ROMs." Since your model hasn't been manufactured in quite a while, your authorized dealer may not have the necessary ROMs in stock, but they can be ordered. The part number to order for the 28-pin ROMs is 320432. Please note that some of the older PETs may require 24-pin ROMs (part number 320433), which are also available through your dealer. The numbers printed on the ROMs and their socket locations were listed on page 69 of the December issue of COMMODORE Magazine.*

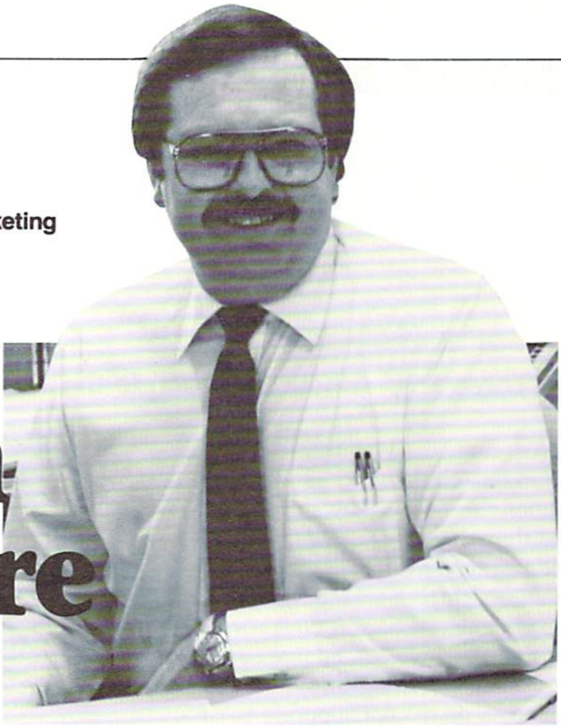
Since the 6550 RAM chips are no longer manufactured, they are getting quite hard to find. Compu-Think (Sunnyvale, CA, 408-245-4033) makes a RAM memory board that can be used as an expansion and/or replacement of the memory now in your computer. Another excellent expansion RAM board is manufactured by Skyles Electric Works (Mountain View, CA, 415-965-1735). Installing these memory boards may be much easier than trying to modify your present board to accept different RAM chips. ■

Just can't get the
answer to your questions
on Commodore Equipment/
Applications???

Write: **HOTLINE**
c/o Commodore Magazine
681 Moore Rd.
King of Prussia, PA 19406

Bruce Downing, Ph.D.
Director of Industry Marketing

From Where I Sit



Planning for Computers in Education

The computer revolution has caught many people by surprise despite the seers of technology, who for some time have been predicting sweeping changes throughout our society as a result of computers. Now that the revolution is actually here, its effects—many good and some bad—are increasingly clear.

The opportunities provided by microcomputers in education are just now being recognized. The very fact that these new educational tools can be used by both teachers and students in the privacy of a home as well as in the traditional classroom signifies the truly ''personal'' aspect of microcomputing. However, effective use of computers in education requires an understanding by educators of both the *potentials* and *limitations* inherent with this latest addition to the classroom.

As with any learning medium, the computer has to fit *within* the context of an educational plan, the intent of which is to provide the best possible learning experiences for students. Designing and executing such plans has been, and will continue to be, a major role for educators.

Often, computer applications are started by a few interested teachers who are able to excite their students and demonstrate the viability of computers to their colleagues. As interest grows, the desire for additional equipment, software, and other resources intensifies. Parent groups and school boards also pressure administrators to ''get into the computer age.''

Control of this growth, while seeking financial funds for its support, becomes a major source of frustration. This problem is particularly evident when educational benefits are unclear and most educators have had no training in computing.

One way to avoid some of these pitfalls is to develop a computing plan before substantial resources are allocated. Experience at many educational institutions indicates that many issues will need to be successfully addressed to ensure the maximum learning benefits for the resources expended. Key elements of such a plan include:

Educational Objectives

Each district, school, or institution needs to decide what should be accomplished with computers. Teaching computer literacy, record keeping, computer-assisted instruction and other applications should be considered. Deciding where to begin is extremely important. But whatever is decided, *know and adhere to your objectives.*

Administration

Who is involved in deciding the applications, expenditures, priorities, and other issues related to computers? Planning activities generally should involve both faculty and administrators, with input solicited from the community.

Training

Training for teachers and administrators needs to be provided for in the plan. Successful planning and implementation requires intelligent decisions based on the knowledge of what computers can and cannot accomplish.

Hardware Selection

The evaluation and selection of particular computers should be guided by factors consistent with the planned objectives. In other words, the computer you choose must be able to handle the educational objectives that should already have been determined. Remember, deciding where to start is the primary objective.

Software Selection

Similar to the choice of hardware, programs chosen to run on your computer should also be evaluated on the basis of how well they meet *your* educational objectives. Obviously, the software must also be completely compatible with your hardware.

Support Personnel

Additional staff may be required for training or software development. If teachers are assuming these roles, released time programs must be considered.

In subsequent issues of this magazine, I will examine each of these significant areas thoroughly. For the time being, this general outline should give you some food for thought. Obviously, the points made are not hard-fast rules. Planning will take different forms, depending on local situations. However, experience shows that these general guidelines are time-tested and successful.

Dr. Downing's column will be a regular feature of the Education section of this magazine. Other Commodore decision-makers will occasionally be featured in this space, in addition to Dr. Downing's comments. ■

Editor's Notes

The Three R's: A new approach to the same answers

Microcomputer-aided instruction is becoming as common to educational institutions as those hallowed disciplines: "the three R's." And, while the goals of education remain basically the same, the approach to teaching has changed dramatically. Perhaps it could even be said that the three R's no longer represent "reading, 'riting, and 'rithmetic." Instead, a more contemporary twist to this alliterative phrase might be "READY, RUN, and RETURN."

Why? Because schools and school districts throughout this country have and are continuing to turn to microcomputers as valuable tools in providing students with new and fun insights into learning. PET microcomputers are leading the way in this exciting educational revolution. The accounts of schools—from elementary through graduate level—that use PET microcomputers are truly endless. And each story has its own unique twist, reflecting the power and flexibility that microcomputers can offer in solving and improving learning situations.

While this new approach to learning may seem incredibly threatening to the more old fashioned segment of our populace, it must be pointed out that the microcomputer revolution has not sat still for this stodgy minority. Instead, it has grown and will continue to grow rapidly, as more and more educators and parents jump aboard the bandwagon.

One of the roles of this magazine is to keep the members of that bandwagon as informed as possible. Consequently, as the ranks of schools using Commodore equipment increase, so does our commitment to sharing as much worthwhile information as we can find. And the best way of finding information is through our educational users. Please help educate us! Tell us—and your fellow readers—what type of courses you have developed, how many microcomputers your school has, or even something we never would have thought of asking.

Incidentally, our interest in your willingness to share information goes far beyond the pages of this magazine. In fact, Commodore is enthusiastically committed to establishing Education Resource Centers in any institutions or mircrcomputer laboratories where Commodore products are used for education applications (see the Education Section in this issue for more detailed information).

And, while you're reviewing that information, take note of the expanded coverage we've accorded the educational field. Without a doubt, Commodore realizes the importance of microcomputers in education. And, unquestionably, Commodore Magazine recognizes the need of our users to be kept informed.■

Paul Fleming

Paul Fleming
Editor

USERS CLUBS: Sound Off!

We're continuing to compile a list of all Commodore Users clubs throughout the country. If you'd like to add your name to the rolls, please send your club's name, address, and other pertinent information to:

Commodore Users Clubs
c/o Editor
Commodore Magazine
681 Moore Road
King of Prussia, PA 19406

And remember, once our list is comprehensive enough, we will begin forwarding valuable information to clubs on a regular basis, including hardware and software updates, technical bulletins, new product announcements, and troubleshooting tips.

ARIZONA

VIC Users Group
1206 N. Fraser Drive
Mesa, AZ 85203
Contact: Paul V.
Muffuletto

ARKANSAS

Commodore/PET Users
Club
Conway Middle School
Davis Street
Conway, AR 72032
Contact: Geneva Bowlin

CALIFORNIA

Lawrence Hall of Science
UC Berkeley
Computer Project, Room
254
Berkeley, CA 94720
(415) 642-3598

PALS (PETS Around
Livermore Society)
886 South K
Livermore, CA 94550
Contact: John Rambo

SCPUG Southern
California PET Users
Group
c/o Data Equipment
Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: First Tuesday
of each month

California VIC Users
Group "VIC-VILLE"
c/o Data Equipment
Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: Second Tues.
of each month

Sacramento PET
Workshop
Box 543
Davis, CA 95616
Contact: John Bowles

Commodore Users Club
1041 Foxenwoods Drive
Santa Maria, CA 93455
(805) 937-4106
Contact: Greg Johnson

Valley Computer Club
2006 Magnolia Blvd.
Burbank, CA
(213) 849-4094
1st Wed. 6 p.m.

Valley Computer Club
1913 Booth Road
Ceres, CA 95307

PUG of Silicon Valley
22355 Rancho Ventura
Road
Cupertino, CA 95014

BAMBUG
1450 53rd Street
Emeryville, CA
(415) 523-7396

North Orange County
Computer Club
3030 Topaz, Apt. A
Fullerton, CA 92361
Dave Smith

Lincoln Computer Club
750 E. Yosemite
Manteca, CA 95336
John Fung, Advisor

PET on the Air
525 Crestlake Drive
San Francisco, CA 94132
Max J. Babin, secretary

PALS (PETs Around
Livermore Society)
886 South K
Livermore, CA 94550
(415) 449-1084
Every third Wednesday
7:30 p.m.
Contact: J. Johnson

SPHINX
314 10th Avenue
Oakland, CA
(415) 451-6364
Every 2nd & 4th Thurs.

San Diego PUG
c/o D. Costarakis
3562 Union Street
(714) 235-7626
7 a.m.-4 p.m.

Walnut Creek PET
Users Club
1815 Ygnacio Valley
Road
Walnut Creek, CA 94596

CONNECTICUT

John F. Garbarino
Skiff Lane Masons Island
Mystic, CT 06355
(203) 536-9789

Commodore User Club
Wethersfield High School
411 Wolcott Hill Road
Wethersfield, CT 06109
Contact: Daniel G.
Spaneas

FLORIDA

Jacksonville Area
PET Society
401 Monument Road,
#177
Jacksonville, FL 32211

Richard Prestien
6278 SW 14th Street
Miami, FL 33144

South Florida
PET Users Group
Dave Young
7170 S.W. 11th
West Hollywood,
FL 33023
(305) 987-6982

ILLINOIS

Shelly Wernikoff
2731 N. Milwaukee
Avenue
Chicago, IL 60647

Central Illinois
PET Owners
Rick Goldsmith
2730 Townway Road
#E-54
Danville, IL 61832

PET VIC Club (PVC)
40 S. Lincoln
Mundelein, IL 60060
Contact: Paul Schmidt,
president

Rockford Area PET
User's Group
1608 Benton Street
Rockford, IL 61107

INDIANA

PET Users
Jerry Brinson
PO Box 36014
Indianapolis, IN 46236
(317) 898-3604

GHS Computer Club
c/o Grangeville High
School
910 S. D St.
Grangeville, ID 83530
Contact: Don Kissinger

Cardinal Sales
6225 Coffman Road
Indianapolis, IN 46268
(317) 298-9650
Contact: Carol Wheeler

CHUG (Commodore
Hardware Users Group)
12104 Meadow Lane
Oaklandon, IN 46236
Contact: Ted Powell

IOWA

PET Users Group
c/o Don Vorhies
1321 42 St. SE
Cedar Rapids, IA 52403

KANSAS

Wichita Area PET
Users Group
2231 Bullinger
Wichita, Kansas 67204
(316) 838-0518
Contact: Mel Zandler

MARYLAND

Assoc. of Personal
Computer Users
5014 Rodman Road
Bethesda, MD 20016

MICHIGAN

David Liem
14361 Warwick Street
Detroit, MI 48223

PET User Group
Peter Oakes
2235 Lakeshore Drive
Muskegon, MI 49441

Toledo PETS
734 Donna Drive
Temperance, MI 48182
Contact: Gerald Carter,
president

Commodore User Club
32303 Columbus Drive
Warren, MI 48093
Contact: Robert
Steinbrecher

MINNESOTA

Twin Cities
John Fung
Twin Cities, MN
(612) 376-5465

MISSOURI

Clearwater Club
Clearwater School
Star Route
Piedmont, MO 63957
Contact: Carolyn Polk

St. Louis Club
Mary Perkinson
46 Westwood Court
St. Louis, MO 63131
(314) 432-5225

MONTANA

Powder River
Computer Club
Powder River County
High School
Broadus, MT 59317
Contact: Jim Sampson

Commodore User Club
1109 West Broadway
Butte, MT 59701
Contact: Mike McCarthy

NEVADA

Las Vegas PET Users
4884 Iron Avenue
Las Vegas, NV 89110

NEW JERSEY

Amateur Computer
Group of New Jersey
John Loofbourrow
UCTI, 1776 Raritan Road
Scotch Plains, NJ 07076
(201) 233-7068

Amateur Computer
Group
18 Alpine Drive
Wayne, NJ 07470

Somerset Users Club
49 Marcy Street
Somerset, NJ 08873
Contact: Robert Holzer

NEW HAMPSHIRE

Northern New England
Computer Society
PO Box 69
Berlin, NH 03570

NEW YORK

Capital District PET Users
Ben Green
Albany Area, NY
(518) 370-1820

Long Island PET Society
Ralph Bressler
Harborfields HS
Taylor Avenue
Greenlawn, NY 11740

PET User Club
of Westchester
Box 1280
White Plains, NY 10602
Contact: Ben Meyer

LIVE (Long Island
VIC Enthusiasts)
17 Picadilly Road
Great Neck, NY 11023
Contact: Arnold Friedman

PET User Group
Westchester, NY
(914) 428-7872
Every 2nd Tuesday

PET User Group
c/o Meyer
35 Barker Avenue
White Plains, NY 10610

NORTH CAROLINA

Amateur Radio PET
User's Group
PO Box 30694
Raleigh, NC 27622
Contact: Hank Roth

Commodore Users Club
4241 Castleton Road
Charlotte, NC 28211
Contact: Ed Harris

OHIO

Dayton Area PET
User Group
933 Livingston Drive
Xenia, OH 45385
B. Worby, president
(513) 848-2065
J. Watson, secretary
(513) 372-2052

Central Ohio PET
Users Group
107 S. Westmoor Avenue
Columbus, Ohio 43204
(614) 274-6451
Contact: Philip H. Lynch

OREGON

NW PET Users Group
John F. Jones
2134 N.E. 45th Avenue
Portland, OR 97213

PENNSYLVANIA

PET User Group
Gene Beals
PO Box 371
Montgomeryville,
PA 18936

Penn Conference
Computer Club
c/o Penn Conference
of SDA
720 Museum Road
Reading, PA 19611
Contact: Dan R. Knepp

PACS PET Users Group
20th & Olney Streets
Philadelphia, PA

Glen Schwartz
807 Avon
Philadelphia, PA 19116

Gene Planchak
4820 Anne Lane
Sharpsville, PA 15150
(412) 962-9682

SOUTH DAKOTA

PET User Group
515 South Duff
Mitchell, SD 57301
(605) 996-8277
Contact: Jim Dallas

TENNESSEE

River City Computer
Hobbyists
Memphis, TN
1st Mon. at Main Library

TEXAS

SCOPE
1020 Summit Circle
Carrolton, TX 75006

PET Users
2001 Bryan Tower
Suite 3800
Dallas, TX 75201

Larry Williams
PO Box 652
San Antonio, TX 78293

PET User Group
John Bowen
Texas A & M
Microcomputer Club
Texas A & M, TX

UTAH

Utah PUG
Jack Fleck
2236 Washington Blvd.
Ogden, UT 84401

The Commodore
User's Club
742 Taylor Avenue
Ogden, Utah 84404
Contact: Todd Woods
Kap, president;
David J. Shreeve,
vice president

The VIClic
799 Ponderosa Drive
Sandy, UT 84070
Contact: Steve Graham

VIRGINIA

Northern VA PET Users
Bob Karpen
2045 Eakins Court
Reston, VA 22091
(803) 860-9116

WASHINGTON

NW PET Users Group
2565 Dexter N. #203
Seattle, WA 98109
Contact: Richard Ball

WISCONSIN

Sewpus
c/o Theodore J.
Polozynski
PO Box 21851
Milwaukee, WI 53221

CANADA

Toronto PET
Users Group
381 Lawrence Ave. West
Toronto, Ontario, Canada
M5M 1B9
(416) 782-9252
Contact: Chris Bennett

Commodore's '3 for 2' Educational Grant Program Going Strong

The trend-setting "three for two" educational grant program started by Commodore Business Machines, Inc., in 1979 has given away more than 13,000 microcomputers worth more than $15.2 million.

Since the inception of the program in September, 1979, some 13,200 micros including PET 2000 and 4000 series and CBM 8000 series units, have been delivered free to public and private institutions from elementary schools to colleges and universities. Including units purchased, the three for two program has helped provide nearly 40,000 micros in use today for education.

The three for two educational program grants a free Commodore microcomputer of up to equal value to education institutions which buy two Commodore micros from authorized dealers.

"While some micro companies are only thinking about giving away units for education purposes, Commodore has been setting the pace for over two and a half years," said Kit Spencer, vice president-marketing for Commodore's Computer Systems Division. "Schools and colleges know that they can go to authorized Commodore dealers today and have the micros they need right away." ■

Here's How to Get Commodore's 'Stamp of Approval' for Your Product

More and more, you'll be seeing Commodore's "Stamp of Approval" on many of the quality products that are compatible with the Commodore line of microcomputers. This Approved Product status is granted to any product that has been extensively reviewed by Commodore and has met the requirements for the approval endorsement.

Generally, this procedure involves the receipt and review of the product by Commodore's Approved Products Department. The hardware or software will undergo strict evaluation, according to the approval criteria established by Commodore. Depending upon the quality and applicability of the hardware or software, Commodore may approve the product and sign a contract with the vendor.

OFFICIALLY APPROVED BY
COMMODORE

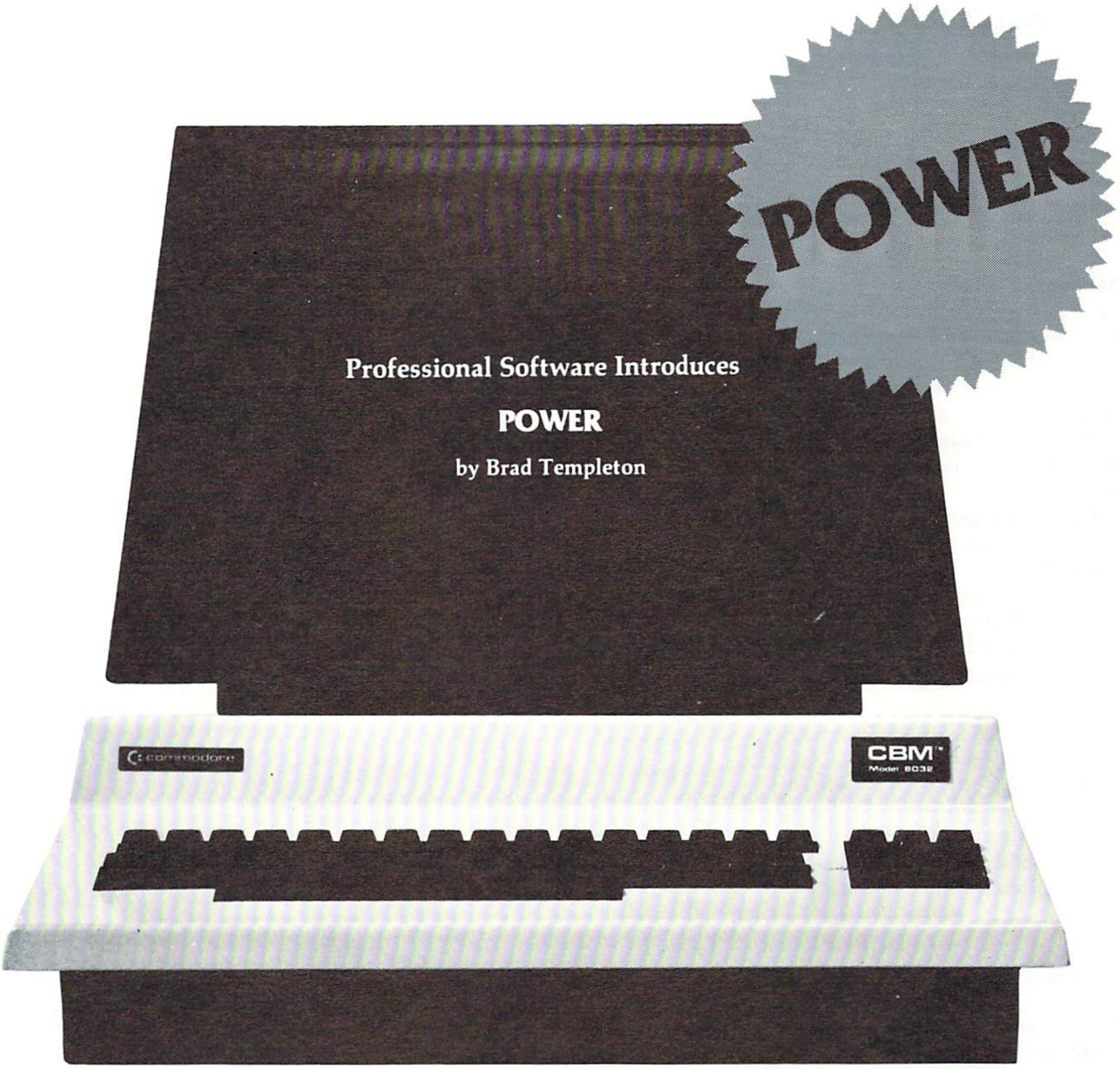
Despite the comprehensive criteria that all hardware and software must meet, the strict evaluation procedures ensure that only the highest quality products will receive Commodore's approval. In turn, a company that can offer an officially approved Commodore product will reap the numerous benefits of this association. Some of these benefits include:

- Use of the widely recognized Commodore logo with all approved products.
- Free listing of product (including Approved Product logo) in Commodore's Software Encyclopedia.
- Nationwide exposure through Commodore's marketing bulletins and dealer alerts.
- Reduced rates for advertising in the increasingly popular Commodore Magazine.
- Choice of preferred booth space at any Commodore-sponsored trade shows.

If your're interested in these and many other benefits, contact us to find out more. Profit the most from your Commodore-compatible hardware/software by making it a "Commodore Approved" product. ■

For more information, contact:

Approved Products Department
Commodore Business Machines
681 Moore Road
King of Prussia, PA 19406



Professional Software Introduces

POWER

by Brad Templeton

ADD POWER TO YOUR COMMODORE COMPUTER

$89.95

POWER produces a dramatic improvement in the ease of editing BASIC on Commodore's computers. POWER is a programmer's utility package (in a 4K ROM) that contains a series of new commands and utilities which are added to the Screen Editor and the BASIC Interpreter. Designed for the CBM BASIC user, POWER contains special editing, programming, and software debugging tools not found in any other microcomputer BASIC. POWER is easy to use and is sold complete with a full operator's manual written by Jim Butterfield.

POWER's special keyboard 'instant action' features and additional commands make up for, and go beyond the limitations of CBM BASIC. The added features include auto line numbering, tracing, single stepping through programs, line renumbering, and definition of keys as BASIC keywords. POWER even includes new "stick-on" keycap labels. The cursor movement keys are enhanced by the addition of auto-repeat and text searching functions are added to help ease program modification. Cursor UP and cursor DOWN produce **previous** and next lines of source code. COMPLETE BASIC program listings in memory can be displayed on the screen and scrolled in either direction. POWER is a must for every serious CBM user.

Call us today, for the name of the Professional Software dealer nearest you.

Professional Software Inc.
166 Crescent Road
Needham, MA 02194
Tel: (617) 444-5224 Telex #951579

™POWER is a registered trademark of Professional Software, Inc.

Even More Software for Commodore Micros . . .

UCSD Pascal Version 4.0

Pascal was developed in 1968 because of the dissatisfaction of Dr. Niklaus Wirth with the major languages whose features and constructs could not be explained logically and were often in opposition to systematic reasoning. Initially developed for teaching computer programming, many implementations have been developed since that time using the initial standards. Because of its adherence to "structured" constructs, Pascal has developed into one of the standard languages used to teach programming and for writing business application packages.

The UCSD implementation of Pascal, originally developed at the University of California at San Diego, has continued to be enhanced and implemented by Softech Microsystems on a large variety of computer systems and has become one of the most popular Pascal implementations for education and business today.

Commodore UCSD Pascal version 4.0 is designed to run with the CBM 8096 (8032 with 64K add-on memory), CBM 4040 and CBM 8050 disk drives, all CBM printers, including the CBM 2022 (with 04 ROM upgrade), the CBM 4022, the 8024 (Mannesman Tally), the CBM 8023, and the CBM 8300, as well as the CBM 8010 modem. The system can also be configured to run with other ASCII printers such as NEC Spinwriter, QUME, TEC, and Epson MX80.

Some of the features included with Commodore UCSD Pascal are:

- Screen Oriented Editor
- YALOE (Yet Another Line Oriented Editor)
- Adaptable Assembler generating relocatable or absolute object code
- Linker
- Macros
- Debugger

This package is ideal for educational environments where computer programming is taught and also for the professional software developer who wishes to create an application package using a truly structured programming language.

The suggested retail price for this package is $175.00.

. . . CMAR Multi-Key File Access System

Attention: software developers looking for an easy file access method!! Commodore has just introduced CMAR, a utility that allows you to set up files, by generic key, and perform all the necessary file maintenance functions such as read, write, change, and delete. CMAR files are dynamic, eliminating the need for reorganization whenever key and data records are modified.

CMAR requires slightly less than 6K for the program and an additional 4K for a buffer pool, leaving 22K for your application program. CMAR allows files to be expanded to any size, limited only by the disk storage capacity. Up to five files can be open for use at the same time, and because of an advanced technique (using a single intermediate level) for finding a key position, even the "worst case" key search is fast.

CMAR is designed to work with the CBM 8032 and CBM 4040 or 8050 disk drives. It will also be configured to run with the CBM Hard Disk when it becomes available. CMAR is written in 6502 machine language and it interacts directly with BASIC 4.0.

This product is exactly what you software developers have been looking for, so come and get it!

The suggested retail price for the package is $150.00.

. . . ATLAS 1200 Equipment Maintenance System

If you are an Independent Service Organization (ISO) or a Third Party Maintenance Company, and would like to increase revenue flow while decreasing time spent on accounting and administration, then the ATLAS 1200 Equipment Maintenance System may be what you have been looking for.

ATLAS 1200 allows the user to maintain service customer and equipment information, keep an accounts receivable on each customer, identify and log all calls for later retrieval, track equipment under warranty as a PM or a standard call, and produce statements on a timely basis. The system also will produce nine Management Reports: PM Schedule by territory, Machine to Territory Listing by model #, Cost of Service-History Report by model, Service Rep Revenue Report-MTD/YTD Hour/Dollars, Projected Company Revenue Report, Customer Listing, Plan Expiration Report by month, and Model Analysis Report.

ATLAS 1200 is designed to run with the CBM 8032 and CBM 8050 disk drive and any of the following CBM printers:

- CBM 4022
- CBM 8023
- CBM 8300P

The package is available now so place your order with your nearest authorized Commodore dealer.

The suggested retail price is $595.00.■

Professional Tax Prep System

The already popular Professional Tax Preparation System has been greatly enhanced this year to aid the "professional" Tax Accountant and/or Consultant. This year the package has the capability to compute all of the following schedules and forms:

- 1040 Long
- Schedules A, B, C, D, G, TC, SE, E, and F
- Numbered Schedules as follows:
  - 2106 - Employee Business Expense
  - 3903 - Moving Expense
  - 2441 - Child Care Credit
  - 3468 - Investment Credit
  - 5695 - Home Energy Credit
  - 2210 - Underpayment of Estimated Tax
  - 1040-ES - Estimated Tax Payments

It also has the capability to do regular or "income averaging," and simultaneous separate or joint returns.

And finally, it will compute State 1040 forms for California, New Jersey, New York, Pennsylvania, and Florida. Other State modules, such as Illinois and Massachusetts, can be obtained from CFI in New York.

The program was designed to print on either the actual IRS forms or an IRS-approved form which is available from Spee-D-Tax.

The package was designed to work with a CBM 8032 or 4032, CBM 8050 or 4040 Disk Drives, and any CBM printer or any properly interfaced ASCII printer.

In-Season Retail Price: $800.00 (This price is effective through April 15, 1982).

Off-Season Retail Price: $300.00 (This price is effective from April 16, 1982 through December 31, 1982).■

COBOL to Become 6th SuperPET Language

Reaffirming its commitment to continued development of a new generation microcomputer with features and languages traditionally found on mainframes, Commodore has announced an agreement with Waterloo Computing Systems Limited to develop COBOL for the SuperPET computer.

With this announcement, COBOL becomes the sixth interpretative language developed by Waterloo for the SuperPET. The other languages are: BASIC, FORTRAN, APL, Pascal, and 6809 Assembler.

This latest SuperPET language is planned to support ANSI-1974 COBOL Standards in the following modules: NUCLEUS, SEQUENTIAL I-O, RANDOM I-O, and TABLE HANDLING. Support Level I in these modules, as well as selected features of Level II, include complete support for the PERFORM, STRING, and UNSTRING verbs. COBOL for the SuperPET is interactive, and includes an interactive debugging system.

Presently, documentation for the SuperPET package includes a system overview and reference manuals for each of the product's five languages. The newly revised package will also include a COBOL reference manual.

For a nominal fee, existing SuperPET users can purchase the new package containing all six languages including COBOL. Keep your eyes and ears open for future status reports on this latest improvement to the SuperPET system.

POWER/PLAY

The Magazine for Commodore Home Computerists, Will Appear in June

You've been hearing about it, and now, in just a few weeks, you'll actually get to see it—POWER/PLAY—Commodore's exciting new magazine devoted exclusively to our rapidly expanding home computer audience. As the name implies, the new magazine will emphasize both the POWERful computing capabilities of Commodore's home computers and the PLAYful nature of home computing.

Subscribers to COMMODORE Magazine will be among the first to see POWER/PLAY, since we will be sending them a free premier issue as soon as it's off the presses in June. It will also be available at Commodore dealers at that time, with quarterly publication scheduled for the remainder of 1982.

Offering "fun, games and beyond with Commodore home computers," POWER/PLAY will provide home computerists with valuable information on new products, applications in the home, programming techniques, learning at home, telecommunications, users clubs, and just about anything else they'll need to know. Freelance articles, programs, photographs and cartoons will be included, as well.

Meanwhile, COMMODORE Magazine will continue to cover the many applications of Commodore products outside the home. Since many of our home computing products are not limited to home use, but are also used in education, science, business and other fields, you will continue to find information about those kinds of applications here in COMMODORE, rather than in POWER/PLAY.

Look for POWER/PLAY in June. It'll knock your socks off.■

Dr. Daley's Software Offer Extended

In the February issue, an advertisement appeared on page 88 for the WIZ, a data management system from Dr. Daley's Software. The ad featured a special limited-time offer for any user to receive trade-in credit on their commercially available data management program, upon purchase of the WIZ. Because the offer expired March 15, 1982, some of our readers may not have had sufficient time to take advantage of the offer. Therefore, Dr. Daley's Software is extending this special trade-in credit until JUNE 15, 1982. For an offer on your present system, just mention this article and call Dr. Daley's toll free number: (800) 548-3289.

Professional Business Software

For The Commodore 8000 Series Computer System

CMS GENERAL ACCOUNTING SYSTEM II:

A fully interactive General Accounting System designed especially for the first time user. All input requests are fully prompted with complete verification of input data. Most reports may be printed either to the screen or the printer and started or stopped at any point. The user is led completely through each function by a series of highlighted prompts fully explaining the required input at each point. A professionally written instruction manual is included which shows sample reports generated by the system and further explains each step and prompt as it is encountered by the user. These user prompts, together with the detailed step by step manual, make it virtually impossible for the user to accidentally crash the program or to get lost in the program and be unable to proceed or backup. Some of the many features of each of the four major accounting functions is shown below.

GENERAL LEDGER:

Up to a 1000 accounts on the Chart of Accounts. Fully departmentalized up to nine departments. Cash Disbursements and Cash Receipts Journal as well as a General Journal for ease of data entry. Maintains account balances for Present Month, Quarter to Date, and Year To Date. User customized financial statements. Accepts postings from Accounts Receivable, Accounts Payable, Payroll, or other programs.

ACCOUNTS RECEIVABLE:

Prints Invoices and Monthly Statements. The finance charge rate and period may be set by the user. Full invoice aging reports with aging breaks set by the user. During invoice data entry a copy of the Invoice is displayed on the screen and the information is typed in exactly as if the Invoice was in a typewriter. Accomodates full or partial invoice payments. Provides for Credit and Debit Memos as well as Invoices. Invoice File capacity is 2000 minus the number of customers multiplied by 1.4. Five hundred customers will allow room for 2100 invoices. Invoices may be distributed among up to nine different General Ledger accounts with automatic updating to the General Ledger.

ACCOUNTS PAYABLE:

Prints Accounts Payable checks with full check voucher detail for each Invoice paid. Prints detailed check register. Automatic application of Credit Memos. Complete invoice aging reports with aging breaks set by the user. Invoice File capacity is 2000 minus the number of vendors multiplied by two. Invoices may be distributed among up to nine different General Ledger accounts with automatic updating to the General Ledger Account File.

PAYROLL:

Maintains Monthly, Quarterly, and Yearly totals for each of up to 350 employees. Prints Payroll checks with full deduction and pay detail. Accomodates Weekly, Bi-weekly, Semi-Monthly, and Monthly employees. Pays regular, overtime, holiday, and piece work hours. Up to eight miscellaneous deductions or payments per employee. Prints Payroll Journal, Payroll Check Register, and an Absentee Report as well as 941 information and W2 forms. Automatic updating to the General Ledger.

See Your Nearest Commodore Dealer For A Demonstration

CMS Software Systems, Inc. 2204 Camp David, Mesquite, TX 75149 214-285-3581

"LEGAL TIME ACCOUNTING PLUS WORD PROCESSING FOR UNDER $6,500. FROM COMMODORE."

—WILLIAM SHATNER

"Before we had our Commodore computer and LTA system, there was no way to keep complete track of our time. And if you can't account for time, you can't bill for it. This system keeps track of every minute spent and issues very detailed invoices. It also keeps us posted on outstanding accounts receivable before they become past due. The truth is, we've had our Commodore computer less than a year and it's almost paid for itself already."

—Don Nelson, Attorney-At-Law
Pasadena, California

No one has to tell you that the practice of law runs on a sea of paperwork. Sometimes it seems like you're drowning in it. Fortunately, a simple, inexpensive solution is at hand: a Commodore computer. Including disk drive, letter quality printer, legal time accounting and word processing programs.

For a modest investment, you get all the features of a sophisticated word processor and a versatile business computer that can help your office operate more efficiently than you ever thought possible.

Commodore's Legal Time Accounting (LTA)[1], for example, keeps track of all the services you perform for your clients and handles billing automatically. It enables you to see instantly who is performing what services for which client and how much time is spent on each activity. Aging analysis reports reflect outstanding receivables over a 90 day period.

The system is easy to use, even for those with no computer experience. Lawyers simply fill out a log sheet at the completion of each activity. A date is entered into the LTA system which stores the information for as many as 500 clients. The data can be cross-referenced to produce nine different reports, from daily summary journals to client billing statements.

And with our Wordcraft 80 word processing program, your Commodore computer is versatile enough to be used whenever you'd normally use a typewriter. For briefs. Memos. Correspondence. In seconds, you can delete, insert, rearrange paragraphs, revise, with no time wasted typing multiple drafts.

In short, a Commodore computer can free up your legal staff to do what it does best... practice law.

Your Commodore computer can be expanded to meet the needs of a growing office. And Commodore dealers throughout the country offer prompt local service.
Visit your Commodore dealer for a hands-on demonstration of the computer that does so much, so easily, at such a low cost.

[1] Legal Time Accounting was created by Cimarron Corporation

Commodore Computer Systems
681 Moore Road, King of Prussia, PA 19406

□ Please send me more information on the LTA System.

Name ____________________
Address ____________________
City____________________ State ____________________
Zip ____________________
Phone ____________________

commodore
COMPUTER

Three New Disk Drives Introduced

Commodore has announced the introduction of three new disks systems to be available this Spring. All three designed to greatly enhance the power of your Commodore micro. Two of the new disks systems are 5¼ inch rigid Winchester technology disk drives, the D9060 and the D9090 and the third is a dual floppy disk unit, the 8250.

The new disk units allow use of an entire drive for a single relative or sequential file. Multiple drives can be connected to one computer, providing the file capacity needed to support large inventory or data base applications on Commodore computers. The D9060 has a 5 megabyte capacity and is priced at $2995. The D9090 has a 7.5 megabyte capacity and costs $3495.

The D9060 and D9090 are both single drive storage devices. The main components are read/write controls, drive motor electronicsand a drive mechanism. The 9060 has two platters with recording devices on both sides and four read/write heads. The 9090 has three platters with recording devices and six read/write heads.

The third new Commodore disk system is the 8250 dual floppy disk unit. The new disk drive doubles the capacity (2.1 megabytes) of Commodore's 8050 disk drive because it is able to use both sides of a single diskette. The 8250 can handle sequential or relative files of over one million characters.

The 8250 uses ordinary diskettes and is read/write compatible with the 8050. Most of the software that can be used on an 8050 can also be used with the 8250 without modification. The 8250 uses a 100 Track per Inch(TPI) two headed drive with a formatted capacity of 1,066,496 bytes (characters) per drive. The 8250 diskettes have 154 tracks, 77 on each side. The 8250 is priced at $2095.

The three new disks were designed around the concept of providing users with large file handling capabilities supported by the same BASIC program statements and DOS commands used with other Commodore disk units. All three of the new units are compatible with any of Commodore's PET or CBM series computers. And as with all of Commodore's disk systems the new disk drives are "intelligent" peripherals and do not use any of the computer's memory. The 8250, D9060, and the D9090 all conform to PET IEEE interface requirements.■

DISK UNIT SPECIFICATIONS

MODEL	D9060	D9090	8250	8050	4040
DRIVES	1	1	2	2	2
HEADS/DRIVE	4	6	2	1	1
STORAGE CAPACITY					
Unformatted	6.38 Mb	9.57 Mb			
Formatted	5.01 Mb	7.52 Mb	2.1 Mb	1.06 Mb	174 Kb
Sequential File	full disk capacity		2.1 Mb	1.06 Mb	174 Kb
Relative File	full disk capacity		1.04 Mb	182 Kb	171 Kb
DISK FORMATS					
Cylinders (Tracks)	153	153	77	77	35
Sectors/Cylinder	128	192	—	—	—
Sectors per track	32	32	23-29	23-29	17-21
Bytes per sector	256	256	256	256	256
Blocks Free	19442	29162	4166	2052	664
TRANSFER RATE					
Internal	5 Mb/s	5 Mb/s	40 Kb/s	40 Kb/s	40 Kb/s
IEEE-488 Bus	1.2 Kb/s	1.2 Kb/s	1.2 Kb/s	1.2 Kb/s	1.2 Kb/s
ACCESS TIMES (milli-seconds)					
Track-to-track	3	3	3	30	30
Average	153	153			
Head settling time	15	15			
Average Latency	8.34	8.34	100	100	100
RPM	3600	3600	300	300	300
File Interface (ALL MODELS): Shared memory — (2114) 4 Kb RAM					
PHYSICAL DIMENSIONS					
Height (in.)	5.75	5.75	6.5	6.5	6.5
Width (in.)	8.25	8.25	5.0	5.0	5.0
Depth (in.)	15.25	15.25	14.35	14.35	14.38
Weight (lb.)	21	21	28	28	28
ELECTRICAL (All Models)					
Voltage (VAC)	100, 117, 220, or 240 VAC				
Frequency (Hz)	50 or 60				
Power (watts)	200	200	60	50	50

COMMODORE NEWS

KUDOS
FOR CUSTOMER SUPPORT

Dear Customer Support:

Today I received the packet you sent me on the Commodore VIC 20. I had asked for the information to help my nephew get the most out of a VIC 20 he earned as a prize in a high school computer programming competition.

The information you sent is exactly what I had hoped for. I'll have it in the mail immediately so Cary can get to work on his VIC 20 right away. He was especially interested in how he could activate the audio functions. I am super pleased with your response. I know he will be too.

Thank you again for your superb *Customer Support*. It is everything the name suggests. Cary and I both truly appreciate your efforts.

Sincerely yours,

Ross W. Farnsworth
Cincinnati, Ohio

As promised in the February issue, we will continue to share some of the questions that are most frequently asked of our valuable Customer Support team. In addition, we are expanding our presentation in this magazine to include queries about the entire Commodore product line. Remember, this comprehensive list of problems and solutions is for your benefit. So the next time you have a problem, think twice before you call for assistance. The answer may very well be in this or future issues of Commodore Magazine.

Frequently Asked Questions . . . and Answers!

PET

1. Q: How can the Pet 2001 be upgraded?

A: Commodore 2000 series machines can be upgraded by changing the BASIC ROM chips. The original 4K and 8K PET computers do not have ROMs to convert the operating system beyond BASIC 3.0. ROM chip sets (called ''BASIC Upgrade ROMs'') are available from most authorized Commodore dealers. Benefits to be gained by upgrading the machine are disk commands, enhanced BASIC interpreter, and a wider range of compatibility with available application software.

2. Q: Where can information be obtained about the PET 2001 parallel user port?

A: Information about the parallel user port of the PET 2001 can be found in *The PET Revealed* or in *PET Interfacing* (published by Howard W. Sams Co., Inc.). These books can be purchased from your authorized Commodore dealer.

3. Q: Can a 4016 be upgraded with additional RAM?

A: Commodore does not recommend or support memory upgrade modifications, although there are some vendors who make an expansion memory system available. Such a device connects through the memory expansion port and is available from Skyles Electric Works; Mountain View, CA; (415) 965-1735.

4. Q: Will U.S. PET/CBM computer work without adaption in Europe?

A: PET/CBM computers are not compatible with the European elecrical system, but Commodore offers a full line of PET/CBM computers and peripherals in our European markets.

5. Q: Where can User Groups be located for Commodore machines?

A: The Commodore Microcomputer Magazine lists any known user groups in each issue.

6. Q: Is the character set now in the PET/CBM encoded in the memory chip?

A: The character set is contained in a chip called the character generator ROM (socket #UF10 in the 2001 and socket #UA3 in the Fat Forty and the 8032).

7. Q: How can a TV set or a monitor be connected to a PET for use in a classroom?

A: The computers with a nine inch screen require a video adapter to generate a composite video signal and a monitor. Units with a twelve inch screen require the video adapter and a high resolution (16 MHz or greater) monitor. This device is available from Madison Computer; Madison, WI; (608) 255-5552, or from CA Computer Systems; Sunnyvale, CA; (408) 734-5811.

8. Q: Using two PETs, a disk drive, and a MUPET, how can two programs be merged?

A: Two programs can be merged by using one of the utility ROMs that add a MERGE command to BASIC. Several of these utilities can be found in the ''Firmware'' section of the *Commodore Software Encyclopedia/Second Edition.*

9. Q: what is the difference between the MUPET and a Procter system of networking?

A: The MUPET is a hardware-only, bus-switching device. The Procter is a software/firmware system which is designed for use in a school environment. It allows for passwords and file access control, among other features.

10. Q: Can a repeating key be created with the graphics keyboard of the 4016 and the 4032?

A: A repeating key can be created on the 4016 and the 4032 if you write a machine language program for our version of BASIC. There is also a 4K ROM (called the Machine Language Utility-Pac) that will fit into an empty ROM slot of the machine which has an Auto-Repeat command included. More information about the Machine Language Utility-Pac is published in the *Software Encyclopedia/Second Edition*.

CBM

1. Q: To which machines can the 64K Memory Expansion Board be connected?

A: The 64K add-on board can be used in conjunction with the 8032. It also works with the Fat Forty computer that is equipped with the universal logic board.

2. Q: What is the maximum usable RAM on the PET/CBM?

A: Both the PET and the CBM have 31,743 bytes of RAM available for use in programming.

3. Q: Where can a program be obtained to utilize the Voice synthesizer?

A: The February 1982 issue of the Commodore Microcomputer Magazine has a program that will utilize the Voice Synthesizer.

4. Q: What is the MUPET and the REGENT?

A: The MUPET and the REGENT are devices that allow several computers to share a common disk drive. These devices switch the bus from computer to computer as the situation requires. Since they are hardware-only devices, they use no RAM in any of the computers.

5. Q: Do Commodore machines have CP/M compatibility?

A: At this time, Commodore doesn't sell this product, but there are units available such as the Softbox from Small Systems, Inc.; Mountain View, CA; (415) 964-8201, and the Z RAM from Madison Computer; Madison, WI; (608) 255-5552. The Softbox attaches like a peripheral outside of the computer, while the Z RAM mounts inside of the computer cabinet.

6. Q: Can a typewriter be interfaced with an 8032?

A: Some typewriters can be interfaced with the 8032 with difficulty. For instance, a program must be written to translate characters to the kind required by the typewriter, and circuit boards may have to be changed.

7. Q: What is the procedure to get into the graphics mode?

A: To get into the upper-case/graphics mode on the 8032, simply PRINT CHR$(142). To get back into normal upper/lower case mode, PRINT CHR$(14).

8. Q: What clock rate is used on the 8032? Why isn't a 4 MHz used?

A: The 8032 uses a 1 MHz clock. The internal architecture of the 6502 chip is more efficient than that of other microprocessors in that the instruction fetch/decode cycle overlaps the execution cycle. Thus a 1 MHz 6502 is faster than a 2 MHz 8080 or Z80 microprocessor.

9. Q: Where is the cursor address stored in the 8032?

A: On the 8032, the column position of the cursor is decimal address 198. The row position is decimal address 216. This information was obtained from the memory map in the October 1981 issue of the Commodore Microcomputer Magazine.

10. Q: How can the machine language monitor be entered?

A: To enter the machine language monitor, use the SYS command to jump to any location containing a binary zero. For example: SYS 4 or SYS 1024. This works on any PET/CBM computer that contains the built-in machne language monitor. This includes all but the first machines—and those machines that have a monitor on tape.

SuperPET

1. Q: What is resident in the SuperPET that allows for multiple high-level languages?

A: The ROMs in the SuperPET contain routines that handle system functions such as input/output, screen handling, keyboard input, and general utility functions. These routines interface with the operating system for the particular language that you load into the computer. This allows different languages to be loaded which use the same utility routines.

2. Q: Will the languages available for the SuperPET work with a 4040 disk drive?
A: The languages will work, but the diskettes supplied are in the 8050 format, so they would have to be copied down to 4040 format first.

3. Q: Can an 8032 be upgraded to be a SuperPET?
A: In the future, a "single-board" upgrade will be available from your Commodore dealer. This will include language disks and the manuals.

4. Q: Why won't VISICALC and other software that uses ROM chips run on the SuperPET?
A: The add-on memory occupies the same address space as ROM slot #UD12. The 6502 "sees" RAM there instead of the ROM chip. A dealer-installed upgrade is available which will allow use of software that requires a plug-in chip. The part number for this upgrade is 9000030.

5. Q: Will the high-resolution graphics made by Micro Technology Unlimited run on the SuperPET?
A: The MTU graphic board will not run on the SuperPET because the display RAM occupies the same address space as the add-on memory.

6. Q: Are the SuperPET languages (FORTRAN, APL, BASIC, PASCAL, and Assembler) limited subsets of the languages?
A: All of the languages included with the SuperPET are complete languages written by the University of Waterloo.

7. Q: How does the RS232 serial port operate on the SuperPET?
A: For a detailed explanation of the operation of the RS232 serial port, refer to the February 1982 issue of the Commodore Microcomputer Magazine.

8. Q: Is debugging available for the languages included with the SuperPET?
A: Debugging is included with all of the SuperPET languages. Debugging allows the computer to work interactively with the programmer in editing and correcting the program.

9. Q: How much memory is used when the languages are inserted into internal memory of the SuperPET?
A: The high-level languages load the interpreter into the upper 64K of RAM and leave the lower 32K of RAM for programming.

10. Q: Are the SuperPET languages stored as interpreters or as compilers?
A: All of the Waterloo languages at this time are interpretive languages which means that the computer "compiles" and executes the program on a line-by-line basis. This saves time in the debugging process.

VIC 20 Personal Computer

1. Q: What kind of BASIC is used on the VIC 20 and how is it different from PET BASIC?
A: The VIC 20 uses PET BASIC 2.0, and there are no differences in the syntax between the two.

2. Q: Can software programs made by other vendors (Atari/Radio Shack, etc.) be utilized by the VIC 20?
A: In general, software that has been written for a particular computer can be used only on that computer. Therefore, software programs made by other vendors cannot be used on the VIC 20.

3. Q: How can programs saved on the 8032 be loaded to the VIC and vice versa?
A: The VIC will load PET/CBM programs from tape without any modification if there is enough memory installed in the VIC. To load a PET/CBM with a VIC program, first POKE 41, 16, then POKE 4096,0, then type NEW and proceed to load.

4. Q: Can software for the VIC bought in another country be used with an American VIC 20 unit?
A: Software bought in another country for the VIC is compatible with American units.

5. Q: How is a MODEM connected to the VIC 20?

A: Two choices are available. One way is to purchase the VIC RS232 Terminal cartridge and use existing RS232 acoustic MODEMs. If, however, you do not have an acoustic MODEM, your best bet is to get the VIC MODEM, which will soon be available from your authorized Commodore dealer. With this inexpensive cartridge, you simply insert the handset cord of your modular phone directly into the MODEM, run the software driver, and you are set for the world of computer telecommunications!

6. Q: When using the VIC MODEM, can information be saved to the disk drive or the printer?

A: Information can be saved to both the disk drive and the printer while using the VIC MODEM.

7. Q: Is it possible to use multiple joysticks with the VIC?

A: Only one joystick can be plugged into the game control port. Any additional joysticks must be interfaced to the user port with the appropriate software.

8. Q: Is it possible to hook up an audio cassette recorder to the VIC?

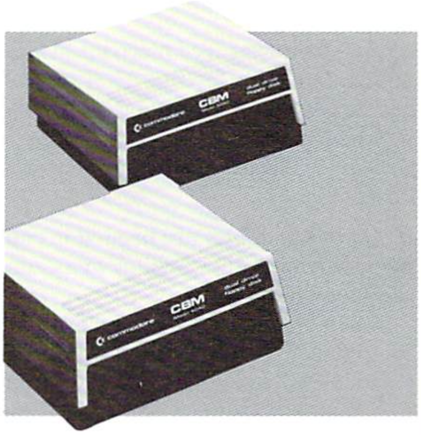
A: A standard audio cassette recorder will not work with the VIC. A VIC Datassette is necessary for use with any cassette tape.

9. Q: How can VIC schematics and a VIC Memory Map be obtained?

A: Schematics, memory map, and other technical information for the VIC 20 are included in the Programmer's Reference Guide, which is available from your Commodore dealer.

10. Q: Can more than one disk drive be connected to the VIC while using the printer?

A: Up to five disk drive units can be "daisy-chained" together. To include a VIC Printer in the system, simply connect it as the last unit of the chain.



Disk Drives

1. Q: What do the lights on the disk drive signify?

A: The light on the top center of the unit indicates an error condition if it turns red and stays on. (Flickering is normal during operation.) The lights on each drive indicate activity in that particular drive.

2. Q: There is a buzzing noise during initialization of a disk — is something wrong with the disk drive?

A: The noise is normal. The head mechanism is creating the noise by vibrating the head against the travel limit stops. This ensures that track number one has been located.

3. Q: What kind of diskettes should be used with the disk drive?

A: We suggest using double-density diskettes with the 8050 and the 8250 because they tend to be slightly more reliable than others. However, any good-quality, soft-sectored, single-density diskette will work well with any model disk drive.

4. Q: How can the disk drive be cleaned?

A: The disk drive can be cleaned by using a commercial head-cleaning diskette and cleaning fluid. These products are available from most computer or office equipment stores.

5. Q: What is the cause of READ and LOAD errors?

A: A number of things can cause errors. Possible causes include: 1) improperly seated diskette; 2) physical or electrical damage to the diskette: 3) dirty heads in disk drive; and 4) drive head misaligned or speed of rotation not properly set. Your Commodore dealer is equipped to evaluate these problems if they persist.

6. Q: How can the 2040 disk drive be upgraded to a 4040 disk drive?

A: A 2040 can be upgraded by simply replacing the ROM chips on the main logic board. Consult your Commodore dealer for ordering information.

7. Q: Can the 8050 disk drive be upgraded to an 8250 disk drive?

A: The 8050 cannot be upgraded to an 8250; however, the diskettes are compatible. Of course, the 8050 only uses (reads/writes) the bottom side of the diskette. The 8250 uses both sides. Therefore, it is the bottom side of the diskette that is compatible with both drives.

8. Q: Can a 4040 dual disk drive read a diskette made on a 2040 disk drive?

A: A 2040 diskette can be read from a 4040 disk drive, but it cannot be written on. This means that a 4040 drive is read compatible but not write compatible with a 2040 diskette.

9. Q: How can a 4040 diskette be copied to an 8050 disk format for use on the 8050 drive?

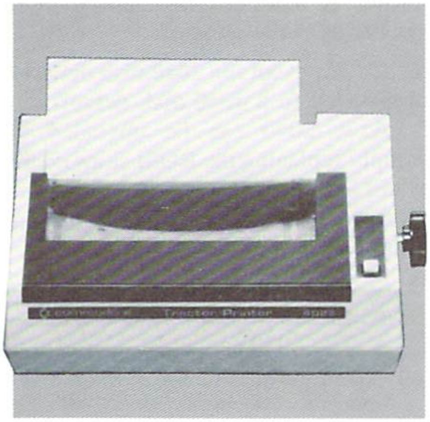
A: The following procedure lists the proper steps: 1) Turn on the computer and the 8050 drive only; 2) Run the "Change 8050" program (included on the Test/Demo diskette); 3) Turn on the 4040 drive; 4) Run the "Unit to Unit" program (found on the Test/Demo diskette). The Unit to Unit program will copy all of your files from the 4040 diskette to the 8050 diskette.

10. Q: How many files will each diskette hold?

A: A 4040 diskette is capable of saving a maximum of 144 files. An

→

8050 diskette is capable of saving a maximum of 224 files. These files may be either sequential, program, relative, or a combination. Once that limit has been reached, the directory is full—even if the rest of the disk is not.



Printers

1. Q: What does the blinking light signify on the 4022?

A: When the red light blinks, it indicates that the printer is out of paper or that the paper is not inserted correctly.

2. Q: Are there bi-directional printers available from Commodore?

A: There are printers available with the bi-directional feature such as the 8023p, the 8300p, and the 8024. The 4022 printer can be upgraded by replacing ROM with an upgrade ROM. The part number of this upgrade ROM is 901631-02.

3. Q: Is the 4022 upward compatible with the 8023?

A: Programs that work with a 4022 will also work with the 8023.

4. Q: Will an 8032 accept two printers?

A: The 8032 can be used with up to four printers connected.

5. Q: Is the 8300 printer supplied with a tractor-feed mechanism?

A: The tractor feed mechanism is an optional feature for the 8300 printer, and it allows bi-directional (up/down) movement of the paper. This feature is available from your Commodore dealer.

6. Q: What cables are needed to connect a Commodore printer to a Commodore computer?

A: In order for the first peripheral to be connected to a Commodore computer, a PET to IEEE cable is required. For every additional peripheral, an IEEE to IEEE cable is required. For this reason, the cables are sold separately from the peripherals.

7. Q: Do any Commodore printers have graphics capability?

A: The Commodore 1515, 2022, 2023, 4022, and the 8023 printers all support PET graphic characters.

8. Q: Do any Commodore printers print characters with true ascenders and descenders?

A: With the exception of the VIC 1515 printer, all Commodore printers are capable of printing characters with true ascenders and descenders.

9. Q: What type of printer heads do the 2022 and the 4022 printers use, and what is their life expectancy?

A: The 2022 printer uses a print head that can be repaired. The 4022 printer uses a print head that can be replaced once it is worn out. The life expectancy of each is approximately 60 million characters.

10. Q: What is the procedure to get an automatic line feed on an 8300 printer?

A: Under the front cover of the 8300, there are two rows of pins next to the small switch block. To get an automatic line feed, the second pins from the left on each row should be connected together using a jumper block. If the jumper block is not supplied with your printer, it can be obtained from your Commodore dealer.■

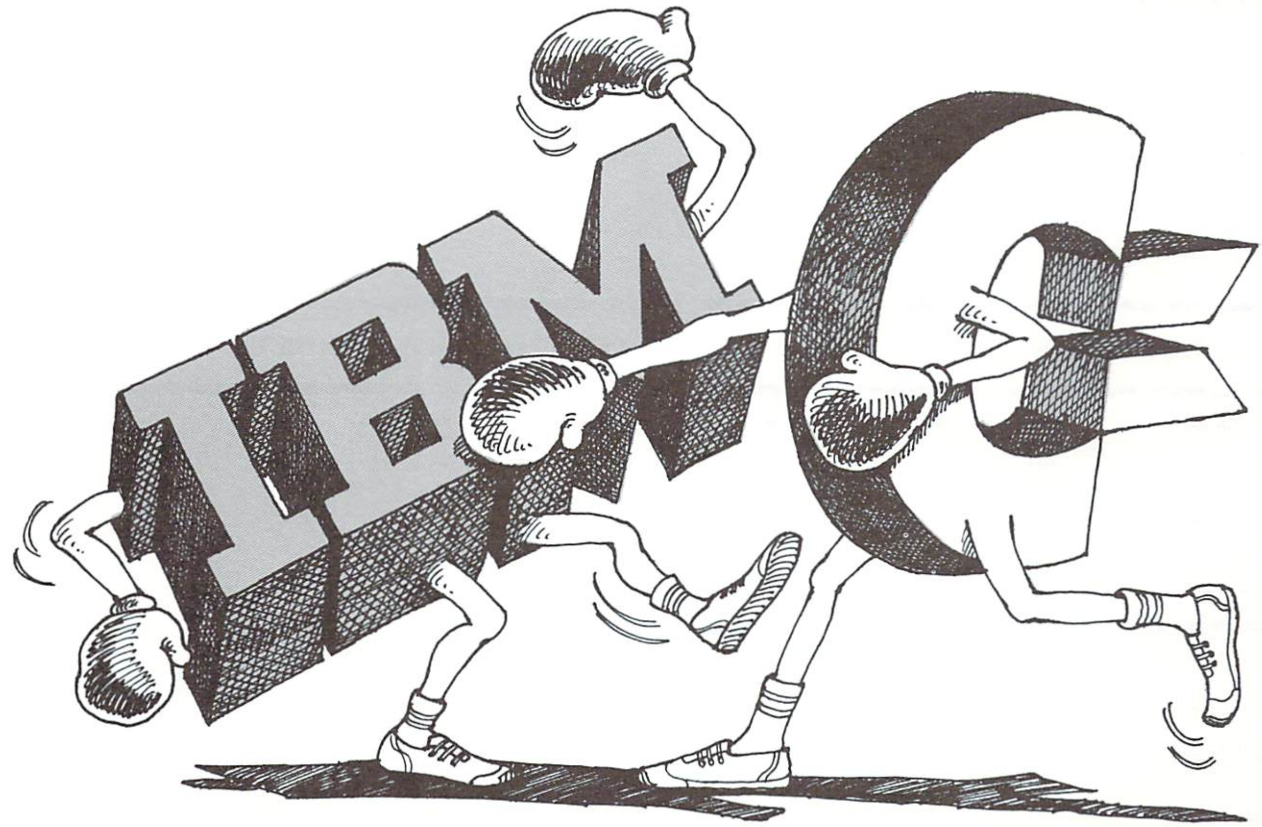
ATTENTION PROGRAMMERS

Commodore is compiling a list of software written for our computers. If you have software you would like to have included in this listing please submit the following for review:

- copy of program on disk or tape
- documentation describing purpose and utility of the program
- specify equipment necessary for program operation
- information on price of program and where the program can be purchased

Please submit this information to:

SOFTWARE Committee
Commodore Business Machines
681 Moore Rd.
King of Prussia, PA 19406



SuperPET vs. IBM: A Close-up Look

The recently announced IBM* Personal Computer is a modular, desk-top system intended by IBM to compete in the same small business markets as Commodore, Apple and Radio Shack computers. The IBM system features an 8-bit microprocessor with up to 256k-bytes of memory. Required IBM peripherals include a keyboard and monochrome or color video display.

Optional peripheral devices offered by IBM include a second video display, dual floppy disks, a dot-matrix printer and an RS-232C serial communications adapter. BASIC and Pascal programming languages are available. A customer-supplied cassette tape recorder is supported.

The Commodore SuperPET provides an excellent opportunity in competitive bidding vs. the IBM Personal Computer. The SuperPET offers dual 8-bit microprocessors with 96k-bytes of memory, integral keyboard and video display, PET/IEEE and User-Parallel ports, and an RS-232C serial port as standard equipment.

Optional Commodore peripherals for the SuperPET include a cassette tape recorder, a variety of printers—both dot-matrix and letter quality, and a choice of single or dual floppy disk units. Five high-level languages (two BASIC's, FORTRAN, Pascal, APL) and two Assemblers are available for the SuperPET.

Commodore and IBM both offer 90-day warranties on system hardware. An Extended Warranty is available from IBM. Both manufacturers can offer annual maintenance contracts. Either 'carry-in' or on-site service is available for both systems.

The IBM Personal Computer is offered in two standard configurations. Configuration #2 (most nearly comparable to the SuperPET) includes the CPU with 64k-bytes of RAM, internal speaker, cassette interface, RS-232C serial port, keyboard and two 160k-byte diskette drives. Video display is not included.

Use of the IBM diskette drives, RS-232C port and lightpen/joysticks requires separate purchase of their Disk Operating System (DOS) diskette. Software support for the SuperPET disks and RS-232C port are built-in. The IBM Pascal Compiler (128k-bytes RAM required) and Linker are also purchased separately. All of the SuperPET language systems (except 6502 Assembler) and DOS support are included with purchase of the CPU.

The IBM System Unit (CPU) provides five slots to plug circuit cards into for memory or peripheral device expansion. Since maximum memory and peripheral expansion would require eight slots, trade-offs must be made in configuring the IBM computer for a given application. Those options which require expansion slots are marked with "*" in the accompanying chart.

The Commodore SuperPET and the IBM Personal Computer are both new systems, being introduced at about the same time. Currently less than a half-dozen off-the-shelf, end-user application programs are available fom IBM for their system. The SuperPET, being downward compatible with the CBM 8032, supports a wide range of readily available applications in many fields.

**IBM is a registered trademark of International Business Machines*

Hardware Features Compared

Feature Description	CBM Super PET	IBM Computer
Microprocessors	MOS 6502 and Motorola 6809	Intel 8088
System Clock Speed (MHz)	1.0 (both)	4.77
Memory Capacity: K-bytes		
Read-Only (ROM)	36	40
Read-Write (RAM)	96	16 to 256
*Memory Expansion Cards		32k or 64k
RAM Memory Speed—Nanosec	300	410
Parity Checking	No	Yes
Sound—Internal Speaker	Yes	Yes
Screen Size (columns x lines)	80 x 25	80 x 25
* Green Phosphor Display	Yes	Yes
*Color Display	No	Customer-supplied
RF Modulator		Customer-supplied
Keyboard	Integral 73-key	Separate 73-key
*Matrix Printer (bi-directional)	Yes	Yes
Print Speed (Model 4022P)	80 cps (80 col)	80 cps (80 col)
(Model 8023P)	150 cps (136 col)	No
(Model 8024)	165 cps (136 col)	No
Letter Quality Printer (8300P)	Yes	No
Print Speed (bi-directional)	To 40 cps	
*Floppy Disk Units	External (IEEE)	Internal
Max Nr. of Drives	1-8 Units	1 or 2 Drives
Formatted Disk Capacity (K-bytes)		
Single-drive (Model 2031)	170	160
Dual-drive (Model 4040)	340	320
Dual-drive (Model 8050)	1050	
Dual-drive (Model 8250)	2100	
Hard-disk Units (1-4 drives)		
Capacity: (Megabytes)		
(Model D9060)	5.0	
(Model D9090)	7.5	
*RS-232C Serial Port	Yes	Yes
Asynchronous Baud-rate	50-9600 bps	50-9600 bps
User-Parallel I/O Port	Yes	No
IEEE-488 I/O Port	Yes	No
Cassette Tape Recorder	Yes	Customer-supplied
*Joysticks/Lightpen	No	Customer-supplied

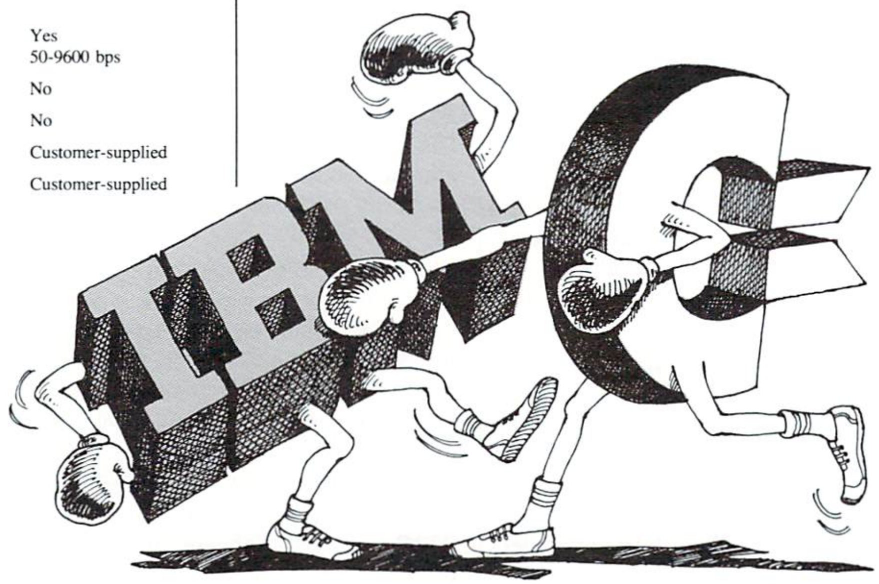
Price Comparison — SuperPET vs. IBM Computer

SuperPET Standard Configuration	Unit Price	Total
Central Processor Unit (CPU)	1,995	
Dual Floppy Disk Unit (4040)	1,295	
Dot-Matrix Printer (4022)	795	
Total $		4,085

Includes: Dual Microprocessors, 96k-bytes RAM, keyboard(73 keys), video display (80 x 25), RS-232C serial port, dual floppy disks (340k-bytes), matrix printer (80 cps, bi-directional) and all SuperPET™ language systems. Higher capacity disk units and faster printers available from Commodore.

IBM Standard Configuration #2		
System Unit/Keyboard (CPU)	3,045	
Memory Expansion Option - 32k	325	
RS-232C Serial Port Adapter	150	
Diskette Drive Adapter	220	
Monochrome Display & Printer Adapter	335	
Monochrome Video Display	345	
Dot-Matrix Printer	755	
Disk Operating System (DOS) Diskette	40	
Total $		5,215

INCLUDES: Processor, 96k-bytes RAM, keyboard (83 keys), video display (80 x 25), RS-232C serial port, two floppy disks (320k-bytes), matrix printer (80 cps, bi-directional), DOS diskette (required to use diskettes & RS-232C) and a BASIC language interpreter.



"MEDICAL ACCOUNTING PLUS WORD PROCESSING FOR UNDER $6,500. FROM COMMODORE."

—WILLIAM SHATNER

The symptoms are common. Missing receipts. Overdue invoices. Neglected insurance forms. And, worst of all, a lot of precious time spent on paperwork that could otherwise be devoted to patient care.

The cure: A Commodore desktop computer. Including disk drive, letter quality printer, and complete medical accounting and word processing systems. For a modest investment, you get all the features of a sophisticated and versatile business computer that can do virtually all your paperwork in a fraction of the time it takes you now.

Commodore's Medical Accounting System (MAS)[1], for example, can provide you with a fast, flexible accounting and bookkeeping system that's as easy to use as it is cost effective. Automating your receivables, invoicing, aging of payables, and revenue analyses. MAS can also generate end-of-the-month "Superbills" as well as standard insurance and Medicare forms. And it gives you a thorough overview of your office activities through a series of reports ranging from diagnostics to referrals.

And with our word processing programs, your Commodore computer is versatile enough to be used whenever you'd normally use a typewriter. For memos. Reports. Correspondence. Proposals. In seconds, you can delete, insert, rearrange paragraphs, even revise as many times as necessary. With no time wasted typing multiple drafts.

If all that time saved on paperwork is used to take on additional patients, just think how quickly your Commodore computer will pay for itself, many times over.

Your Commodore computer can be expanded to meet the needs of a growing office. And Commodore dealers throughout the country offer prompt local service. Visit your Commodore dealer for a hands-on demonstration of the Commodore computer that does so much, so easily, at such a low cost.

[1] Medical Accounting System was created by Cimarron Corp.

Commodore Computer Systems
681 Moore Road, King of Prussia, PA 19406

☐ Please send me more information on the MAS System.

Name ____________

Address ____________

City ____________ State ____________

Zip ____________

Phone ____________

commodore
COMPUTER

Every PET™ Needs a Friend.

CURSOR is the best friend your Commodore PET™ will ever have. Since July, 1978 we have published over 160 of the most user-friendly programs for the PET available anywhere. When we write or edit a program, we spend lots of time fussing about how it will treat you. We pay attention to lots of little things that help make using a computer a pleasure instead of a pain.

Naturally, **CURSOR** programs are technically excellent. Each program that we purchase is extensively edited or rewritten by a professional programmer. But imagination is just as important as being user-friendly and technically good! We delight in bringing you off-beat, unusual programs that "show off" the abilities of your PET or CBM.

CURSOR is user-friendly, technically great and full of imaginative programs. And every issue of **CURSOR** is still available! We continue to upgrade previously published programs so that they'll work on the three varieties of Commodore ROM's (Old, New, and 4.0). All issues from CURSOR 23 on also work on the 80 column CBM. Please note, however, that CURSOR programs do *not* work on the VIC-20 computer.

For only $5.95 you can buy a sample tape and judge for yourself. Each **CURSOR** comes to you as a C-30 cassette with five programs and a graphic Front Cover, ready to LOAD and RUN on your PET.

Who knows? After your PET meets **CURSOR** things may never be the same!

Published By: THE CODE WORKS

Box 550
Goleta, CA 93116
805-683-1585

Cover1	Dots bounce over your screen.
Brick	Throw a brick as close to the window as possible. Addictive!
Wander	A (drunk) worm crawls over the screen of your PET.
Shark	You are a great white shark that eats people. Ouch.
Plot	Demonstration of plotting.
Typer	A simple PET typewriter demo. (For small-keyboard PETs)
Cover2	Snakeskin: an animated geometrical design.
Race	A one or two person auto race game.
Zap	You control a missile and try to hit targets.
Est	Estimate length and area in metric units.
Guess	Try to trap a number the computer selects.
Mad	Fill in the blanks of five crazy stories.
Cover3	Confusion - a random walk about town.
Bar	A vertical bar graphing program.
Dots	The game of Dots - you against the computer.
Quix!	Memory game with great sound effects.
Flash	Review "flash cards" in random order.
Cards	Produces the tapes for the Flash program.
Cover4!	Animated piano keyboard with three catchy songs.
Bop	Demonstrates Chisanbop, the Korean technique of counting.
Calc	Calculator for hex, octal, decimal. (Integer math only.)
Clock!	Digital alarm clock with sound.
Inp	"Return-proof" subroutine to replace "INPUT".
Ced	A simple text editor for old roms only.
Cover5	Press a key and display it on the screen as seen by the Pet.
Hanoi	You move a "tower" from one peg to another.
Shoot	Shoot a gun at a falling target.
Bship	A super game of Battleship - you against the computer.
Face	Draw almost any face. (You, too, can be a police artist?)
Hman	The word game Hangman.
Cover6	Loop the loop.
Box	Difficult logic game. (It may drive you crazy!)
Bjack	Double down, take insurance and bet your paycheck.
Pack	Eliminate blanks from your Basic programs. (Old ROMs only)
List	Makes that PACKed program readable again. (Old ROMs only)
Cedpr	Prints files produced by CED Cursor Editor. (Old ROMs only)
Cover7	Kaleidoscope pattern.
Pricer	Job cost estimation.
Sound!	The CURSOR sound effects library.
Mind	Try to guess a pattern of four colors.
Fball	Football on your PET.
Paper	Become a famous wallpaper designer.
Cover8	Weaves patterns on the screen.
Revers!	You against the PET or another player. (Othello)
Dbook	An easy-to-use date book for birthdays, etc.
Space!	Fly your PET into deep space as you pursue the enemy.
Maze	Find all the hidden gold in a maze.
Add	Addition drill that makes the concept of "carrying" clear.
Cover9	Spiral pattern.
Yahtzee	Yahtzee for one to four players.
Slot!	The Cursor Casino, with sound!
Flip	Helps convert text from old to new character generator ROMs.
Circle	Great circle navigation. How far to *your* favorite vacation spot?
Gammon	16K game of Backgammon, you against the PET.
Cover10!	Musical cover with visual effects to accompany the music.
Titrate	Practice titration. A fine example of Pet graphics!
Finance	Calculate mortgage, pension plans, savings, etc.
Course	Obstacle course, with varying degrees of difficulty.
Asm	An assembler for the 6502.
Reader	Turns machine code into DATA statements.
Cover11	A fireworks display.
Demon!	Capture the Demon.
Hi Calc	A high-precision calculator.
Wipeout	Roll the dice and try to eliminate the 12 numbers.
Peg	A clever pegboard game.
States	Learn the states and capitals of the U.S.
Cover12	Symmetrical graphic designs.
Canyon!	Train to become a starship pilot.
Gauss!	The PET drops 1000 balls an hour into one of 12 bins.
Pickup	Gather all of the dangerous chemicals.
Piegram	You and the PET throw pies at one another.
Flight!	Canadian astronauts land on the moon. Animated cartoon.
Cover13	Two fish swimming in the ocean.
Ratrun	You are a rat in a perspective maze. Difficult (but fun).
Cars	Maintain cost records for one or more vehicles.
Ferry!	Ferry supplies across the asteroid belt.
Leap	Challenging logic game.
Tcard	Helps you wade through mounds of time cards.
Cover14	An optical illusion. Watch carefully!
Match	The game of concentration.
Search	Generate excellent word search puzzles. (Uses Pet printer.)
Bat!	Control the Super Bat, overcoming gravity and inertia.
Morse!	Use this program to learn Morse code.
Cops	In this two-person game the Cops chase the Crooks.
Coming	Amaze your friends with this delightful hoax.
Cover15!	A Christmas tree, with traditional Christmas carols.
Dungeon	Search for gold and avoid the beasts in a dark dungeon.
Fifteen	A classic puzzle.
Gomoku	Try to get five stones in a row.
Everest	Go climb a mountain. Watch for snowstorms and avalanches!
Hawaii!	Two guys go to a tropical isle. (Animated cartoon.)

Cover20	Plaid pattern.
Music!	The toy piano of the future.
Bets	Play "indian poker".
Checkers!	The Pet plays a modest game of checkers with you.
Curves	Display interesting patterns on the screen of your Pet.
Equip	Maintain an equipment inventory.
Cover21	The Cursor Chorus!
Capture!	Two wild beasts pursue you. Surround them with obstacles!
Dance!	An animated "Rain Dance" cartoon.
Boswain	A computer mystery: where *are* the rubies, anyway?.
Ouranos!	Warfare with the ultimate weapon: weather!
Drag	Addition and subtraction drill.
Cover22	The Waves of Change?
Kalah	Try to get all of the stones into your Kalah.
Poker	Try your hand against our dealer.
Match	Remember where the numbers are.
Thunt!	Find the hidden treasure while evading the robots.
Compare	Compare two programs (on disk) and display differences.
Cover23	Consider the spider...
Recipe	Convert your favorite recipe so you can feed an army.
Ambush!	Isolate your enemy, don't get trapped.
Orrery	The solar system in action on your PET.
Enigma	Your very own encryption machine.
Mwhiz!	Quick! What's two plus two?
Cover24	A profusion of patterns...
Defend!	Fight off invaders from four directions.
G-Word	Guess the secret word with a little help from your PET.
Racer!	How fast can you ride and avoid the side?
Printsit	Create pictures on the screen and save them on paper.
Re-Num	Renumber your entire disk file, or just a portion of it.
Cover25	Try to trap all the gas molecules.
Maxit	Can you maximize your score in this logic game?
Mail	An easy-to-use mail list utility.
Rescue!	Fly your own space shuttle and rescue survivors.
Repair	Maintain an inventory of equipment to repair.
Xref	Cross-reference your PET Basic programs.
Cover26	A PET volcano explodes on your screen.
Lawn!	Can you mow the lawn before you run out of fuel?
Tank!	A two-person shoot-em-up.
Ram	Who can get across first?
Safe!	Can you find the combination before the police find you?
Test	Calculates statistics for a set of test scores. (Needs 16K)
Cover27	Music on the screen of your PET.
Attack!	Protect your treasure from the enemy.
Emaze!	Find your way through a fast-changing electronic maze.
Duel!	Two-person strategy board game.
Miser	Can you find the treasure while you explore all the rooms?
Prochar	Design programmable characters for Commodore printer.
Cover28!	Watch the weaver at a loom.
Skeet!	Shoot those clay pigeons!
Blasto!	You have two minutes to destroy the mines!
Stop!	Can you reach the top first? (But don't push your luck...)
Voz	Which chess piece is which? (Very challenging.)
Flags	Practice your semaphore code.
Cover29	The juggler. (Such coordination!)
Pong!	Four *very* fast arcade-style games.
Clone	They are out to get you. But—which one is *you?*
Drone!	Attack the enemy with pilotless aircraft.
Contact	A complex (but fun) dice game.
Krypto	Select a function of five numbers that yields a given result.
Cover16	As patterns appear press a key to fill them in.
Nab!	Can you escape a head-on crash?
Fire!	You are the pilot of a fire department helicopter.
Aliens!	Protect Earth from the attacking aliens.
Bonzo!	It's chutes and ladders, with great graphics.
Catch!	Play catch with the Pet. (Not easy, however!)
Cover17	Flashing rectangles fill the screen.
Police!	You must protect the city against a crime wave.
Spot	Who can stack four markers in a row first?
Ruler	For kids: learn how to read a ruler.
Letter	Big numbers and letters appear when you press a key.
Merge	Merges two Basic programs. (Requires disk).
Npack	Squeeze blanks out of Basic programs. (New ROMs only).
Cover18	Blocks are stacked and unstacked in this graphic design.
Dromeda!	Cursor Creature Feature: a cartoon in black and white.
Joust	Put on your armor and challenge a friend to a joust.
Weather	A program that may help you predict the weather.
Hi-Res	High-resolution graphics on a small section of your screen.
Sheep	Herd the sheep into the barn before they eat the crops.
Cover19	Perpetual steps.
Frog!	An animated frog that is hungry.
Godzilla!	Save Japan from the King of the Monsters.
Miner!	Grab your pick, drill and some dynamite to mine for gold.
Rail	You are the engineer in this tricky train yard.
Gradebook	Program to maintain teacher's gradebook. (Uses printer.)

Individual CURSOR tapes are $5.95 each in the U.S. and Canada. Any assortment of 12 or more tapes only $4.95 each.

$ ________ CURSOR tapes. Issues: ______________________________
$ ________ California residents add 6% state tax.
$ ________ Foreign orders add $2.00/tape for shipping and handling.
$ ________ TOTAL enclosed.

Make checks payable to: **The Code Works**, Box 550, Goleta, CA 93116 USA. Terms: Cash, checks, money orders and credit cards accepted. Payment must accompany purchase orders. VISA and MasterCard will be charged shipping.

VISA ______ Mastercard ______ Card Number ______________________________

Signature ______________________________ Expires ______________________

Name ____

Address ____

City ______________________________ State __________ Zip ______________

EDUCATIONAL RESOURCE CENTERS

Are you interested in becoming part of a network of educators who can help share information with present or potential users of Commodore computers. Commodore wants to create a spirit of cooperation with institutions and microcomputer laboratories where Commodore products are used for education applications. The end result of this combined effort will be the creation of EDUCATION RESOURCE CENTERS throughout the county.

Are You Qualified?
Can your institution provide assistance in these areas:

- Forming of local education user support groups.
- Supplying news items, articles, and photographs about your computer applications.
- Working with local Commodore dealers to demonstrate education applications.
- Providing occasional workshops or seminars for other teachers or for the public.
- Submitting samples of new public domain software.
- Offering assistance in analyzing and documenting software.

What Are the Benefits for You?
Commodore will provide you with the following:

- Samples of public domain software which other user groups have submitted.
- Special Commodore Education Resource Center Newsletters.
- Mention of your organization and its resources in *Commodore Magazine* and Software catalogues.
- Publishing of articles which you submit, describing the activities of your school
- News of other Commodore users or clubs in your area, with who you may share ideas.
- Possible mention of your site in national advertising.

Presently, we would like any institution interested in becoming an EDUCATION RESOURCE CENTER to complete the form provided in this magazine and return it to Commodore. We will evaluate the eligibility of your site for membership. Your willingness to share this valuable information is greatly appreciated.

APPLICATION FORM

Educational Resource Center

Your Name: ______________________________

Computer Site Name: ______________________________

Address: ______________________________

____

Telephone #: ______________________________

Contact Person #1 ______________________________

Contact Person #2 ______________________________

Number of Commodore Computers at this location

Vic-20 ____________________

4016 ____________________

4032 ____________________

8032 ____________________

SuperPET ____________________

Number of non-Commodore Microcomputers at this location:

and Type ______________________________

____

Are you willing to be listed in Commodore Publications as an Educational Resource Center?

Yes__________, No__________

Do you have the staff and facilities to evaluate educational software?

Yes__________, No__________

Fold Here. Staple or Tape Bottom.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 4 NORRISTOWN, PA 19401

POSTAGE WILL BE PAID BY ADDRESSEE

Commodore Business Machines, In.
300 VALLEY FORGE SQUARE
681 MOORE ROAD
KING OF PRUSSIA, PA. 19406

Attn: Dr. Bruce Downing
Director - Industry Marketing

Bibliography—Microcomputers in the Classroom

Special thanks to Dr. Richard Brickley of Research and Information Services for Education (RISE), King of Prussia, PA, for this comprehensive bibliography on Microcomputers in the Classroom. Topics found in the bibliography include: applications, software or courseware, hardware, research, staff training, and some existing programs. The current status of each item is also listed.

Please Note: *This is simply a list of available resources. Neither Dr. Brickley nor his staff are distributors of this information.*

Explanation of symbols

AB — abstract of article
CA — complete article
EX — excerpt from article
mf — microfiche

Overview

1. CA **Ayers, George E.**
1980 The learner and the computer. Paper presented at the National Conference on Computer Based Education (Bloomington, MN; October 23-25, 1980) 9p. [ED 201 305].

2. CA **Eisele, James E.**
1981 Computers in the schools: now that we have them . . .? Educational Technology 21:10, 24-27 (October).

3. CA **Gleason, Gerald T.**
1981 Microcomputers in education: the state of the art. Educational Technology 21:3, 7-18 (March) [EJ 244 284].

4. EX **Oldham, Alison D., Ed.**
1981 Using computers to improve instruction. Croft Leadership Action Folio #148. Waterford, CT; Croft Educational Services. 31p. (section A, 1-9).

Educational Applications

5. CA **Alexander, Wilma J.**
1981 Microcomputers: impact on society and education. Business Education Forum 35:8, 19-21 (May) [EJ 245 276].

6. CA **Bell, Kathleen**
1980 The computer and the English classroom. English journal 69:9, 88-90 (December) [EJ 238 448].

7. CA **Caldwell, Robert M.**
1980 Guidelines for developing basic skills instructional materials for use with microcomputer technology. Educational Technology 20-10, 7-12 (October) [EJ 234 638].

8. CA **Derringer, Dorothy**
1981 Building better math education. Classroom Computer News 2:2, 22-23 (November/December).

9. AB mf **Hallworth, H. J. and Ann Brebner**
1980 Computer assisted instruction in schools: achievements, present developments, and projections for the future. Edmonton, Alberta; Calgary University Faculty of Education. 243p. [ED 200 187] - (3 cards).

10. EX **Oldham, Alison D., Ed.**
1981 An exemplary CMI (Computer-managed instruction) system. Croft leadership Action Folio #148. Waterford, CT; Croft Educational Services. 31p. (section A, p. 10-17).

11. EX **Oldham, Alison D., Ed.**
1981 Learning about computers. Croft Leadership Action Folio #148. Waterford, CT; Croft Educational Services. 31p. (section B).

12. CA **Rogers, Jean B.**
1981 An introduction to computers and computing: a high school course outline. The Computing Teacher 8:7, 30-33.

13. CA **Saltinski, Ronald**
1981 Microcomputers in social studies: an innovative technology for instruction. Educational Technology 21:1, 29-32 (January).

14. CA **Scandura, Joseph M.**
1981 Microcomputer systems for authoring, diagnosis and instruction in rule-based subject matter. Educational Technology 21:1, 13-19 (January) [EJ 240 912].

Software and Hardware

15. CA **Carter, Jim A., Jr.**
1981 How to buy microcomputers—and how and where to use them. School Shop 40:8, 28-33 (April) [EJ 241 922].

16. EX **Douglas, Shirley and Gary Neights**
1980 A guide to microcomputers. Harrisburg, PA; Pennsylvania Department of Education and Colonial Northampton Intermediate Unit. 42p. (title page, table of contents, 2-11).

EDUCATION

17. CA **Douglas, Shirley and Gary Neights**
1981 Instructional software selection. Harrisburg, PA; Pennsylvania Department of Education and Colonial Northampton Intermediate Unit. 22p. [R.I.S.E. Document 03389]

18. AB mf **Dresden Associates**
1980 School microware: a directory of educational software. Over 500 programs/packages for APPLE, PET, TRS-80. Dresden, ME; Dresden Associates. 52p. [ED 196 431] - (1 card).

19. CA **Editor**
1981 A level-headed guide to software evaluation. Classroom Computer News 1:6, 22-23 (July-August).

20. EX **Freiberger, Stephen J. and Paul Chew**
1980 A consumer's guide to personal computing and microcomputers. Rochelle Park, NJ; Hayden Book Co. 198p. (title page, table of contents, 44-53) [Montgomery County Professional Collection].

21. CA **Holznagel, Donald C.**
1981 The microSIFT model. Classroom Computer News 1:6, 26 (July-August).

22. CA **Milner, Stuart D.**
1980 How to make the right decisions about microcomputers. Instructional Innovator 25:6, 12-19 (September) [EJ 232 615].

23. CA **Robbjer, M. D.**
1981 When is it "good" courseware? Standards for microcomputer courseware. Educational Technology 21:10, 47-54 (October).

Research

24. AB mf **Cox, Dorothy Anna Howard**
1980 Early adolescent use of selected problem-soving skills using microcomputers. Ph.D. Dissertation; The University of Michigan. 264p. [ED 200 449] - (3 cards).

25. AB mf **Loop, Liza and Paul Christensen**
1980 Exploring the microcomputer learning environment. Independent research and development project reports. Report #5. San Francisco, CA; Far West Lab for Educational Research and Development. 90p. [ED 201 307] - (1 card).

26. CA **Spencer, Mima and Linda Baskin**
1981 Classroom computers: do they make a difference? Computer Classroom News 2:2, 12-15 (November/December).

Staffing/Training/Organizational Issues

27. EX **Baker, Justine**
1976 Computers in the curriculum. Fastback #82. Bloomington, IN; Phi Delta Kappa Educational Foundation. 41p. (title page, table of contents, 8-39).

28. CA **Boyd, Richard**
1981 Attacking the in-service education problem. The Computing Teacher 8:7, 50-51.

29. CA **Michael, Frederick W., Jr.**
1980 Funding sources for microcomputers. Instructional Innovator 25:6, 26-28 (September).

30. CA **Poirot, Jim et al**
1981 Teacher education. In R&D Speaks: Microcomputers and Education. Austin, TX.; Southwest Education Development Laboratory. 150p. (53-60).

31. AB **United States Office of Education, Comp.**
1981 Computeronics: gifted child project. In Educational programs that work. Vol. VIII. San Francisco, CA; Far West Lab for Educational Research and Development. 260p. (section 12, p. 14).

32. AB **United States Office of Education, Comp.**
1981 Computer-assisted-diagnostic-prescriptive program in reading and mathematics. In Educational programs that work. Volume VIII. San Francisco, CA; Far West Lab for Educational Research and Development. 260p. (section 9, p. 10).

33. AB **Upper St. Clair School District**
1977 Computer related learning. Upper St. Clair, PA; the District. 2p.■

Terminating PET/CBM BASIC Programs

by Dr. Hubert L. Youmans
Department of Chemistry
Western Carolina University

PET/CBM BASIC programs should be terminated with an END statement, if the programs are to be stored on cassettes that have been repeatedly overwritten. Omission of END statements will eventually lead to garbage being loaded with saved programs. A program will LOAD and LIST normally, then fragments of a second copy of the program will suddenly, in an almost random fashion, overlay the first. I have never seen the problem occur with a new cassette.

Overlaying undoubtedly reflects the quality of tapes used and the number of times they have been overwritten. But inexpensive tapes and END statements should represent good economics to teachers who go around hallways picking up pennies discarded by taxpayers' children.

Our students in BASIC are supplied with only one cassette per student and each student repeatedly saves programs on the same segment of tape. Therein is the genesis of a problem.

Last spring a student in the computer science BASIC course told me that her teacher took cassettes to the language laboratory to erase them because the PETs erased inadequately. That seemed strange to me, for I had lent the computer science laboratory the PETs they were using and I had never encountered the problem.

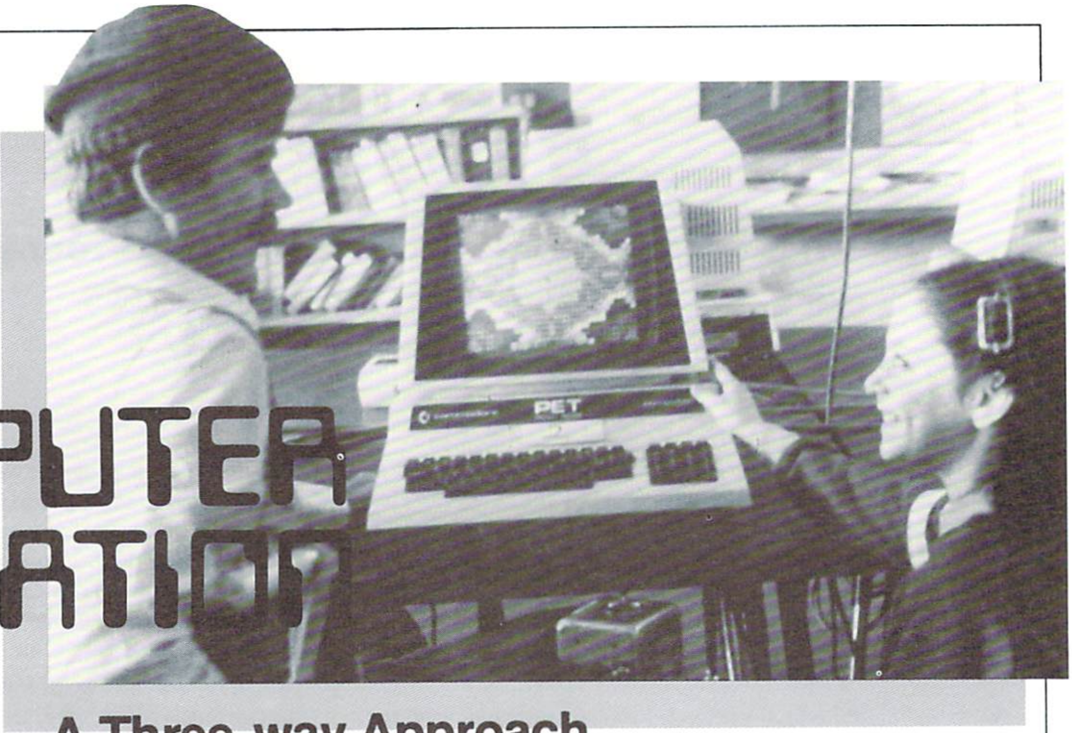
This past summer, however, loading started going haywire for some of my National Science Foundation Student Science Training Program (SSTP) participants. They were omitting END statements, as suggested by the literature. Including the statements solved what appeared to be a hardware problem. Shortly therefore the computer science department head told me that PET recorders were a major source of problems for them. They were not using END statements and their problems seemed similar to those of the SSTP students.

Such documents as Commodore Business Machines' "PET User Manual" and Osborne and Donahue's "PET/CBM Personal Computer Guide" say that END statements are optional for PET/CBM BASIC. That is true, if the programs are not to be stored or if they are to be stored on clean tapes. But teachers often must use the same tape many times and that can result in a garbage problem.

One solution to the problem could be to erase used tapes before recording on them, as the computer science professor did. This is a very inefficient solution, compared with a terminal statement such as 63999 END. Another solution that sometimes works is to clean the recorder magnetic heads. Of course, the heads should be kept clean, but cleaning the heads was not a satisfactory solution to our garbage problem. Head alignment also failed as a solution to the problem for the computer science department.

To terminate a program with an END statement is good programming, and a useful habit when the programmer goes on to other languages, such as FORTRAN and assembly. All my students now terminate their programs with END statements. And no student has informed me recently that my stupid old PETs were chewing up his programs and spewing out garbage.

Demands for me to troubleshoot instruments have fallen precipitously. The cure seems to be permanent or at least long-lasting; I have a music library program that has been overwritten on the same tape dozens of times during the past two years. It still loads properly. An END statement terminates the program. The cassette is the same brand my students used last summer.■



Dr. Chesire and a student enjoy a Commodore graphics routine.

COMPUTER EDUCATION

A Three-way Approach

Dr. Fred Chesire of the Royal Palm School in Phoenix, Ariz., related the story of the triad approach his school is using to implement their 22 PET computers. This three-fold plan includes:

1. Developing computer literacy with teachers.
2. Promoting development of problem-solving skills for gifted students in the school's Resource Center.
3. Finding/developing software for academic classes.

The following account by Dr. Chesire provides some thoughtful insights into education with microcomputers.

Computer Literacy With Teachers

Arizona State Univerisity offered an NSF institute to metropolitan Phoenix teachers, providing them skills for training their colleagues in computer literacy. Some 25 teachers represented 12 school districts. Another teacher and I represented Washington Elementary District. To hold up the flame of computer education we offered our first teacher computer literacy class and had twice the number of applicants that we could handle. The course was similar to others across the country: micro comparisons, software and hardware sources and evaluation techniques, introductory BASIC, software manipulation, relevance and extensions of microprocessors, computer history, and current trends. The texts we used were Adam Osborne's *Running Wild*, Christine Diorr's *Microcomputers and the 3 R's*, and Carol Alexander's *Feed Me, I'm Your PET Computer*.

We finished that class in December. The class celebrated with a cake decorated with computer statements and commands.

We have started instructing our second batch of teachers. Our approach has changed. More and more we are following a hands-on approach. The teachers do simple programs to learn the concepts of programming and the scope of higher level language. They manipulate these programs to become familiar with the mistakes that their students will make. In lieu of a lecture on computer history, we have them run a tutorial on computer history or give them a reading followed by a multiple choice computer quiz. We like to drill teachers on necessary terms and definitions using computer multiple choice questions that require the correct answer before continuing. We push program manipulation so teachers will feel that they are in charge.

One evening's highlight is to open a PET, identify the major parts with the assistance of a simple diagram, and remove a snack that has been planted inside the PET. We want them to have only good things associated with their PETs.

We still provide teachers with stacks of materials on resources and readings. Using material as relevant to their individual curriculum as possible, we drill teachers on all the main cassette and disk drive commands. Our belief is that, after this eight-hour introductory computer literacy course, they will have lost initial fears and have some idea of the direction they want to go. Some of them will remain users (with some idea of how to evaluate software), while others will want to take a programming class. The latter will now have some idea of the commitment required. We feel that these classes are broadening our numbers of informed teachers who will be communicating with each other and who will be enticing new teachers into the computer literacy realm.

Problem solving with Students

This year Royal Palm is a fourth through eighth grade school. I begin my fourth graders on challenging activities of educational value. They are currently working on graphics manipulation using screen locations and peeks and pokes to make pennants wave on castles. The fifth graders begin introductory programming with emphasis on development of debugging skills. They are presented with software that enables them to find the answers to problems with correct usage of the program. The sixth graders develop more sophistication in their programming and become involved with simulations. The resource center students use software which helps to develop logic and problem-solving skills. The students learn programming skills and apply what they have learned to developing individual projects. The seventh and eighth graders are encouraged to develop their own projects in music, graphics, CAI,, sports statistics, and topics relevant to their special interests.

Computers in the Classroom

We are very fortunate to have a 22-PET computer center. The teachers of all curriculum areas bring their students to use software from the San Mateo Softswap and the Canadian Education Advisory Board. Some basic drill and practice programs have been modified for social studies and a French mini-course. I am delighted to say that teachers are beginning to write their own material and to modify available material to meet their own classes' needs. The software is developed and modified by teachers in the science, language arts, math, and social studies curriculum. Many of our students look with envy as teachers take home PETs over the weekend to plan lessons.

Where Are We Going?

This next year our school will have only seventh and eighth grade students. We will be implementing a five to six week computer literacy course for all of our 1000 junior high students. The course will be built into either the math or science curriculum. I plan to offer an intermediate level BASIC class with projects centered around assisting the classroom teachers. We will have a typing class and, of course, a natural follow-up will be word processing. Who knows what else! Maybe we will tie into an electronic mail company and provide research capabilities.

We at Royal Palm have miles to go. There will be workshops and conferences and authoring languages and development of a stronger state CUE. Do you know what this means? This means improved education . . . better-trained teachers, better-prepared students, and a better world!

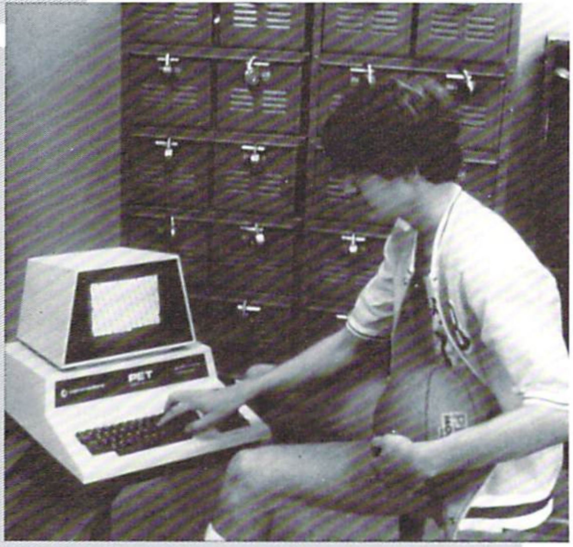
According to Chesire, this large commitment to Commodore computers is not just restricted to the Royal Palm School. "I believe our Phoenix area is a potential hotbed for Commodore," said Chesire. "The local Madison School District is putting in PETs, Scottsdale Schools are using PETs, and Mesa is putting numerous SuperPETs in their high schools. Nearby Chandler, Paradise Valley, and Alhambra districts also have PETs," he said.

As far as Royal Palm School is concerned, Chesire admits they still have "miles to go." Future plans for the school include workshops, conferences, authoring of languages, and development of a stronger state CUE. What does all this mean? For Chesire, it spells "better trained teachers, better prepared students, and a better world!" ➡

EDUCATION

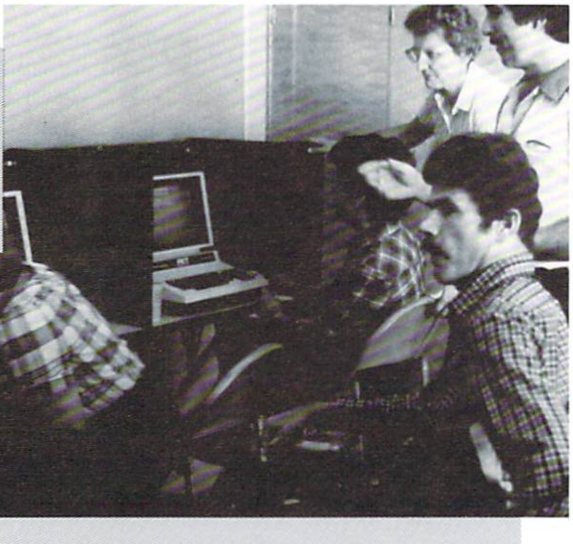
At left, a computer music program, written and submitted by this student to the Arizona Programming Contest, won him first prize—a VIC microcomputer.

Below, a junior high basketball student learns to handle data and files by keeping basketball statistics.

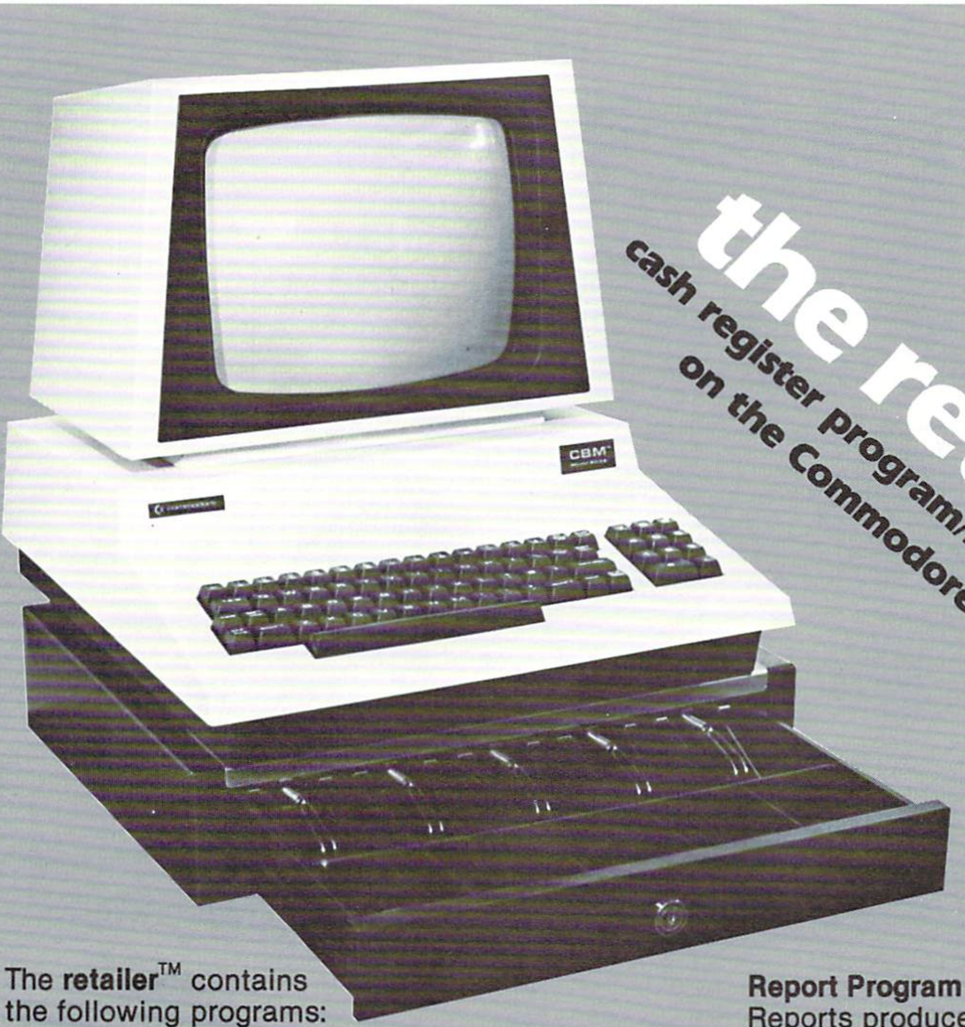


At right, the National Anthem rings over the school's P.A. system. The student wrote this program to use CB2 sound capabilities.

The school's computer center, left, hosts 22 PETs in a stimulating learning environment for students in all academic areas.



Dr. Chesire, right, teaches an after school class for teachers desiring to improve their computer literacy skills.



the retailer™

cash register program/inventory management
on the Commodore CBM

The **retailer**™ contains the following programs:

Cash Register
- Records sales transactions
- Prints register tape or invoice
- Automatically computes discounts and/or sales tax
- Computes change due
- Prints daily sales summaries

Inventory Manager
- Creates and maintains complete information on every item in inventory (over 3000 items)
- Adds, changes, deletes inventory on command
- Automatically updates inventory from register transactions
- Maintains stock balance, records dollar amounts sold as well as cost of items sold
- Prints price tags

Report Program
Reports produced:
- Listing of:
  - All Inventory items
  - Items sold in past week
  - Items on order
  - Items on hand by season
  - Items with zero stock balance
- Dollar listing by stock category of:
  - Weekly sales
  - Cost of weekly sales
  - Merchandise on hand
  - Merchandise on order
  - Merchandise at mark down
  - % sales to total sales
  - % stock to total stock
  - % gross profit
  - % stock at mark down
- Daily summary of sales by month

available now
$650.00

Dealer inquiries invited

RETCOM Systems, Inc., 61-B Mountain Blvd., Warren, New Jersey 07060 (201) 561-3112

Microcomputing in the Troy, Michigan School District

In an era when many public schools are conserving budgets and cutting back on student services, the schools in Troy, Michigan purchased 76 Commodore microcomputers, 26 disk drives and 15 printers. The story of their efficient district and the enthusiasm with which they have integrated the microcomputer into diverse curricula at many levels is a case study in "how to do it right."

The mere, purchase of microcomputer hardware does not guarantee that teachers or students will use them, nor does it guarantee that teachers will find legitimate or creative ways to integrate the microcomputers into the curriculum. The example provided by the Troy, Michigan school administrators is a welcome illustration of planning, training and curriculum integration. Their approaches to low-cost workshops, communication techniques, local friendship groups or clubs and their funding strategies could be applied quite universally by other school districts.

Troy, Michigan is located north of Detroit in southeastern Michigan. The Troy School District has twelve elementary schools, four middle schools and two high schools, serving more than 11,000 students.

Hardware Resources

The high schools in Troy provide 12 Commodore business computers (CBM 8032 models) to the business students and the data processing students.

In the twelve elementary schools, 59 Commodore PET computers are installed throughout the district. Nearly half of these computers were purchased through funds raised by the parent-teachers organization. Twenty-seven disk drives (2031 models) were added to the PETs in the second year of operation so that students would not lose time while waiting for cassette tapes to load programs into the computer.

Now entering their third year using microcomputers, the district has followed a plan of enlarging their equipment base each year in parallel with the growth of teacher awareness and skill. The search for microcomputer hardware followed a simple checklist: what hardware features were available for the least dollar expenditure.

Commodore's long-standing 3 for 2 pricing structure (see Commodore News section) awards equipment grants to qualified schools. The ability to purchase two microcomputers and receive a third machine as an equipment grant was a primary factor in the purchase of the Troy microcomputers.

The School Board, as well as the teachers and the administrators, must be given information. Stephen Shotwell, organizer of the training plan, said, "*the teacher requires hands-on skills and implementation techniques, and the administrator needs to know the benefits, implementation methods and management techniques for the computer education program. The school board member,* *on the other hand, requires concrete rationale for such a program and a periodic update on the status of the program.''*

Shotwell and others developed workshop curricula for all three of these groups, the content aiming to satisfy the unique needs of each group. Early training included organized visits to other schools and surveys of experiences from other schools. Workshops are often repeated once for elementary teachers, once for middle school teachers, and again for high school teachers.

Not until the third inservice workshop was any instruction given in the writing of educational software. Local materials were adapted from several sources to give clear directions on how to connect printers, computers and disks, or how to initialize new disks, etc. Eventually a concise manual for teachers was compiled titled ''Teaching Your PET to Speak Essential BASIC.'' One clever outline for a workshop is shown below.

Getting to Know Your PET

I. Taking your PET out of the box.
II. Did your PET come with papers?
III. Taking your PET on trips.
IV. Hooking up your PET's Leash (cassette player)
V. Grooming your PET.
VI. The PET Vet.
VII. Turning your PET on!
VIII. Bytes! (That smarts!!!)
IX. Your PET gets loaded.
X. Running your PET.
XI. Stopping your PET from running.
XII. Follow directions (obedience training?)
XIII. Canned PET food.
XIV. Your PET's reproductive system.
XV. Programming your PET.
XVI. Reading your PET's mind (list).

The quality and extent of the inservice training can best be seen in the group journal called ''The Troy Computer News.'' It is available at mailing cost from the TROY MICRO COMMITTEE, 3601 Forge Barnard Elementary School, Troy, Michigan 48084.

Software Resources

Each school in the Troy system has about 300 software programs available to support various curricula at all grade levels. The selection and acquisition of software followed a planned cycle of growth and discrimination.

Initially, in 1980, a small set of computer programs was supplied to each school enabling the students to experience the variety of graphics and drill which the computer can provide.

Meanwhile, a software committee of teachers researched the software ket, isolating those programs which they would recommend for purchase. The district awarded about $500 to each school for the purchase of software which supports specific curricula and lesson objectives. In addition, the Parent Teachers Organization funded the purchase of additional special-purpose software to satisfy the special needs of certain schools.

With the installation of disk drives in 1981, the resources of the intermediate and County levels of organization made additional programs available to all of the schools.

Meanwhile, the teachers are beginning to create specialized software, tailored to the needs of the students and the community.

In grades kindergarten to five, special assistance is sometimes needed to guide the children in their early experiences with computers. To meet this need, volunteer parents are invited to attend school for a few hours to assist the children and the teachers.

This raised the question of how parents could be trained to use the equipment. Two workshops were organized for parent volunteers. Thirty-eight parents responded and most of them continued through a 6-week course of study. Now, some children don't have to wave good-bye to their parents in the morning. Instead, the parents go to school with their kids.

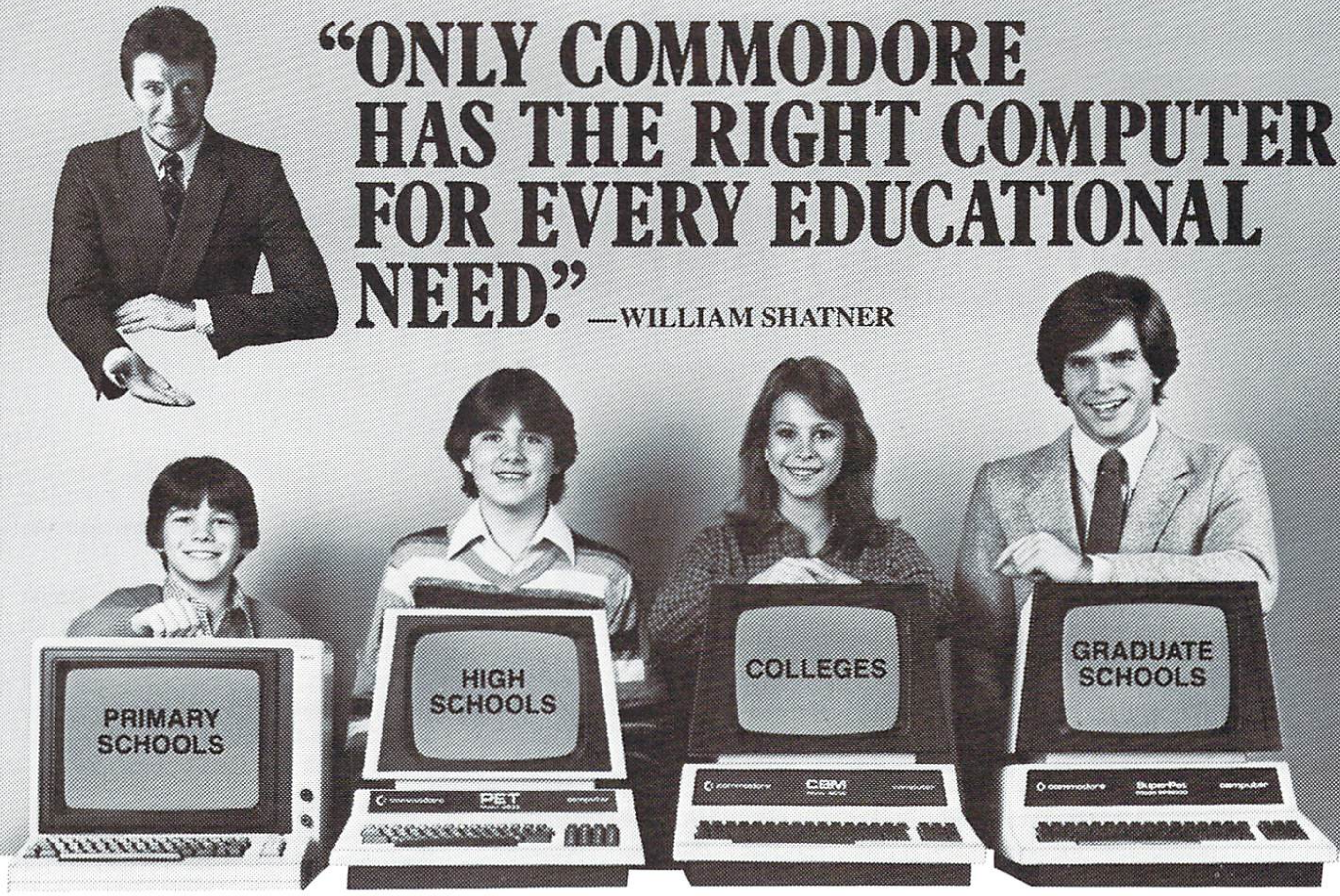
Quality Inservice Training Plan

Administrators in the Troy district believe that the key to establishing and maintaining a microcomputer education program is to provide continuing quality inservice training. Such training, when it involves hundreds of individuals, cannot be provided without thoughtful planning.

The Troy administrators were determined to avoid the pit-falls which often occur in some districts. Without quality training, users may not be able to find the on-off switches. Others will wonder why the machines were pur chased and complain about the costs.

Computers eventually gather dust unless teachers are given the opportunity to learn the various services which microcomputing can provide. Such training does not occur in one sitting. It requires continuing in-service classes and a curriculum for the teachers which provides software examples, printed instructions, helpful aids and open communication channels.■

"ONLY COMMODORE HAS THE RIGHT COMPUTER FOR EVERY EDUCATIONAL NEED." —WILLIAM SHATNER



VIC-20

- So simple, a child can be running programs in minutes.
- Ideal for improving basic reading and arithmetical skills.
- Large easy-to-read characters.
- 16 screen color/4-sound generators.
- Real typewriter-style keyboard.
- Memory expandable to 32K.
- Easy-to-follow instruction guide and programming course.

$299.95

PET® 4000 Series

- World's most widely used classroom computer.
- Our lowest priced, completely self-contained computers combining keyboard, display and processor in one unit.
- Memory expandable to a full 64K.
- Built-in BASIC language.
- 128 graphic character set.
- Crisp, easy-to-read 12-inch green phosphor display.

$995

CBM™ 8032

- Students can use true business software, such as word processing, VisiCalc[1], the electronic worksheet and hundreds more.
- Professional 80-character phosphor screen.
- Easy-to-read 12" green phosphor screen.
- Print in Basic/full screen editing capabilities.
- Business-oriented numeric keypad.
- Full 32K memory expandable to 96K.
- Built-in BASIC.

[1] VisiCalc is a Trademark of Personal Software Inc.

$1495

SuperPET Computer

- Compatible with all CBM 8032 Software.
- Offers four (4) high-level languages developed by Waterloo specifically for teaching: including BASIC, APL, Pascal and FORTRAN.
- Waterloo: MicroEDITOR, 6809 Assembler & Linker programs.
- Provides 134K of memory and dual microprocessors.
- Built-in sophisticated hardware features IEEE-488 and RS-232 interfaces.
- Applications can be uploaded directly to Mainframe.
- 80 characters per line display.

$1995

Commodore's commitment can best be measured by our comprehensive range of computers, accessories, software and grant support to education.

Whatever your grade level or computer requirements there is a Commodore that will give your school cost-effective computing. From the incredible VIC-20, the only full-feature computer for under $300 to the sophisticated new SuperPET computer that offers the capabilities of a large mainframe system, at the low cost of a desktop computer.

And all Commodore computers offer the widest choice of intelligent peripherals, including printers, disk drives, cassette recorders, telephone modems and more which are compatible throughout the full range.

We wrote the book on software. Commodore's Commitment to Software... Your Commodore dealer has our new 380 page Education Resources Manual which includes four disks of 185 public domain software programs. Also, the latest edition of our Software Encyclopedia details over 1000 applications including specialized education software Math, Science, Social Studies, Verbal Skills and Administration.

Commodore's Commitment to Service... Our 600 local dealers support you and maintain your equipment. In addition a nationwide service contract is available through TRW's 200 locations for fast, reliable maintenance of your systems.

Commodore's $4.5 million Grants to Education... Commodore's Education Advisory Board granted a computer free for every two purchased on certain models. Last year this amounted to over $4.5 million.

Commodore Computer Systems
681 Moore Road, King of Prussia, PA 19406

Canadian Residents: Commodore Computer Systems
3370 Pharmacy Ave., Agincourt, Ont.,
Canada, M1W 2K4

□ Please send me more information about Commodore computers for Education.

□ I would like to subscribe to the Commodore Magazine. Enclosed is a check for $15.00.

□ I would like a copy of the latest Software Encyclopedia. Enclosed is a check for $9.95.

Name__________ Phone__________

Grade level ______ Course __________

School name__________

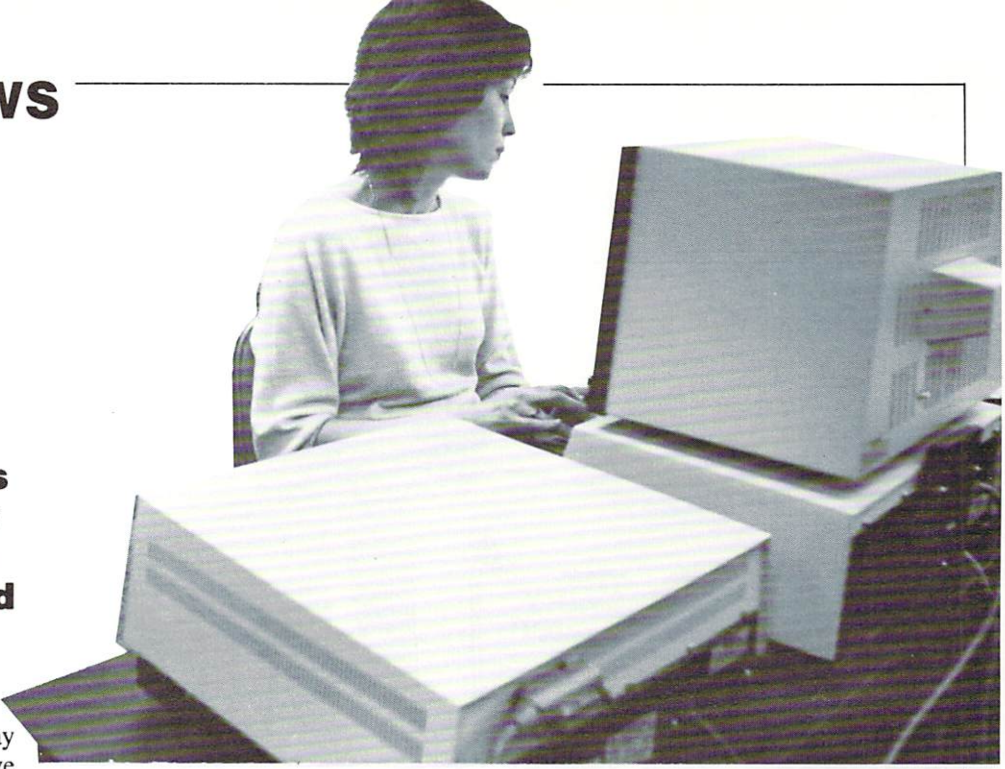
School address__________

City__________ State ______ Zip ______

commodore
COMPUTER

SENSATIONAL SCHOOL OFFER:

Buy 2 PET (16K or 32K) or 2 CBM computers.
Get 1 absolutely free! Offer good for a limited time only.
SEE YOUR COMMODORE DEALER



The hardware configuration used by the law firm consists of two Commodore CBM-8032 computers, two CBM-8050 floppy disk drives, and a parallel interface sharing a Spinwriter letter quality printer. This configuration allows each work station to print out to the same printer.

A Case Study

Law Firm Improves Office Productivity Using Commodore Microcomputer and Info-Designs Software

Attorneys throughout the country today are looking for more cost-effective ways to improve their office productivity and to speed up collections from their clients. Some attorneys have even spent $30,000 to $50,000 on a minicomputer system to help control time, billing, and receivables.

But, many attorneys are now discovering the tremendous benefits that a Commodore Business System can offer their practice. The law firm of Bernard P. Paige, PC, in southeastern Michigan is one example of how a Commodore computer system is providing solutions to office automation needs.

The hardware configuration used by the law firm consists of two Commodore 8032s, two CBM 8050 floppy disk drives, and a parallel interface sharing a Spinwriter letter quality printer. This configuration allows each work station to print out to the same printer. The firm also uses the Time Management/Client Billing System, a Commodore-approved software package developed by Info-Designs, Inc., of Birmingham, Mich.

''We have had two Commodore CBM(TM) Systems and the Time Management/Client Billing software in place for six months now,'' explained Bernard Paige, ''and it has helped to streamline our operation considerably.''

''In this economic climate, every hour of legal services must be billed, and the Time Management/Client Billing System gives us an efficient method of tracking attorney time, maintaining accurate records by client and matter, and providing flexible billing for any matter at any time,'' he said.

''This system has been completely reliable. I've heard of all the 'horror stories' that some of my colleagues had in getting their first computer system in place and running properly. And, certainly, getting the right software is the key! We must have great hardware and software, because in over six months of constant use, we have never been down, never lost any files, and had very few problems in operating the system.

The Time Management/Client Billing System, developed by Info-Designs, Inc. of Birmingham, Michigan, provides a simple ''fill-in-the-blanks'' screen format for entering client billing and other information.

''Probably the best feature of the system is the management reporting,'' continued Paige. ''The reports provide me with invaluable information to evaluate our business by type of practice and by attorney utilization. I can spot potential problem areas by reviewing the time value of our services versus the amount of fee we bill.''

On the same computer as Time Management, Paige's firm also has a word processing system (WordPro(TM)) which provides his firm with the same features and capabilities for much less money than the IBM Office Systems 6 word processor that he replaced. Other applications include: Accounts Payable, Payroll and General Ledger which automate much of the administration of the legal practice. Obviously satisfied with his choice Paige said he would recommend the Commodore system with Info-Designs Time Management/Client Billing to any attorney who is looking for a cost/effective solution to his office productivity bottleneck.

WordPro is a registered trademark of Professional software, Inc.

For more information on Professional Practice software solutions, contact your local Commodore dealer or write to Info-Designs, Inc., 6905 Telegraph Road, Birmingham, Michigan 48010.■

Commodore Computer Helps Blooming Business Grow

by Bob West

How would you like to have an employee who works fast and accurately 24 hours a day, seven days a week, including holidays, doesn't take a vacation, is seldom sick, doesn't complain, and follows instructions to the letter?

Salary? About $10,000 the first year, no unemployment insurance or Social Security payments and no pay the following years.

Unbelievable? No, there is such an employee! In fact, there is one or more available for each nursery that has an opening.

Wells Nursery Inc., Penrose, NC, hired one. And I taught it to do its job. It's called a nursery computer system.

Designing the System

A computer can do many different jobs if given the proper instructions. Each step of every function must be described in minute detail in precise order. And all of this must be written in a language the computer undrstands.

My first step was to learn the manual system used at Wells Nursery. This was accomplished through interviews with Jeremy Wells, president of the nursery, and his staff. Some of my questions were: What takes place when an order comes in? What kind of records are kept? What kind of information is in those records? How many records are in each file? What reports are needed? Are there any reports that would be useful if you could get them?

I organized all of this information into work flow charts and looked for ways to let a CBM 8032 do most of the work with a minimum of effort from the operator. However, there were some things that we decided to leave to human judgment, even though the computer logically could have been programmed to do them. One such decision was substituting for unavailable plants on orders.

Operator Involvement

Another consideration in the design of the system was operator involvement. I wanted to make the 8032 easy to work with. This could be accomplished in part by messages and prompts on the video screen. The computer would let the operator know what to do next, or what it is.doing, at all times. When possible, it would catch operator errors, tell what was wrong, and provide the opportunity to correct the information.

It could caution the operator when it was instructed to do something the operator might want to think twice about. I wanted to make the computer so easy to operate that with a few minutes of instruction, anyone on the nursery staff could use it.

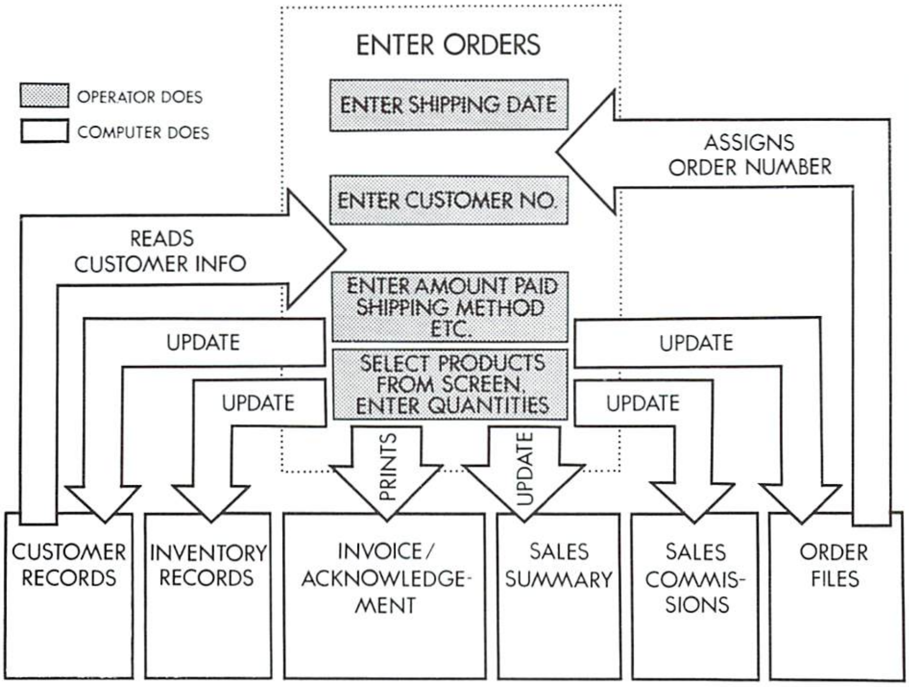
One of the most helpful features of the system would give the operator the ability to select any inventory item from a list on the video screen. This could be done without having to know the product number.

After I reviewed my initial notes, the sample reports and the list of requirements, and developed flow charts, I had some more questions. At this time I reviewed the system with Wells so that he could correct any misunderstandings or misdirection in my efforts.

Writing the Programs

A month had passed since I began

Bob West uses the step-by-step owner's manual in a training session with secretary Sheila Hunter. Because the computer is an automated version of the familiar manual system, the Wells Nursery staff found it easy to learn.



This flow chart is for one of the computer processes, "enter orders." It shows how many steps are automatically performed by the Wells Nursery computer system.

working on the nursery project. Now I was ready to share my knowledge with the 8032 and begin writing "job descriptions"—a list of instructions for the computer to carry out.

I wrote a program for each of the jobs I wanted the computer to do, like processing orders, changing orders, maintaining customer records, recording inventory, processing end-of-month and end-of-year materials and printing out various reports.

Most of these programs turned out to be rather complex, for there were all sorts of error-checking routines, verifying, cross-referencing and so on. For example, the "order program" not only makes it easy to enter an order, but it automatically updates the customer record, the appropriate inventory records, the sales summary and the sales commissions. This program also records the order on a disk in the appropriate season file and prints out the acknowledgement or invoice.

These programs use a lot of memory. One program at a time was about all a microcomputer could handle, because each program carried so much information. But this posed no problem. All of the programs could be stored on magnetic floppy disks and called into the computer one at a time as they were needed.

Menus and Debugging

Just as menus in restaurants are used to give guests a choice of foods, menus in the computer system allow the operator to select from a list of jobs. Each job has a number. When the number is entered, the computer calls in the program to perform the job. When the job is completed, the computer calls in the menu again so that another selection can be made.

Over the next two months, I wrote 21 programs and set up 18 files on the two disks that make up the software of the system. Dummy data for the customer and inventory records were used to test the programs. Orders were entered; some were changed or canceled. Reports were printed. I tried to use every possible type of operator response with each of the programs to see how the computer would react.

During this time there were two other conferences with Wells. The computer was allowed to sit in on these. When it seemed like the system was bug-free, it was time to put it to work at the nursery.

Putting It On-Line

As soon as the equipment was installed, I began training the Wells staff to work with the computer. This was easy for two reasons. First, the 8032 is just an automated version of their familiar manual system. Second, the computer continually prompts the operator with messages on the screen.

Test disks with the dummy data were used to familiarize the staff with the system and the functions of the various programs and items on the menu. Then the changeover to Wells' system disks was made, and real data was entered.

While it appeared the system was error-free before installation, I knew that this was unlikely. It is virtually impossible to completely test a system under every possible condition in the programming laboratory. The real test is under actual day-to-day working conditions. So naturally a few minor problems showed up, but they were corrected promptly.

Evaluation of the Computer

How has this new "employee" worked out? The nursery computer has been on the job for almost four months. And its employer, Wells, reports that he feels it has already earned its "salary." Orders are processed in one-tenth of the time previously required. Weeks are saved on some jobs.

Current information on inventory is always available on the computer. Some information is now available for management decisions that was not possible without the manual system. Much of this increased efficiency, accuracy and savings was anticipated in the decision to automate. But other benefits are showing up.

According to Wells, the computer has increased the value of each staff member's time. For example, the secretary, freed from the time-consuming tasks of keeping track of details in piles of paper and notebooks, is able to use her time to keep in touch with customers by telephone. This use of time makes the job more interesting. Wells, who was spending much of his time making those calls, can now use his time in more profitable ways.

Now, if the secretary is out sick, the office work does not come to a halt. The Commodore microcomputer is on the job to assist those who need information or want to process an order. Even the company president can provide service after hours for a special customer and process an order with minimum effort if the need arises.

And that is not the end of the story. The nursery computer gets the inventory and billing chores done in short order. It is so efficient that it has time for other things, such as payroll, accounts receivable, general ledger accounting and word processing.

After the computer helps with all of those jobs, the staff might find some spare time for a few space games, with the nursery computer, of course.■

Robert A. West is president of Bob West Computers, Brevard, NC.

How to Buy a Nursery Computer

by Jeremy Wells

Jeremy Wells, left, and Bob West discuss ways to adapt the Wells Nursery manual system to a computer system.

It all started with a pile of paper.

We were in the midst of our 1980 spring shipping season. As usual, shipments were flowing out of the nursery, and the piles of adjusted inventory sheets, invoices, corrected acknowledgements and bills of lading were mounting so high that only the secretary's eyes showed over the top.

On April 1 a late request for plants came in. Because our inventory control was in books, we could not immediately get an availability listing. I started to sort through the piles of yellow and white paper, but after an hour, I realized it was hopeless. We lost the order because I could not give an accurate listing. It was then I decided that we had to come up with a better system.

Our Paper System

Inventory control is difficult in any size nursery. We had tried many different systems, and our paper system had become relatively efficient. We maintained 60 different varieties of rhododendrons and azaleas in 10 individual sizes for a total of 600 categories. These categories were color coded by size. For example, all 12- to 15-inch plants were listed on pale blue paper.

We had four large books that contained all information on available salable inventory. We also maintained an order receipt book showing all orders received by month and an accounts receivable book based on time of shipment. All six books were correlated and cross-referenced.

A small order took about 15 minutes to process; a large order could take up to two hours. Errors in addition or subtraction were common and double or triple checking was mandatory.

Up to half of our secretary's annual work load was spent on office inventory control. When this time was added to the hours spent on inventory control in the production and propagation area, a total of almost 1,500 working hours was spent on the process. This equals approximately nine months of work for one person.

I thought it was necessary to search for an electronic system that could speed up the process.

Streamlining the System

We revamped our field systems to allow one person to take inventory by count in late spring. In late summer I took these count sheets and updated the count, size and number of flower buds on the plants. We reduced our total field time to approximately two working weeks.

Our next step was to reduce office time.

"Mr. Wells, you don't know what you are talking about!"

As I was unfamiliar with computer systems, I planned my search for the right equipment as if I were purchasing any normal piece of nursery-related equipment. I set out to inspect all nationally known computer companies. Little did I realize how long the search would take.

Requirements for our Office

We wanted a simple, flexible system that would allow our current paper cross-referencing and provide accurate checks and balances. It should be a system that any individual within the company would be able to operate. We wanted safeguards for our corporate information.

In addition to inventory control, we wanted the system to handle customer information, daily sales analysis, monthly and annual sales analysis, sales commissions and customer mailing lists. We also wanted it to write orders and print a variety of reports. We wanted all printing to be of letter quality. The entire system had to be purchased or leased for less than $10,000.

Frustrating Search

It was a substantial set of requirements. Almost none of the companies con-

→

tacted were helpful. Some of their comments were:

"Mr. Wells, you don't know what you are talking about!"

"Your inventory system is too complicated!"

"We can't set up a program for nurseries!"

"We can program for nurseries, but it will take three years!"

"You will have to program your own system."

"The entire system will cost $30,000!"

"We will sell it to you for $15,000, but with no warranties or service!"

"Your company is too small for a computer!"

After a year of searching, it seemed that no company was interested in custom designing a nursery system and supplying hardware and service with it. If we were a supermarket, a system existed. Otherwise, we could forget it.

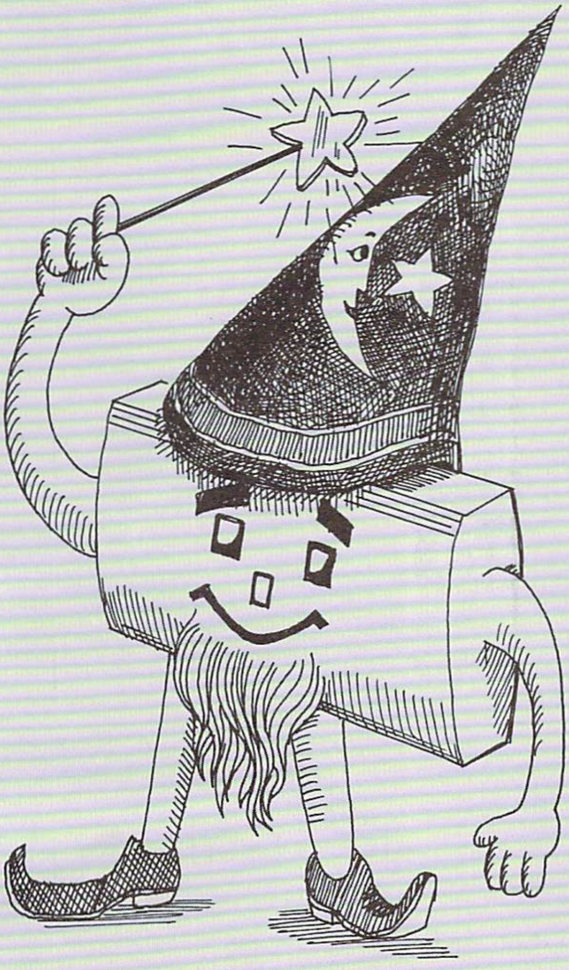
The Solution

it was March, 1981, when Robert West of Bob West Computers came to Wells Nursey in response to a telephone call. He represented Commodore microcomputers. After a year of computer salesmen, I watched this one enter our office with a jaundiced view. He listened to my remarks for a few minutes and then said, "Mr. Wells, how can I help you make money with my computer system?"

This comment stopped me. No other salesman had said this to me. I was beginning to see a glimmer of hope. We talked in detail for a couple of hours. Then West said, "Mr. Wells, I feel that I can design a system for you as you desire it. I am so convinced of this that I will return your deposit if I fail to give you a system that you, sir, can operate. I need three months to complete the task. May I have your order?"

Now, there was confidence in one's abilities! I was so impressed that we launched into the project. Many details were hammered out, and some changes were made in the original concepts for our program. Our system was on line by August 25, 1981—3½ months after signing the contract.

Jeremy Wells is president of Wells Nursery Inc., Penrose, NC.■



The VIC Magician

Your First Practical Program

by Michael S. Tomczyk

Creating "Practical" Programs

Some people can't cook (like the VIC Magician). But by following a recipe point by point, even the VIC Magician can come up with some tasty concoctions. Programming your VIC-20 is just like cooking — the ingredients in a program can be mixed and matched like a recipe. All you need is a menu of neat little programs you can put together in various ways to create sophisticated *practical programs for home, school or business. This modular approach helps you do complex computing with a minimum of experience . . . and allows you to take advantage of some sophisticated programming techniques, even if you're just a beginner.*

One Step at a Time

Writing a practical program is easy. Just take everything one step at a time. You'll find that most practical programs use the 6 steps shown below. There are many other approaches but this should give you a good start. Oh . . . one caution . . . don't get bogged down with "cosmetics" like graphics or screen placement. Your first task is to get the program WORKING in terms of instructions, calculations and results. The cosmetics and adjustments are the LAST elements to add to your program. To use our cooking analogy, broil your steak first, then add the seasoning. Here then are 6 elements involved in most practical programs:

1. DISPLAY (PRINT) QUESTIONS OR CATEGORIES ON THE SCREEN.
2. CONVERT THE USER'S TYPED-IN RESPONSES TO "INPUT VARIABLES." (In other words we translate what the user types in from the keyboard into short string or numeric variables the VIC can understand and work with.)
3. USE THE INPUT VARIABLES TO CALCULATE THE RESULTS YOU WANT TO SHOW.
4. DISPLAY (PRINT) THE RESULTS OF THE CALCULATIONS, ALONG WITH SOME DESCRIPTIVE WORDS.
5. REPEAT THE PROGRAM OR END THE PROGRAM.
6. ADD COSMETICS (to make the program "friendly").

Each program is different, of course, but these are the basic parts of most practical programs. The hard part is putting it all together. Most of us are intimidated by long programs with lots of fancy commands and mysterious variables . . . but those programs didn't start out to be that long or fancy. They started very simply and grew as the programmer kept adding elements to make the program more powerful or "friendly."

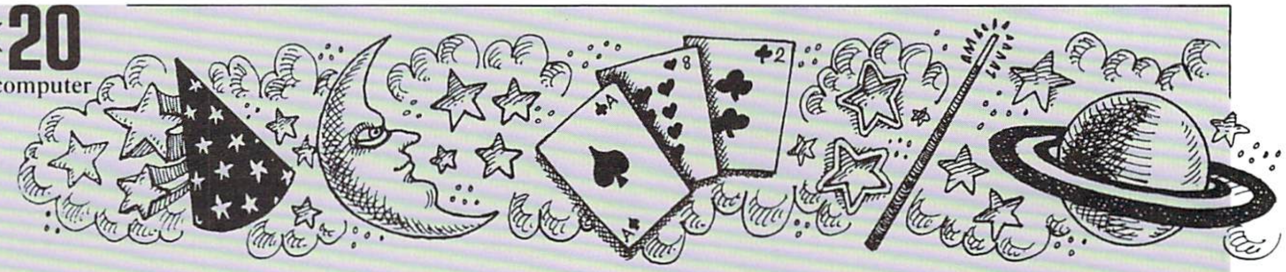
Plan Your Program Carefully

So you're ready to write your practical program. The first thing to do is make a CHECKLIST of what you want your program to accomplish. Try to write it in chronological order. If it's a long list, break the elements down into the simplest steps and build the program slowly, starting with the program's LOGIC. Here's an example of the checklist used for INCOME/EXPENSE program described in detail below:

1. NAME OF PROGRAM: INCOME/EXPENSE BUDGET
2. CLEAR SCREEN
3. ENTER TOTAL INCOME
4. ENTER EXPENSE CATEGORY (3 ITEMS)
5.

   ADD TOTAL EXPENSES (E1+E2+E3 = EXPENSES)
7. COMPUTE NET INCOME (INCOME − TOTAL EXPENSES = BALANCE)
8. FIGURE PERCENT OF TOTAL EXPENSES FOR EACH CATEGORY (E1/E = %)
9. EXPENSES = WHAT PERCENT OF INCOME ? (EXPENSES/INCOME = %)
10. INCLUDE ROUNDING FUNCTION TO ROUND ALL NUMBERS TO 2 DECIMAL PLACES (CENTS) (POSSIBLE DEF FN).

→



From here, you can either do a more detailed programming flowchart using standard flowcharting symbols and notation . . . or . . . you can simply make programming notes next to your original checklist, as shown above in parentheses.

Some Practical Programming Tips

Before we get into our program example, let's go through some fairly standard PRACTICAL PROGRAMMING TIPS, with some shorter examples to show you how various techniques work.

CLEAR the Screen

The first line in most practical programs is a CLEAR SCREEN command. Clearing the screen is covered in the VIC-20 user's guide, but here's a quick refresher. Normally, you CLEAR the screen by holding down the SHIFT key and hitting the CLR/HOME key. In your program, you will simply add an opening line number and PRINT the CLEAR command, like this:

10 PRINT" ♥ "(Type SHIFT CLR/HOME here . . . the reverse heart symbol appears whenever you CLEAR the screen inside quotation marks)

The CLEAR command doesn't have to be on a separate line. You can include it in your first PRINT statement, if you like, for example:

10 PRINT " ♥ MONTHLY INCOME"

Use PRINT Statement to Prompt (Instruct) User

A "prompt" can be a question, word, phrase, number, instruction or category. It simply helps the user type in the right information. Here's an example of a PRINT statement which "prompts" the user to type in his monthly income:

10 PRINT "MONTHLY INCOME:"

Use INPUT & VARIABLES to Accept User Information

An INPUT statement automatically places a question mark (?) on the next line after the prompt you PRINTed, and causes the VIC-20 to wait patiently until the user types in a response to the prompt. Here's an example of an INPUT statement:

What we did here was first PRINT the prompt message . . . then we told the VIC to wait for an INPUT (response) to be typed from the keyboard, and assigned the variable name M to the value representing monthly income. The PRINT message in line 30 simply proves that the VIC accepted this information.

INPUTs can and usually are on the SAME LINE as the PRINT statement. You can save memory space and increase your program efficiency by combining the PRINT and INPUT statements on one line. Be sure to separate both commands with a colon (:) as shown:

10 PRINT "MONTHLY INCOME":INPUT M

You can also combine several INPUTs on one line. For example:

10 PRINT "ENTER 3 NUMBERS":INPUTA:INPUTB:INPUTC
20PRINTABC

VICtip for First-Time Computer Owners —

TO ENTER A PROGRAM LINE INTO THE VIC-20 — just type the line number, the program commands or statements, and hit the RETURN key.

TO SEE THE PROGRAM LINE(S) OPERATE — type the word "RUN" and hit RETURN.

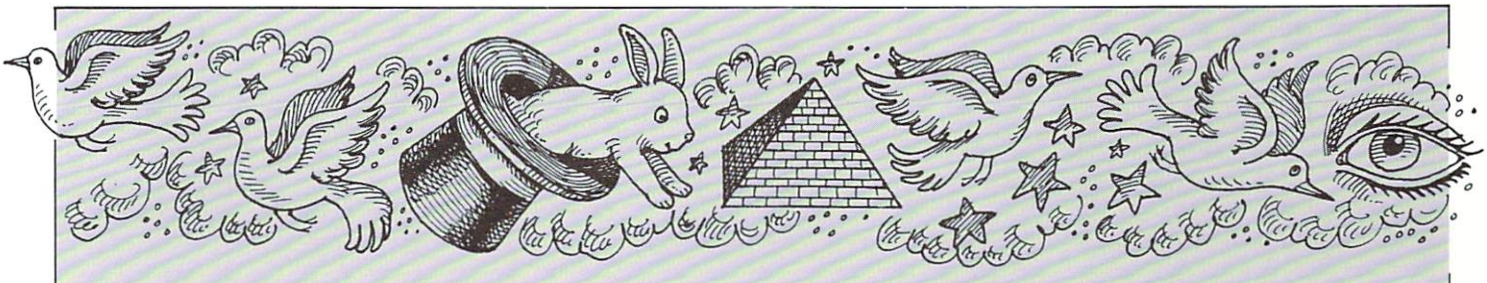
TO SEE YOUR PROGRAM DISPLAYED LINE-BY-LINE — for editing purposes, type the word "LIST" and hit the RETURN key.

TO CHANGE A PROGRAM LINE — either use the CURSOR keys to move to the line and change it, then hit RETURN . . . or . . . RETYPE the line number and hit RETURN (the new line with the same number will replace the old one with that number).

TO DELETE A PROGRAM LINE — simply type the line number of the line you want to delete and hit RETURN (to delete line 10 just type the number 10 and hit RETURN).

YOU TYPE THIS:
10 PRINT "MONTHLY INCOME:"
20 INPUT M
30 PRINT "YOUR INCOME IS" M
(type RUN & hit RETURN)

VIC SCREEN SHOWS THIS
MONTHLY INCOME: ← **VIC displays this and "waits"**
? ← **You enter this**
1000
YOUR INCOME IS 1000 ← **VIC responds**



How VARIABLES Work With INPUT Statements

VARIABLES allow the VIC to accept information typed in by the user and then use that information in the program! Variables are in fact the key to interactive programming (this series places a lot of emphasis on variables because they're so important . . . your ability to program increases greatly if you understand how variables work).

You can think of a variable as a "storage compartment" where the VIC stores the user's response to your prompt question. For example, you can write a program which asks the user to type in his name. In this case, you might assign the variable **N$** to the name typed in. Now every time you PRINT N$ in your program, the VIC will automatically PRINT the name the user typed-in! Type the word NEW, hit RETURN, and try this example:

```
10 PRINT“YOUR NAME”:INPUT N$
20 PRINT“HELLO,”N$
```

You might have noticed that we used variables like A,B,C and M to represent numbers in earlier examples, but here we use the variables N$ to represent a name. We used N to remind ourselves that this variable stands for "NAME" and we used a dollar sign to signify a string variable. This is important because variables come in two flavors: numeric variables and string variables.

Numeric variables are used to store number values such as 1, 100, 4000, etc. A numeric variable can be a single letter (A), any two letters (AB), a letter and a number (A1), or two letters and a number (AB1). You can save a little memory by using shorter variables but letters and number (A1, A2, A3) are often best if working with different categories in the same program.There is also a special type of numeric variable called an integer variable which eliminates decimal places from your numbers (helpful if you want calculations represented as whole numbers instead of numbers with long decimals). To get integer (whole) numbers, simply put a percent (%) sign at the end of your variable name (i.e. A1% or AB%) which makes the VIC treat all numbers typed in as whole numbers only, dropping any decimal places. Note that the percent sign here doesn't mean you are calculating real percentages. It's simply a symbol used with integer type variables.

String variables look just like numeric variables except they end with a dollar sign ($) like A$ or A1$. String variables are used to store words, phrases, sentences, graphic symbols (i.e. VICGRAPHICS) and "numbers which are used like words." By "numbers used like words" we mean numbers which will not be used in a calculation. Your social security number is an example of a number which is identified with a string variable instead of a numeric variable, because your SSN is used like a "label" instead of a "value."

Another good example is your **age.** If you ask the user to type his age for "reference" in a program, you might use a string variable because you will not be doing any calculation using his age . . . BUT . . . if you plan to calculate his date of birth by subtracting his age from this year, you are using age as a **number value** and must use a numeric variable. String variables take the same form as numeric variables, except they are followed by the **S-shaped dollar sign** ($). Examples of string variables are: A$, AB$, A1$, AB2$, and so on. Here are some examples:

Example 1

```
10 PRINT“ENTER A NUMBER”:INPUTA
20 PRINTA
```

Example 2

```
10 PRINT“ENTER A WORD”:INPUT A$
20 PRINTA$
```

Example 3

```
10 PRINT“ENTER A NUMBER”:INPUT A
20 PRINT A“TIMES 5 EQUALS”A*5
```

Note in Example 3 that the MESSAGES or PROMPTS are INSIDE the quotation marks, while the variables are OUTSIDE. In line 20, PRINTed A (the number you entered when prompted), then the message "TIMES 5 EQUALS," and then a CALCULATION (multiply the number A*5).

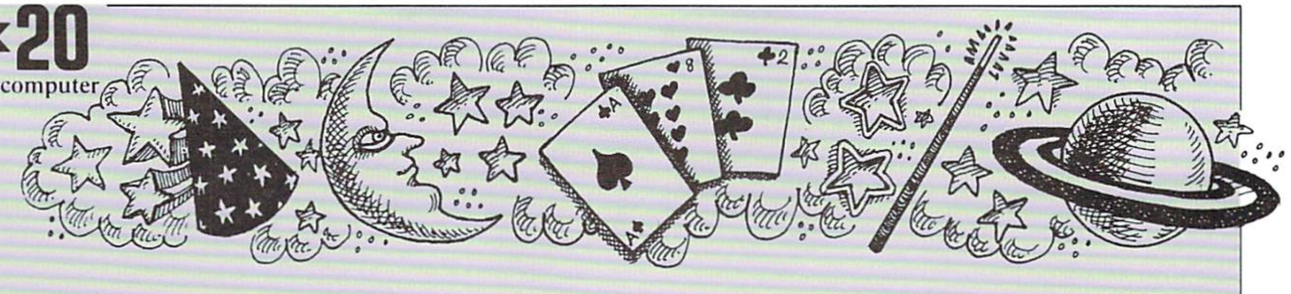
Calculations Make Practical Programs "Practical"

CALCULATIONS are important in most programs, from games to business aids. You have a choice of using either "actual numbers" or variables when doing calculations inside a program, but if you're working with numbers supplied by the user you must use numeric variables. Begin by asking the user to type in two numbers, like this:

```
10 PRINT“TYPE 2 NUMBERS”:INPUTA:INPUTB
```

Now multiply those numbers together to create a new

➔



variable C (C is number A multiplied by number B). The calculation uses the variables like this:

```
20 C=A*B
```

. . . and this lets you now PRINT the result as a MESSAGE:

```
30 PRINTA"TIMES"B"EQUALS"C
```

Enter these 3 lines and RUN the program. Note again that numeric or string variables are always OUTSIDE the quotation marks but any message information must be INSIDE the quotes. That's why we put words like TIMES and EQUALS inside quotation marks. Similarly, if we want a dollar sign in front of a number we've calculated, the dollar sign ($) must be PRINTed INSIDE quotes, in front of the numeric variable, like this (hit RUN/STOP and RESTORE, then enter this line and hit RETURN, then type RUN and hit RETURN):

```
40 PRINT"$"C
```

The dollar sign goes in quotes because the variable C only represents a number and can't contain the dollar sign. If the number represented by C was 100 then the VIC screen will display $100. However, if you tried to PRINT $C without putting the dollar sign in quotes, you'll get a SYNTAX ERROR IN LINE 10 error message because the dollar sign has to be inside the quotes.

One last tip about dollar signs. If you like, you can, in the first or second line of your program, create a variable that you can substitute for the dollar sign for use with numeric variables. If, for example, you type:

```
10 Z$="$"
```

. . . you can use the string variable Z$ wherever you want a dollar sign. Try this:

```
10 Z$="$":INPUTA
20 PRINTZ$A
```

The key is the Z$A in line 10. We begin by defining the dollar sign as a string variable called Z$. Then we INPUT a number which we call A. Finally, we PRINT both variables side by side and we get Z$ which is the dollar sign, and our number A printed together on the screen. By doing this, you can ALWAYS use Z$ (or whatever string variable you choose) next to a numeric variable when you need a dollar sign . . . and you'll probably find it easier than typing "$" every time you want to calculate dollars.

Time Delay Loops

TIME DELAY LOOPS are very helpful in slowing down a program, where appropriate. A full time delay loop looks like this: FORT=1TO1000:NEXT and can be inserted almost ANYWHERE in your program, either on a separate program line or INSIDE a line, separated by colons (:). Change the duration of the delay by changing the line, separated by colons (:). Change the duration of the delay by changing the "1000" to a larger number (slows down delay) or smaller number (speeds up).

A Word About Cosmetics

COSMETICS include all the small extras which you can add to your program after the logic is working. Here's a short list of "extras" to consider as you polish and refine your program . . . but don't worry about these cosmetics until the LOGIC of the program is finished and working!

- Insert blank lines where appropriate to make your screen display more readable (a PRINT statement on a line by itself automatically inserts a blank line when the program is RUN).

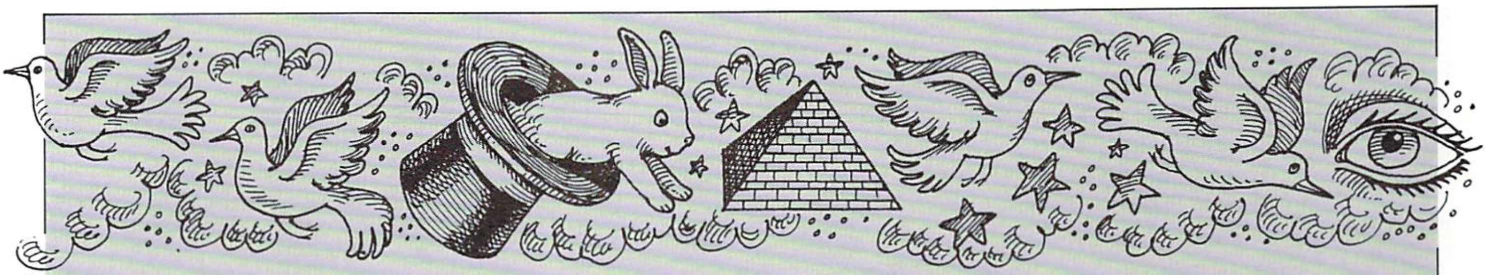
- Consider different colors . . . check your user's guide to see how POKE36879,X works and consult the SCREEN AND BORDER COLOR appendix for color variations. Use this POKE as the first or second line in your program to make the opening display appear in special colors.

- Sound effects can jazz up your program. A good one-line sound effect format is described below.

- Title your program with a REVERSE title at the top . . . or . . . use the special title technique described under COSMETICS (see below).

- Use upper and lower case letters instead of all capitals. Just type PRINTCHR$(14) in your program to switch to Upper/Lower case and type PRINTCHR$(142) to switch back to Upper Case Only. These PRINTCHR$ statements should be used like this: 10 PRINTCHR$(14)"Income"

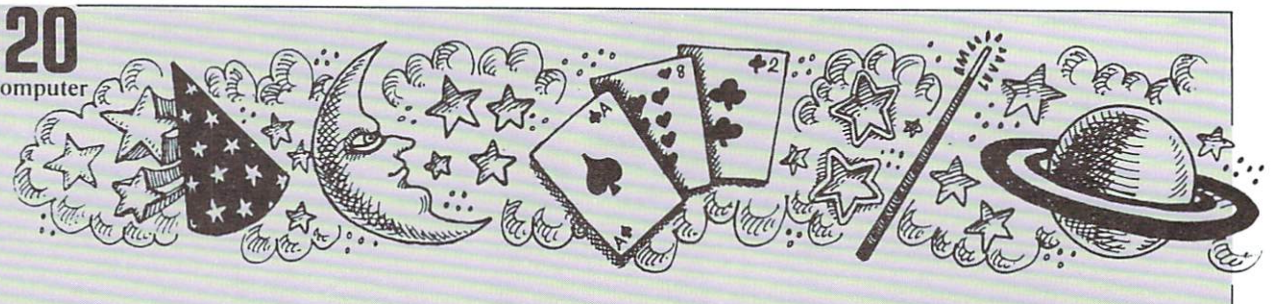
- Graphics are easy on the VIC-20, which includes a full set of "business" graphics which give you different types of lines and bars which allow you to separate different items by lines or bars, add highlights, etc.



INCOME/EXPENSE BUDGET EXAMPLE

```
  5  PRINT“ ♥ ”                                                  (clears the screen)
 10  PRINT“MONTHLY INCOME”:INPUT IN                               (PRINT/INPUT statement)
 20  PRINT                                                        (Inserts blank line)
 30  PRINT“EXPENSE CATEGORY 1”:INPUT E1$                          (Expense Cat 1= E1$)
 40  PRINT“EXPENSE AMOUNT”:INPUT E1                               (Expense Amt = E1)
 50  PRINT                                                        (Blank line)
 60  PRINT“EXPENSE CATEGORY 2”:INPUT E2$                          (Expense Cat 2 = E2$)
 70  PRINT“EXPENSE AMOUNT”:INPUT E2                               (Expense Amt 2 = E2)
 80  PRINT                                                        (Blank line)
 90  PRINT“EXPENSE CATEGORY 3”:INPUT E3$                          (Expense Cat 3 = E3$)
100  PRINT“EXPENSE AMOUNT”:INPUT E3                               (Expense Amt 3 = E3)
110  PRINT“        ”                                              (CLEAR SCREEN)
120  E=E1+E2+E3                                                   (Add Expense Amts = E)
130  EP=E/IN                                                      (Calculate Expense/Income%)
140  PRINT“MONTHLY INCOME: $”IN                                   (Display Income)
150  PRINT“TOTAL EXPENSES: $”E                                    (Display Total Expenses
160  PRINT“BALANCE EQUALS: $”IN-E                                 (Display Income – Expenses)
170  PRINT                                                        (Blank line)
180  PRINTE1$“=”(E1/E)*100“% OF TOTAL EXPENSES”                   (Lines 180-200 calculate
190  PRINTE2$“=”(E2/E)*100“% OF TOTAL EXPENSES”                    % each expense amount is
200  PRINTE3$“=”(E3/E)*100“% OF TOTAL EXPENSES”                    of total expenses)
210  PRINT                                                        (Blank line)
220  PRINT“YOUR EXPENSES =“EP*100”% OF YOUR TOTAL INCOME”         (Display E/I%)
230  FOR X=1 TO 5000:NEXT:PRINT                                   (Time Delay Loop)
240  PRINT“REPEAT? (Y OR N)”:INPUTY$:IFY$=“Y”THEN5:IFY$=“N”THENGOTO250
250  PRINT“ ♥ ”:END
```

→



A Screen Title Format

If you REPLACE LINE 5 in the program example, you can include a screen title which looks like the title is part of the BORDER.

```
5 POKE36879,30:PRINT " ♥ "
CHR$(18)"     INCOME/EXPENSE PROGRAM"
```

We used Line 5 because that makes it the first line in our program — the important thing is to make the screen format program the first line in the program (so the title comes on first). Note that a CLEAR SCREEN command will erase this title so if you clear the screen anywhere later in your program, you'll have to put this line in again, immediately after you PRINT the CLEAR SCREEN.

This is an illusion which actually comes from REVERSING the top line in the screen area to make it APPEAR to be part of the border. The program line does the following things, in order:

1. POKEs the border to dark blue.
2. CLEARS the screen.
3. Turns on REVERSE lettering
   (hold down CTRL and type RVS ON key).

The result is a title in white letters across the top of the screen . . . actually the letters are inside the screen area, but because we REVERSED the top line, it looks like part of the border.

Changing Screen/Border Colors

Try adding the following line to your program:

```
5 POKE36879,X:PRINT " ♥ "
```

where X can be any number between 1 and 255.

Adding Sound Effects

To use a sound effect, it's usually a good idea to include this line at the beginning of your program (it would normally start at line 10 as the first program line but we're using line 2 to add it to the beginning of our program above):

```
2 POKE36878,15:S1=36874:S2=36875:
S3=36876:S4=36877
```

Now you can use any of the VIC's four speakers by referring to S1, S2, S3 and S4. S1 is the deepest (lowest) voice and S3 is the highest (soprano). S4 is used for "white noise sound effects." Here is a sound effect you can add to your program above. Simply insert this line wherever you want a "beep" in your program (for example at Line 35):

```
35 POKES3,200:FORT=1TO200:NEXT=POKES3,0
```

VIC Glitch

Some minor problems have been brought to our attention by users attempting to type two (2) of the programs in the appendix of the VIC Manual.

In the first program, "TANK-V-UFO," a problem exists in line 135. That line is exactly 88 characters long if typed without the space after the line number. Users are unable to enter this line into the program. The following solutions will correct the problem:

1. Type the line exactly as is but do not type the space after the line number. Then after the last character of the line, the 0 in 170 is typed, the cursor will be on a blank line. Pressing return now will enter nothing. Before pressing return have the User 'cursor up' to the 1 of 135, then press return.
2. The line may be typed using the abbreviated commands:
   P shifted E for PEEK
   P shifted O for POKE
   G shifted O for GOTO

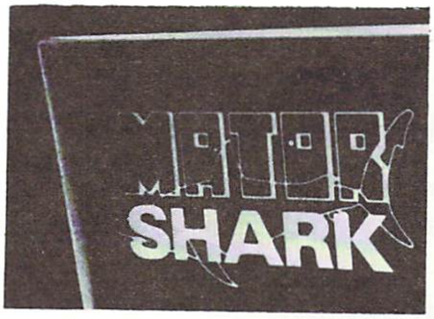
To edit the line, the space after the line number will have to be deleted and the cursor repositioned as in entry method 1, and the last character "0" re-entered into the line.

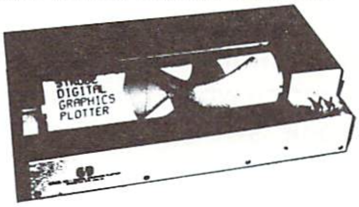
In the second program, "KILLER COMET," users are having difficulty in trying to find the reverse T character in line 0. This is the delete character the author used to remove the line number and the REM during listing of this program.

Either ignore this line entirely, do not enter it, or enter it without the quotes. e.g. 0 REM KILLER COMET BY . . . ETC.

If you're interested in this character, you can type:

1. 0 REM" then press return
2. With cursor controls repositioned cursor to the position just after the quotes
3. Press shifted insert 6 times
4. Press delete 6 times
5. Press quotes
6. Proceed entering the remainder of the line ■





VIXEL™

Vixel Volume One

Fire

Fly a water-dropping helicopter, and try to put out the high-rise fire before it spreads.

Draw

Be an artist! This high-resolution drawing program makes it easy to create pictures on the screen, and then save them on tape.

Race

Race the computer, head-on! Simple but lots of fun.

Vixel Volume Two

Superfont

Design programmable characters on your VIC-20 with this easy-to-use program. Especially useful when creating animations, since you can edit four characters at once as a 2 × 2 block. SUPERFONT creates DATA statements after you have finished designing the characters.

Safari

You are a photographer on an African Safari in this great game. The jungle animals run past as you try to snap their pictures. An excellent example of how to use large blocks of programmable characters on the VIC to create animation effects.

Quix

How good is your memory? QUIX presents patterns of color and sound that gradually get longer and harder to remember.

The programs Fire, Draw and Race are available on VIXEL Volume One for only $12.95. The programs Superfont, Safari and Quix are available on VIXEL Volume Two which is also $12.95. Both Vixel #1 and Vixel #2 work with the standard 5K VIC-20. Foreign orders add $3.50 for shipping. CA residents add 6% tax. VISA and MasterCard welcome. Please add $1.00 shipping for credit card orders.

VIXEL is a trademark of The Code Works.
VIC-20 is a trademark of Commodore Business Machines.

(805) 683-1585

THE CODE WORKS

Box 550
Goleta, California 93116
USA

ULTIMAX and Commodore-64 Will be Available Soon

Two new Commodore computer systems, complementing the current VIC line at both the high and low end, will soon be appearing on your dealer's shelves. Both these new products are breakthroughs in terms of both price and performance, and will be of real interest to anyone who currently owns Commodore equipment or who is considering a personal computer.

ULTIMAX

The new low-end computer, called the ULTIMAX, provides surprising graphic and music capabilities, and will teach computing with the use of a BASIC language plug-in cartridge. It has a flat-panel keyboard and contains a small amount (2K) of RAM (Random Access Memory).

You can also add a cassette recorder to the ULTIMAX for program loading and saving.

Some of the software initially available for the ULTIMAX includes a number of "space" games, a "music maker" cartridge, and converted versions of programs currently running on the VIC 20, such as Sargon II chess and the Bally/Midway arcade games like GORF and OMEGA RACE.

COMMODORE-64

At the other end of the home computing spectrum is the COMMODORE-64 which, on the surface, looks a lot like a VIC 20, but packs features normally found only on much larger microcomputer systems. The COMMODORE-64 contains a huge 64K RAM, can accept a Z80 microprocessor on a plug-in cartridge, and supports multiple levels of high-resolution graphics. It is a sophisticated personal computer for those who can make use of its power and capabilities, and can even handle small business applications.

Besides looking like a VIC 20 clone, the COMMODORE-64 contains many of the same interfaces as the VIC, including an 8-bit user port for attaching the VIC modem or accessory communication cartridges. A cassette interface allows use of programs and files created on all other Commodore computers. This compatibility means most BASIC programs written for 40-column PET computers will run without modification.

The only exceptions to that are programs that POKE screen memory locations, an area that is different on each Commodore system. To ease that incompatibility problem, Commodore will have a PET emulator that will eliminate BASIC program conversion and make the '64 operate like a PET in most areas. Machine language would still need some work to operate properly, however.

The COMMODORE-64, through a serial port, can also use VIC peripherals such as the VIC single disk drive and VIC graphic printer. With the addition of an IEEE-48 cartridge, the '64 will run any Commodore peripheral, such as a dual disk drive or CBM printer.

Although all the games designed for the ULTIMAX will operate on the COMMODORE-64, the '64 is not a game machine. In fact, with the 64K of RAM (about 40K is user accessible for BASIC programs and 52K for machine language programs), you can put some real business applications on this machine. Versions of the popular WordPro and WORDCRAFT word processing programs will be available, along with an "electronic spreadsheet" package. Data base programs and sophisticated personal financial applications are also part of athe first software offerings planned for the COMMODORE-64.

One of the most impressive features of the COMMODORE-64 is its ability to run CP/M®, using the Z80 cartridge, giving users access to one of the largest collections of microcomputer software available.

The Heart of the Systems

Although they are designed for very different purposes, both new computers have certain things in common. Both rely on state-of-the-art integrated circuits designed and produced by Commodore's MOS subsidiary. A new 6500 family microprocessor, the 6510, is common to both computers. It uses the same instruction set as the familiar 6502 — the heart of all other Commodore systems — but contains additional input/output (I/O) lines to handle the processing required by the new systems.

The 6510 microprocessor, upward compatible from the 6502, allows machine language programs running on other Commodore computers to be easily converted to run on the ULTIMAX or COMMODORE-64.

Graphics Capabilities

Both machines also rely on a new display chip to handle all the display characteristics that normally would require a character generator and other supporting circuits for color and graphics.

The video display produced by both machines is 40 columns by 25 lines, with 255 foreground/background color combinations, 16 text colors and all 64 PET graphic characters. Additionally, the user can construct programmable characters to replace the normal character set.

Both machines have a high-resolution graphic mode of 320 by 200 points (pixels) and can use 16 colors simultaneously. To facilitate game graphics and animation, 256 independently movable display objects can be created, with up to 8 objects per line. Each object is 21 by 24 pixels in size, and can be up to 3 colors. Both computers provide collision detection between objects, so a program can tell if one object hits another and can determine what to do next. You can also select object display priority to determine whether one object will move in front of or behind another.

Independent magnification in both horizontal and vertical directions for each object is also possible, to add a lot of versatility to creating graphics. To make movement easier and smoother, you can scroll objects in horizontal and vertical directions, pixel by pixel.

In addition, the COMMODORE-64 has a number of other high-resolution modes that are not possible with the ULTIMAX. These extra modes give you additional colors in each pixel zone and allow more flexibility in designing graphics.

Sound Capabilities

Another feature of both machines is the SID (Sound Interface Device) sound synthesizer circuit. It can produce music and sound that rivals some of the dedicated music synthesizers now available.

The SID produces three independent voices, each with a nine-octave range. Four waveforms are possible: sawtooth, triangle, variable pulse and noise. The sound synthesizer also contains a programmable ADSR (attack, decay, sustain, release) generator, and a programmable filter, independently selectable for each voice, that contains low pass, high pass, band pass and notch outputs. The sound synthesizer also has variable resonance and a master volume control.

With all these sophisticated features you have almost complete control over the type of sound produced by either machine. Hooked up to a good quality audio system, you'll be amazed at the orchestration you can command.

Other Similarities

In addition, both machines can accept a variety of plug-in program cartridges. These cartridges are small — about 2 inches by 2½ inches — but can contain RAM or ROM (Read Only Memory). The cartridges will allow up to 16K of ROM and 2K of RAM.

Both machines also have two game controller sockets. Each socket will accept a joystick, double paddle or lightpen. And each computer has a direct audio and video output for connection to an audio amplifier and video monitor. An RF modulator is supplied for hook-up to a standard TV set.■

–Mike Heck

CP/M is a registered trademark of Digital Research, Inc.

QUICK REFERENCE FACT SHEET

	ULTIMAX	Commodore-64
Memory	2K built-in 2K RAM, 16K ROM add-on	64K built-in 16K ROM add-on
Screen Size	40 col × 25 lines	40 col × 25 lines
Graphics	320 × 200 pixels 225 foreground/ background color combo 16 text colors 64 graphic characters 256 movable Sprites Independent magnification	320 × 200 pixels 225 foreground/ background color combo 16 text colors 64 graphic characters 256 movable Sprites Independent magnification Extended Hi-res modes
Sound	6581 Sound Interface Device (SID) 3 independent voices, 9 octaves each Programmable ADSR Programmable filter Variable resonance & master volume control	6581 Sound Interface Device (SID) 3 independent voices, 9 octaves each Programmable ADSR Programmable filter Variable resonance & master volume control
Games	All Commodore games on cartridge	All Commodore games on cartridge
Peripherals	Datassette Joystick Double paddle Lightpen	Datassette Joystick Double paddle Lightpen Z80 microprocessor on cartridge VIC MODEM VIC disk drive VIC graphic printer With IEEE-48 8 cartridge: All Commodore peripherals
Compatibility	Same game cartridges will work on Commodore-64.	Other BASIC Commodore programs easily converted. PET emulator to be available.

VIC Mail

Overview
VIC-MAIL is a general mailing list program for the VIC 20. It is designed to run on the unexpanded VIC with 5K of RAM.

The program allows you to enter names, addresses and miscellaneous data, and then record this data on cassette tape. During data entry you can easily edit any of the fields before writing the data to cassette.

A second part of the program, Display/Print Records, reads back the data you recorded and allows you to look at the data or print mailing labels on the VIC graphic printer.

Because of the sophistication of the program and the small amount of memory on the VIC, certain limitations are present in the program.

As each record is entered, it is immediately recorded on tape. This is because there is not enough memory left for much internal data storage. Due to this design, you cannot access individual records without sequentially reading each back into the VIC.

Operation
Load the program with the normal LOAD function. After the program is loaded, type RUN.

After a brief opening message, the following display will be shown:

```
------------------------------------------------------
                    < VIC MAIL >
------------------------------------------------------
                   Please Select

            A    ADD AN ADDRESS
            D    DISPLAY/PRINT RECORD
            E    END PROGRAM
```

AFTER THE PROGRAM IS RUNNING, REMOVE THE PROGRAM TAPE AND INSERT A DATA TAPE IN THE DATASETTE RECORDER.

Add An Address

Selecting 'A' from the Main Menu will allow you to enter names and addresses. The system will first prompt with:

```
PLEASE ENTER STARTING
RECORD NUMBER: ?
```

Enter the number of the name you are starting with. For example, if this is the first name you are entering into the system, enter a '1.' The system will keep track of the numbers from now on.

Before you exit the system it will display the last record number, so you can start with the next higher number when you enter data at another time. After you enter the record number, the following data entry screen will be displayed:

```
RECORD NUMBER: 1
F. NAME:?
L. NAME:
HOUSE #:
ROAD:
CITY:
ST, ZIP: ..   .....
MISC: ...-....
```

The cursor will be positioned next to the first name field, ready for you to start entering data. Just type the information asked for. If you make a mistake, use the INST/DEL key to erase the mistake. After each field is completed, hit RETURN to move to the next field.

After all fields are completed, the following will be displayed:

```
CORRECT (Y/N)
```

If everything is correct, hit 'Y' for yes. If you want to change anything, enter 'N' and the computer will display:

```
      CHANGE WHAT FIELD:
 F NAME  L NAME  HOUSE#
   ROAD  CITY  ST  MISC
```

Just hit the letter of the field you want to change and the cursor will be positioned at that field. Type the new information and then RETURN to get to the next field. If the remaining fields are correct just hit RETURN until you go through all the fields. At that point you will be prompted with the CORRECT prompt again. If everything is correct hit 'Y' and you will then be prompted with:

```
WRITE RECORD (Y/N)
```

If you want to enter the record on tape, type 'Y.' The first time you enter a record, you will be prompted with:

```
PRESS PLAY AND RECORD ON TAPE
```

After that record is entered you don't have to touch the keys on the datasette, as the program will control starting and stopping the recorder.

After the record is entered, the system will prompt with:

```
ENTERED, ANOTHER NAME (Y/N)
```

If you want to enter some more names, type 'Y' and you will be returned to a blank data entry mask. If you answer 'N' for no, the computer will display:

```
LAST RECORD NUMBER:
X
```

and then say:

```
HIT < RETURN > TO CONTINUE
```

Copy down the last record number and hit RETURN to return to the Main Menu.

Display Print Records
Select 'D' from the main menu to Display or Print the information you entered.

The computer will prompt with:

PLACE DATA TAPE
IN DATASETTE

PROCEED (Y/N)

Rewind the data tape and place it in the datasette. Then enter 'Y' to continue. You will then be prompted with:

PRESS PLAY ON TAPE

As each record is read from tape, it is displayed on the screen. You are then prompted with:

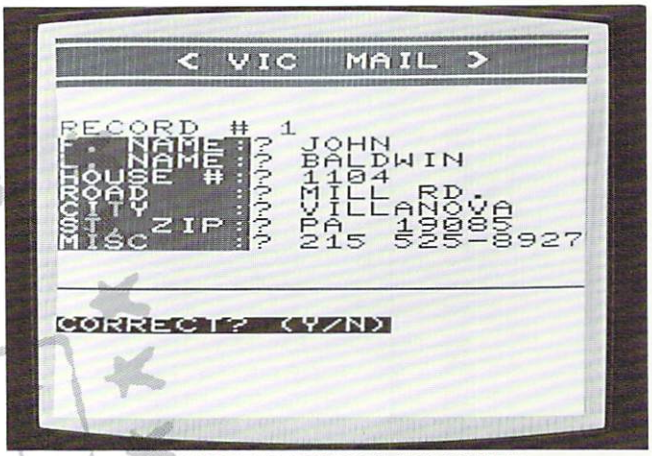
PRINT ADDRESS (Y/N)

If you enter 'Y' the address is printed. If you enter 'N' the following is displayed:

NEXT ADDRESS (Y/N)

NOTE: IF YOU DON'T HAVE A PRINTER ATTACHED, A "DEVICE NOT PRESENT" ERROR WILL OCCUR. JUST TYPE RUN AND CONTINUE FROM WHERE YOU LEFT OFF. YOU DON'T HAVE TO REWIND THE TAPE.

If you answer 'Y' then the next address is read from tape and the above procedure repeated. If you answer 'N' then:



```
< VIC MAIL >

RECORD # 1
F. NAME :? JOHN
L. NAME :? BALDWIN
HOUSE # :? 1104
ROAD    :? MILL RD.
CITY    :? VILLANOVA
ST. ZIP :? PA 19085
MISC    :? 215 525-8927

CORRECT? (Y/N)
```

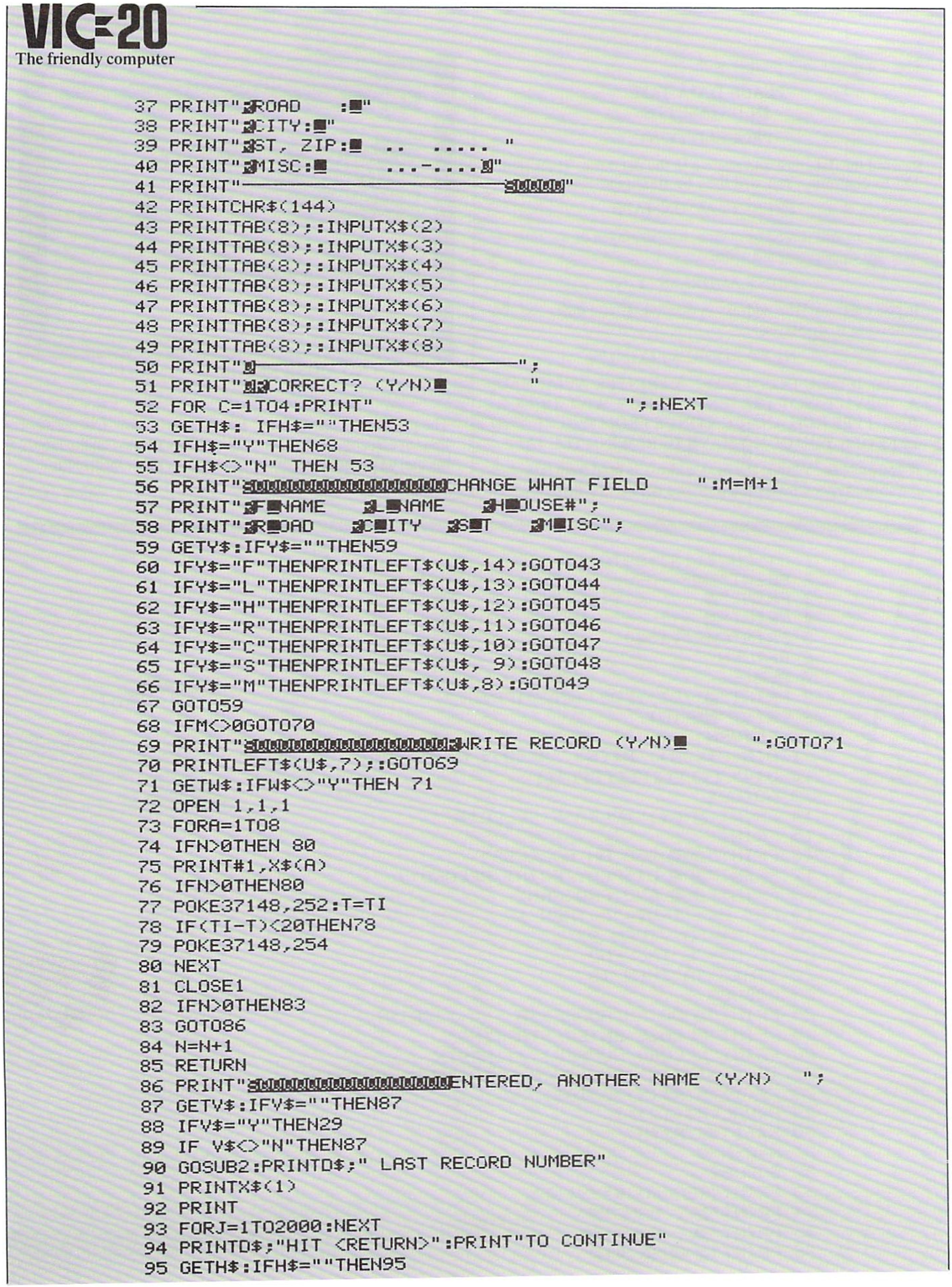
NO MORE RECORDS SELECTED

is displayed and you are returned to the main menu.

The program is set up for 1″ labels which allow five address lines to be printed. The Misc. field is not printed on the label, but can be used for telephone numbers or other information.

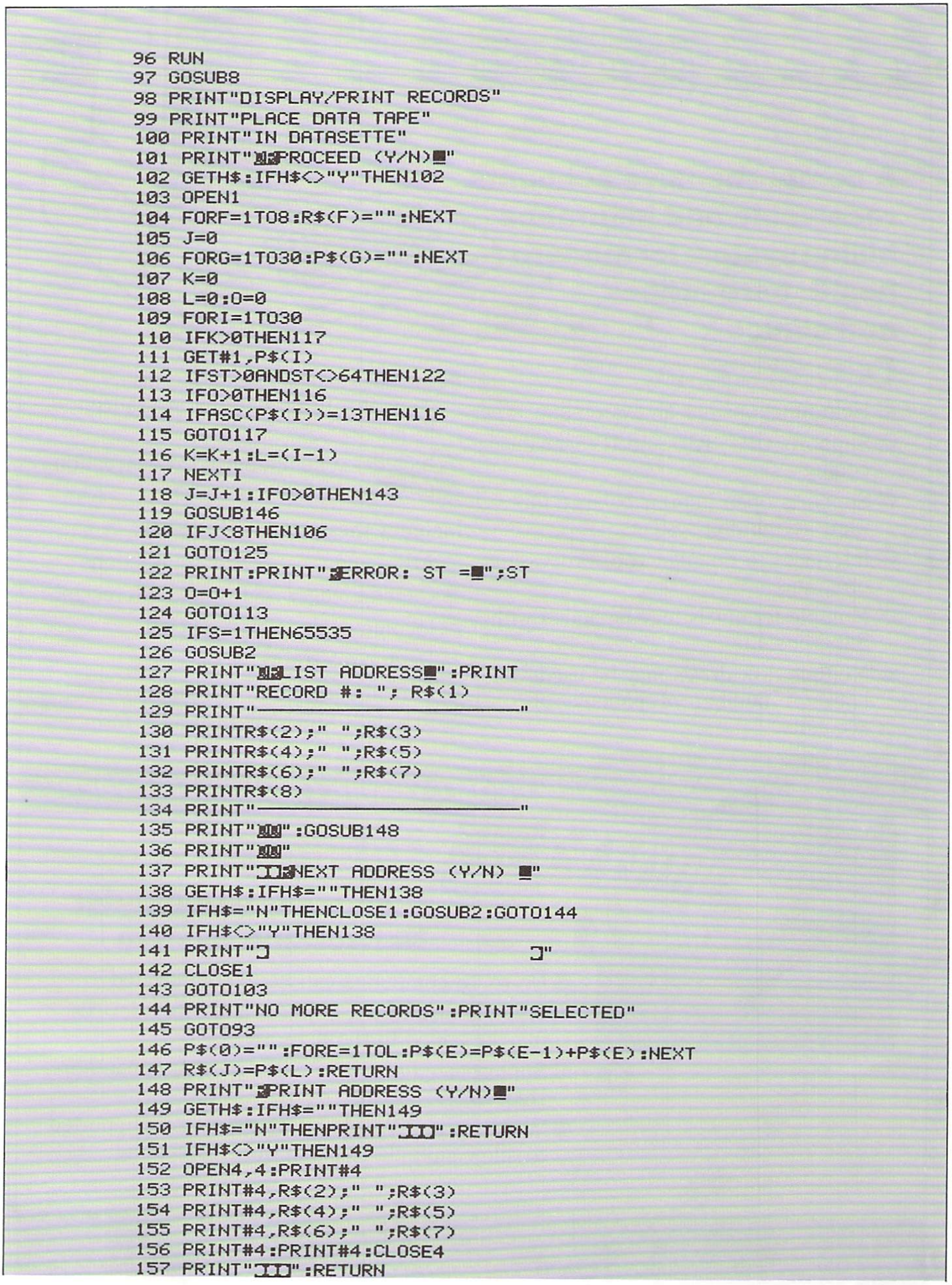
–Mike Heck

```
1 C$=CHR$(13):GOTO11
2 PRINT"⊐▨────────────────────■";
3 PRINT"▨      < VIC  MAIL >    ■";
4 PRINT"▨─────────────────────■"
5 PRINTCHR$(156)
6 RETURN
7 REM
8 PRINT"▧";D$;:FORJ=0TO5:PRINT"                              ";:NEXT
9 PRINT"▧▩";D$
10 RETURN
11 DIM P$(30), R$(8), X$(8): D$="▩▩▩▩"
12 U$="▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯"
13 GOSUB2
14 PRINT"▧"D$;"▨PLEASE SELECT■"
15 PRINT
16 PRINT"▨A■ ADD AN ADDRESS"
17 PRINT"▨D■ DISPLAY/PRINT RECORD";
18 PRINT"▨E■ END PROGRAM"
19 GET H$: IFH$="" THEN 19
20 IF H$="D" THEN 97
21 IF H$="E"THENEND
22 IFH$<>"A"THEN 19
23 GOSUB8
24 PRINT"▨ADD AN ADDRESS■      ":PRINT
25 PRINT"PLEASE ENTER STARTING"
26 PRINT"RECORD NUMBER:";:INPUTX
27 IF X<>INT(X)ORX<1THEN PRINT"▯▯▯":GOTO26
28 X$(1)=STR$(X):GOTO31
29 H=VAL(X$(1))+1
30 X$(1)=STR$(H)
31 N=0:M=0
32 GOSUB2
33 PRINT"RECORD #";X$(1)
34 PRINT"▨F. NAME:■"
35 PRINT"▨L. NAME:■"
36 PRINT"▨HOUSE #:■"
```



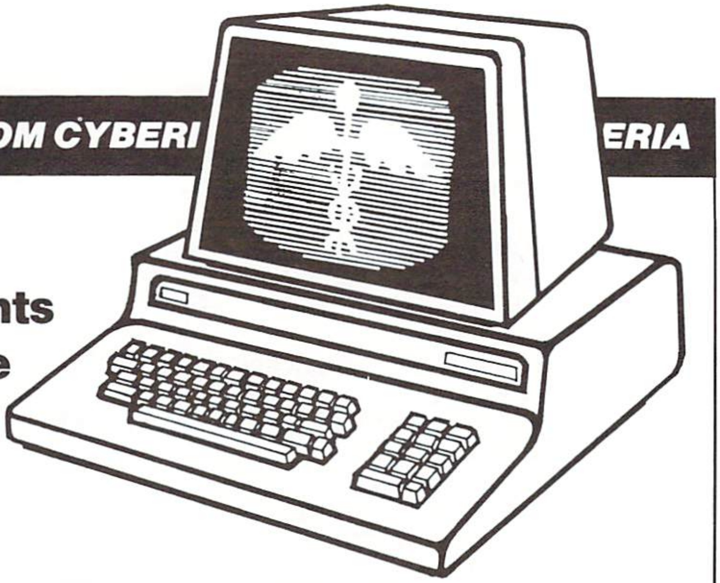
VIC=20
The friendly computer

```
37 PRINT"▨ROAD    :■"
38 PRINT"▨CITY:■"
39 PRINT"▨ST, ZIP:■  ..   .....  "
40 PRINT"▨MISC:■        ...-....▨"
41 PRINT"────────────────────────────SQQQQ"
42 PRINTCHR$(144)
43 PRINTTAB(8);:INPUTX$(2)
44 PRINTTAB(8);:INPUTX$(3)
45 PRINTTAB(8);:INPUTX$(4)
46 PRINTTAB(8);:INPUTX$(5)
47 PRINTTAB(8);:INPUTX$(6)
48 PRINTTAB(8);:INPUTX$(7)
49 PRINTTAB(8);:INPUTX$(8)
50 PRINT"▨──────────────────────────";
51 PRINT"▨▨CORRECT? (Y/N)■         "
52 FOR C=1TO4:PRINT"                       ";:NEXT
53 GETH$: IFH$=""THEN53
54 IFH$="Y"THEN68
55 IFH$<>"N" THEN 53
56 PRINT"SQQQQQQQQQQQQQQQQQQQQCHANGE WHAT FIELD     ":M=M+1
57 PRINT"▨F■NAME   ▨L■NAME   ▨H■OUSE#";
58 PRINT"▨R■OAD   ▨C■ITY  ▨S■T    ▨M■ISC";
59 GETY$:IFY$=""THEN59
60 IFY$="F"THENPRINTLEFT$(U$,14):GOTO43
61 IFY$="L"THENPRINTLEFT$(U$,13):GOTO44
62 IFY$="H"THENPRINTLEFT$(U$,12):GOTO45
63 IFY$="R"THENPRINTLEFT$(U$,11):GOTO46
64 IFY$="C"THENPRINTLEFT$(U$,10):GOTO47
65 IFY$="S"THENPRINTLEFT$(U$, 9):GOTO48
66 IFY$="M"THENPRINTLEFT$(U$,8):GOTO49
67 GOTO59
68 IFM<>0GOTO70
69 PRINT"SQQQQQQQQQQQQQQQQQQQQQ▨WRITE RECORD (Y/N)■       ":GOTO71
70 PRINTLEFT$(U$,7);:GOTO69
71 GETW$:IFW$<>"Y"THEN 71
72 OPEN 1,1,1
73 FORA=1TO8
74 IFN>0THEN 80
75 PRINT#1,X$(A)
76 IFN>0THEN80
77 POKE37148,252:T=TI
78 IF(TI-T)<20THEN78
79 POKE37148,254
80 NEXT
81 CLOSE1
82 IFN>0THEN83
83 GOTO86
84 N=N+1
85 RETURN
86 PRINT"SQQQQQQQQQQQQQQQQQQQQENTERED, ANOTHER NAME (Y/N)   ";
87 GETV$:IFV$=""THEN87
88 IFV$="Y"THEN29
89 IF V$<>"N"THEN87
90 GOSUB2:PRINTD$;" LAST RECORD NUMBER"
91 PRINTX$(1)
92 PRINT
93 FORJ=1TO2000:NEXT
94 PRINTD$;"HIT <RETURN>":PRINT"TO CONTINUE"
95 GETH$:IFH$=""THEN95
```



```
96 RUN
97 GOSUB8
98 PRINT"DISPLAY/PRINT RECORDS"
99 PRINT"PLACE DATA TAPE"
100 PRINT"IN DATASETTE"
101 PRINT"▣▚PROCEED (Y/N)■"
102 GETH$:IFH$<>"Y"THEN102
103 OPEN1
104 FORF=1TO8:R$(F)="":NEXT
105 J=0
106 FORG=1TO30:P$(G)="":NEXT
107 K=0
108 L=0:O=0
109 FORI=1TO30
110 IFK>0THEN117
111 GET#1,P$(I)
112 IFST>0ANDST<>64THEN122
113 IFO>0THEN116
114 IFASC(P$(I))=13THEN116
115 GOTO117
116 K=K+1:L=(I-1)
117 NEXTI
118 J=J+1:IFO>0THEN143
119 GOSUB146
120 IFJ<8THEN106
121 GOTO125
122 PRINT:PRINT"▚ERROR: ST =■";ST
123 O=O+1
124 GOTO113
125 IFS=1THEN65535
126 GOSUB2
127 PRINT"▣▚LIST ADDRESS■":PRINT
128 PRINT"RECORD #: "; R$(1)
129 PRINT"————————————————————"
130 PRINTR$(2);" ";R$(3)
131 PRINTR$(4);" ";R$(5)
132 PRINTR$(6);" ";R$(7)
133 PRINTR$(8)
134 PRINT"————————————————————"
135 PRINT"▣▣":GOSUB148
136 PRINT"▣▣"
137 PRINT"⊐⊐▚NEXT ADDRESS (Y/N) ■"
138 GETH$:IFH$=""THEN138
139 IFH$="N"THENCLOSE1:GOSUB2:GOTO144
140 IFH$<>"Y"THEN138
141 PRINT"⊐                           ⊐"
142 CLOSE1
143 GOTO103
144 PRINT"NO MORE RECORDS":PRINT"SELECTED"
145 GOTO93
146 P$(0)="":FORE=1TOL:P$(E)=P$(E-1)+P$(E):NEXT
147 R$(J)=P$(L):RETURN
148 PRINT"▚PRINT ADDRESS (Y/N)■"
149 GETH$:IFH$=""THEN149
150 IFH$="N"THENPRINT"⊐⊐":RETURN
151 IFH$<>"Y"THEN149
152 OPEN4,4:PRINT#4
153 PRINT#4,R$(2);" ";R$(3)
154 PRINT#4,R$(4);" ";R$(5)
155 PRINT#4,R$(6);" ";R$(7)
156 PRINT#4:PRINT#4:CLOSE4
157 PRINT"⊐⊐":RETURN
```

NEW FROM CYBERIA ★ NEW FROM CYBERIA ★ NEW FROM CYBERIA



Mari

A comprehensive accounts receivable and insurance billing system for modern health care offices and clinics

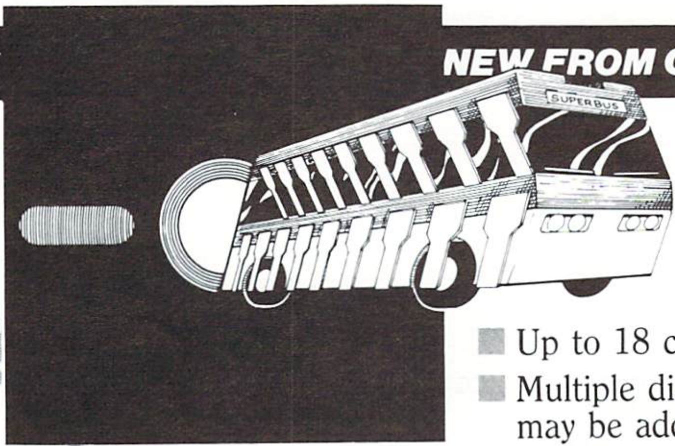
DESIGNED FOR COMMODORE 8000 SERIES COMPUTERS AND DISK DRIVES

- ▪ Help functions are always on-line
- ▪ Supports CPT, ICD and RVS medical coding
- ▪ Supports open item or balance forward accounting systems
- ▪ Interfaces to popular word processing programs
- ▪ Multiple terminals may be added with Superbus 4.0
- ▪ Multiple disk drives may be used—no limit on number of patients or accounts
- ▪ Includes a data base and forms generator to fill out any insurance form
- ▪ Includes a computer aided instruction program to train new users

Includes these standard reports:

- ▪ A/R aging
- ▪ A/R transactions
- ▪ General ledger
- ▪ Instant cash receipt
- ▪ Income analysis by physician
- ▪ Doctor referral report
- ▪ Patient/account cross reference
- ▪ Standard SuperBill insurance form

NEW FROM CYBERIA ★ NEW FROM CYBERIA



SUPERBUS 4.0

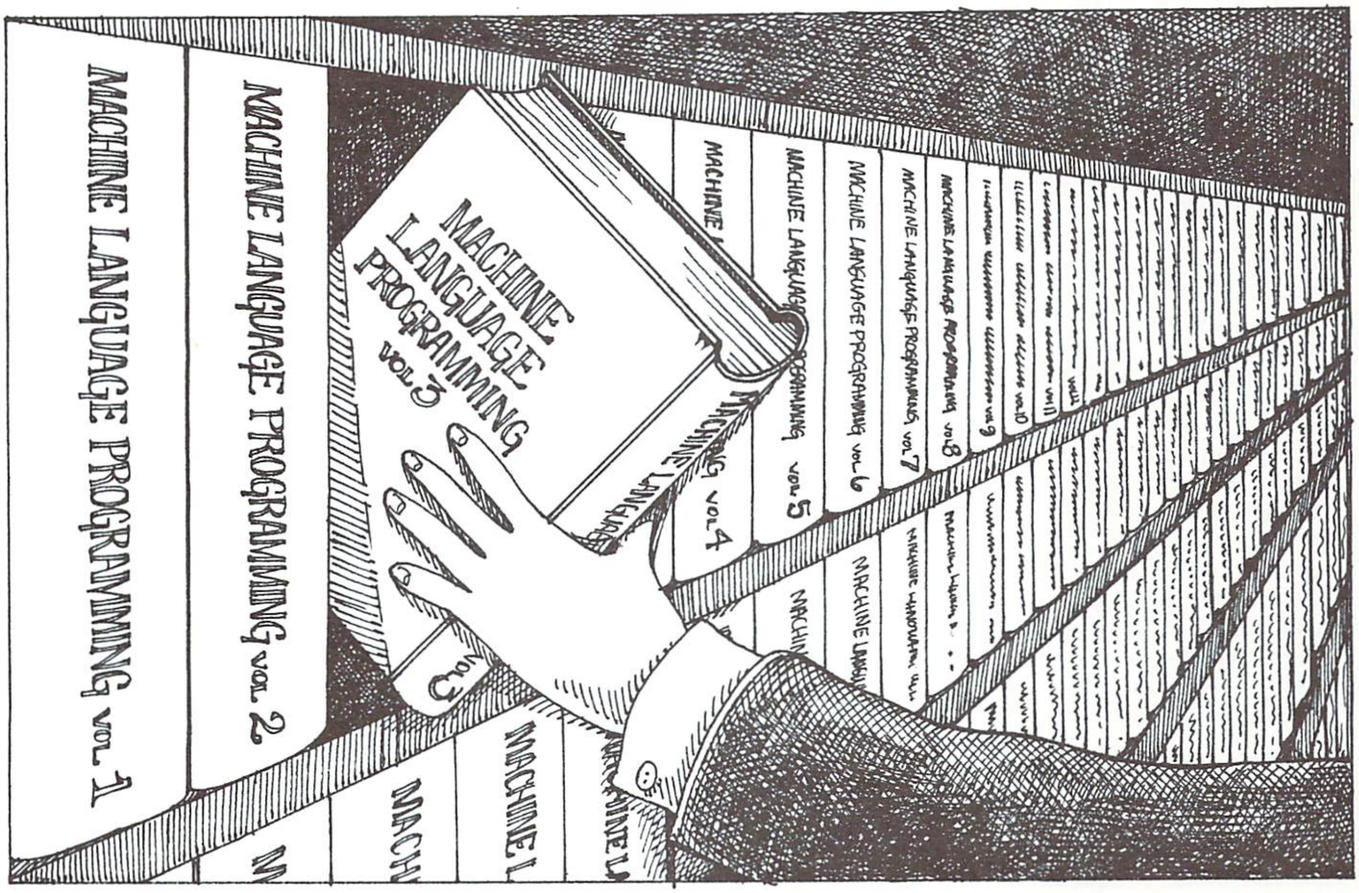
Commodore Computer Networking System

- ▪ Up to 18 computers can be interconnected
- ▪ Multiple disk drives, printers and other devices may be added to the network
- ▪ Sophisticated security system prevents unauthorized use
- ▪ WordPro and Wordcraft programs are supported to allow multiple terminal word processing

NEW ★ NEW ★ NEW

*WordPro is a trademark of Professional Software, Inc.

CYBERIA inc. 2330 LINCOLN WAY, AMES, IA 50010 • 515-292-7634


MACHINE LANGUAGE PROGRAMMING vol 1
MACHINE LANGUAGE PROGRAMMING vol 2
MACHINE LANGUAGE PROGRAMMING vol 3
MACHINE LANGUAGE PROGRAMMING vol 4
MACHINE LANGUAGE PROGRAMMING vol 5
MACHINE LANGUAGE PROGRAMMING vol 6
MACHINE LANGUAGE PROGRAMMING vol 7
MACHINE LANGUAGE PROGRAMMING vol 8
vol 9
vol 10

Machine Language Programming: Volume 3

In the last article on Machine Language Programming (December, 1981), you were introduced to some of the various modes, registers and instructions of the 6502 Instruction Set. There was also a short example which simply printed out some portion of the character set to the screen. Using that article as a background, we will, in this session, take our first step into the world of Assembly Language, look at the function of some more instructions, and finally look at a program which actually has a real life application.

Assembly Language

You may have thought, after reading the last article, that programming in machine language via the Monitor could really be a chore if you had to write a complete application. And you would have been correct, but at least we have a Monitor which conveniently allows us to type in a hex code at a certain memory location. Before our programming time, this wasn't possible. On early machines, programmers had 8 or 16 (or 32) toggle switches which could be set in one of two positions—on or off. To load an instruction, for instance a hex "60" (return), you would first have to set the eight toggle switches in the binary representation of 60, then push a "load" button to load the code into the machine. Our simple program from the last article might take an hour to load! But let's not dwell on the past, since higher level "english-like" languages have been developed for our use.

Assembly Language developed because even though sophisticated monitors and keyboards were developed, it still was a chore to remember and code in actual hex (or binary or octal) instructions, or look at a "hex dump" of a program and be able to see what the program did. (I do understand though, that programmers still exist who can do this at will, but they are kept chained inside the old ENIAC computer.) For the same reasons that higher level languages such as BASIC began to replace Assembler programming, Assemblers were developed to replace actual Machine Language programming.

What is an Assembler? An Assembler is basically a program (which can be written in any language, even BASIC) that reads in a source file that contains lines of mnemonic instructions and "assembles" each mnemonic instruction into its Machine Language equivalent. The nice feature about Assembly Language is that instead of having to remember all of the various hex forms of an instruction, you only have to remember one mnemonic. For example, the instruction to "Load the Accumulator" can be represented by "A9" in the immediate mode, "AD" in the absolute mode, "A5" in the zero page mode, and so on. In Assembly Language, the instruction is merely represented as "LDA" and the Assembler resolves which form the source line is representing. So you can see that writing in Assembly Language is much easier than hand-coding in Machine Language, yet it yields the same benefits.

If you turn now to the listing of the program accompanying this article, you will first notice that it is broken down into four distinct columns. The first column represents a line number which is arbitrarily assigned for reference purposes; the next column is the actual starting memory location at which this line of code will be loaded; the next column contains the actual hex code generated for that line; and finally, the actual Assembly Language source text.

(If ever you should read about or come across a program written in Assembly Language that might be useful to you, and the author provides a listing such as the one with this article, then all you need do to implement it is to type in the code and assemble it. If you don't have an Assembler, then simply go into the Monitor and begin typing in the code as listed in the third column, starting at the memory locations listed in the second column. When finished, do a Machine language "save" and you will have a copy of the program.)

Assembly Language Format

Without actually worrying about what exactly this program does, let's look at the program listing from a syntactical point of view. The first rule you will notice about this Assembler (Commodore's Assembler Development System) is that a comment line begins with a semicolon (;). In my listings, I generally put a lot of comments, and since there is no overhead associated with it, I feel it is a good practice for you to do as you develop an Assembly Language program.

The next portion of the program (lines 24-38) contains examples of the Assembly Language assignment statement. This statement allows you to assign a value to a variable (or label). If you precede the value with a "$", then the Assembler assumes a hex value, otherwise it assumes decimal (this is true whenever referring to values throughout your programs). Hexadecimal is usually more convenient because most memory maps and other listings are in this format. By using the assignment statement (in the beginning of a program ONLY), you tell the Assembler to use the associated value every time it encounters that label in the program. For instance, line 30 assigns the hex value "0070" to the variable "chrget". If you look at the actual code that references this variable (line 47) you can see the purpose of the assignment statement is: 1) to make your program more meaningful and easier to read; and 2) to make it easier to change (if the value for "chrget" changes in the future). In general, assignment statements are used to refer to the memory locations of routines (or subroutines) that are either part of the' operating system (as in this case), or some of your own routines.

The next part of the program, lines 47-50, is the main routine which basically executes three subroutines. Notice that instead of specifying an exact location to "jump to," I have used the labels that were defined in the beginning. The label "aschex" is not listed in the beginning of the routine because it is actually part of this program, at line 57. Notice that the Assembler generated the address of "0344" for this label because, as line 57 specifies, the memory location that the routine or label "aschex" would begin at is "0344."

As you can see by now, labels within an Assembly Language program work similarly to their counterparts in BASIC and other high level languages. So one obvious advantage Assembly Language has over Machine Language is the fact that all "jump to" and "branch to" (as in line 61) addresses can be resolved for you by the Assembler. The next part of the program, lines 57-82, is the actual routine which we will look at later in more detail.

The final part of the program at line 88 illustrates what is known as an Assembler directive. An Assembler directive is basically a command that tells the Assembler to perform some operation at "assemble time," but it is not a part of the actual program. In this Assembler, directives must be preceded by a period (.). The "byte" directive in line 88 tells the Assembler to set aside one memory location, and in this case, put a hex "0" there. In this instance, the memory location used is "0371." Notice that all the code that references the label "answer" (lines 59, 71, 76, 79, 80) contains the address "0371."

More Instructions to Work With

Let me first give you some background on the accompanying article. If you have already paged through this issue, you may have noticed another article which describes a routine that allows you to program the cursor directly from BASIC, using X,Y coordinates. For example, to move the cursor to line 15 and column 40, you could type "sys address,15,40". The "sys" command cannot directly pass the parameters "15" and "40", so one of the functions of the "sys routine" must be to accept these parameters.

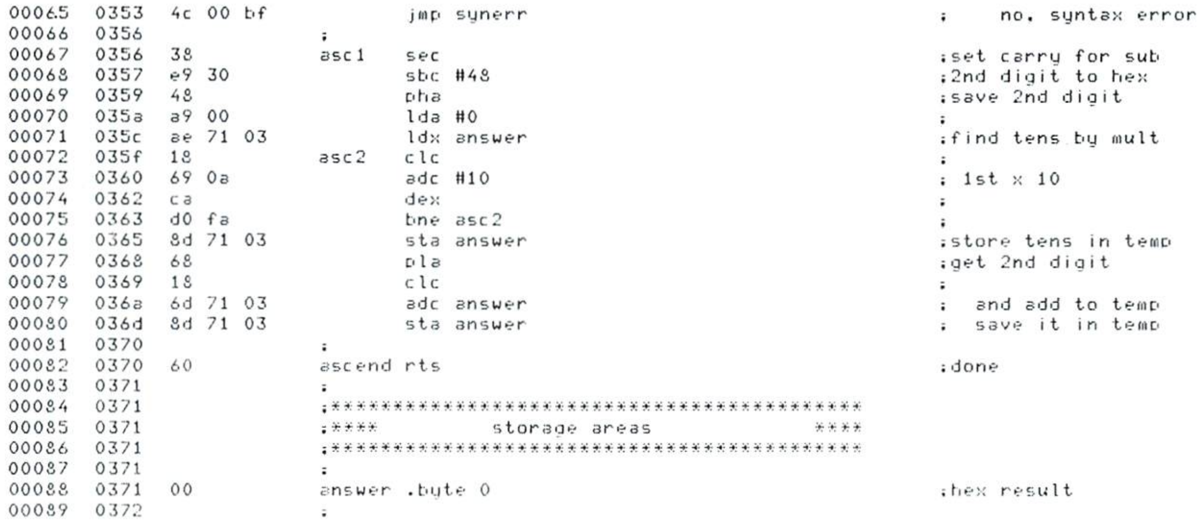
Sounds easy so far, right? Well yes, except that when you code "sys address,15,40", the number "15" is actually represented as two characters in memory—hex 31 (for "1") and hex 35 (for "5"). This is what is known as the ASCII character representation. If we want to be able to use the number "15" as some sort of counter (of the number of times to cursor down), then we must convert the ASCII representation of "15" to the actual numeric value. We will look here at the subroutine that does this.

Still sounds easy, right? Yes, except that this explanation overlooks one thing—how do we pull the number "1" and the number "5" from our BASIC statement. Well I won't explain in detail how this works in this article, except to tell you that your PET/CBM already comes with a routine which will do this. This routine is referenced in my listing by the label "chrget", and when it is invoked (via the "jsr"), the next character or token from the current BASIC line will be loaded into the Accumulator. And the pointer to the current character will be incremented so that it now points at the next character in the line.

Now let's look at the program.

```
1:aschex.src......page 0001

line# loc    code         line


00001  0000               ;*************************************************************************
00002  0000               ;****                                                                 ****
00003  0000               ;**** program name = aschex.src                                       ****
00004  0000               ;**** author = dave scott. cbm  us                                    ****
00005  0000               ;**** date = 03/03/82                                                 ****
00006  0000               ;****                                                                 ****
00007  0000               ;**** purpose: the purpose of this program is to convert a two digit ****
00008  0000               ;****          ascii number to hex.  the routine is entered via the  ****
00009  0000               ;****          "sys" command and the two digit number should follow  ****
00010  0000               ;****          the sys command as in the following example:          ****
00011  0000               ;****                                                                 ****
00012  0000               ;****          sys 826,25                                             ****
00013  0000               ;****                                                                 ****
00014  0000               ;****          where 826 is the sys address and 25 is the number to  ****
00015  0000               ;****          convert.  the converted number will be stored at      ****
00016  0000               ;****          location 0371 hex.                                     ****
00017  0000               ;****                                                                 ****
00018  0000               ;*************************************************************************
00019  0000               ;
00020  0000               ;********************************************
00021  0000               ;****          kernal routines           ****
00022  0000               ;********************************************
00023  0000               ;
00024  0000               outchr=$ffd2           ;print char in acc
00025  0000               ;
00026  0000               ;********************************************
00027  0000               ;****      rom and system routines       ****
00028  0000               ;********************************************
00029  0000               ;
00030  0000               chrget=$0070           ;char get (in 4.0)
00031  0000               chrgot=$0076           ;char got (in 4.0)
00032  0000               synerr=$bf00           ;syntax err (in 4.0)
00033  0000               ;
00034  0000               ;********************************************
00035  0000               ;****     system storage areas           ****
00036  0000               ;********************************************
00037  0000               ;
00038  0000               txtptr=$0077           ;current char
00039  0000               ;
00040  0000               ;*******************************************
00041  0000               ;****         main routine              ****
00042  0000               ;*******************************************
00043  0000               ;
00044  0000                      *=$033a
00045  033a               ;
00046  033a               ;
00047  033a  20 70 00            jsr chrget                                  ;get 1st digit
00048  033d  20 44 03            jsr aschex                                  ;convert number
00049  0340  20 70 00            jsr chrget                                  ;get char before
00050  0343  60                  rts                                         ; return to basic
00051  0344               ;
00052  0344               ;********************************************
00053  0344               ;***  convert number from ascii to hex  ****
00054  0344               ;********************************************
00055  0344               ;
```

```
1:aschex.src......page 0002

line# loc    code         line


00056  0344               ;
00057  0344  38           aschex sec                                         ;set carry for sub
00058  0345  e9 30               sbc #48                                     ;1st digit to hex
00059  0347  8d 71 03            sta answer                                  ;store it in answer
00060  034a  20 70 00            jsr chrget                                  ;get second digit
00061  034d  90 07               bcc asc1                                    ;goto to asc1 if
00062  034f               ;                                                      btwn 0-9
00063  034f  c9 2c               cmp #44                                     ;  is it = ","
00064  0351  f0 1d               beq ascend                                  ;    yes. done
```



```
00065  0353  4c 00 bf          jmp synerr                       ;    no. syntax error
00066  0356                ;
00067  0356  38            asc1    sec                          ;set carry for sub
00068  0357  e9 30                 sbc #48                      ;2nd digit to hex
00069  0359  48                    pha                          ;save 2nd digit
00070  035a  a9 00                 lda #0                       ;
00071  035c  ae 71 03              ldx answer                   ;find tens by mult
00072  035f  18            asc2    clc                          ;
00073  0360  69 0a                 adc #10                      ; 1st x 10
00074  0362  ca                    dex                          ;
00075  0363  d0 fa                 bne asc2                     ;
00076  0365  8d 71 03              sta answer                   ;store tens in temp
00077  0368  68                    pla                          ;get 2nd digit
00078  0369  18                    clc                          ;
00079  036a  6d 71 03              adc answer                   ;  and add to temp
00080  036d  8d 71 03              sta answer                   ;  save it in temp
00081  0370                ;
00082  0370  60            ascend rts                           ;done
00083  0371                ;
00084  0371                ;****************************************
00085  0371                ;****        storage areas           ****
00086  0371                ;****************************************
00087  0371                ;
00088  0371  00            answer .byte 0                       ;hex result
00089  0372                ;
```

Lines 47–50

This series of lines is the main routine of this program. When the "sys" command is encountered in your BASIC program or from immediate mode, control is passed to the routine starting at the memory address following the "sys." In this case, the address is 826 or hex 033a. The first thing the routine does is get the next character from the BASIC line. (The format for the "sys" parameter is: sys 826,x. When the routine is first entered, the character in the Accumulator is a ",".) This routine will then load the Accumulator with the first (and maybe the last, if it is a single-digit number) digit of the number.

Line 48 will then jump to the subroutine that converts the ASCII representation of the number to hexadecimal. Note that the first digit of the number is sitting in the Accumulator. We will explain the actual routine in a minute.

Line 49 will jump to the subroutine that gets the next character in the line. At this point, the routine has already extracted the final digit (if there was a second digit) and converted the number passed by the "sys" command. Calling this subroutine will fetch the character following the "sys" command and its parameters. We do this because the BASIC interpreter will resume its operation when the return in Line 50 is executed. The character in the Accumulator must, at this point, be the one following our "sys" command (perhaps a ":").

Lines 57–59

This set of lines introduces us to three new instructions. The first instruction "sec" will "Set the Carry Flag." This is done because of the nature of the "sbc" or Subtract With Carry" instruction. The "sbc" instruction will subtract the value following it from the value in the Accumulator, placing the result in the Accumulator. If the resultant value is greater than or equal to 0, then the Carry Flag will be set. If the result is less than 0 (i.e., if the value in the Accumulator was less than the value being subtracted), the Carry Flag is cleared. Because of the nature of binary arithmetic (two's complement form), it is required that the Carry Flag be set when subtracting single precision numbers. For a more detailed explanation of the Subtract instruction, refer to the MOS Programming Manual or some other suitable 6502 reference guide.

Getting back to our subtraction, remember that the Accumulator already contains the ASCII representation of the first digit (or "tens" column digit for a 2-digit number). To find the actual numeric value of this digit, we must subtract the ASCII value of the character "0" from the ASCII value of the digit. The easiest way to do this is to use the immediate form of the Subtract instruction, which is represented by the "#". (As was mentioned before, the Assembler will assume a decimal value unless specified). For instance, the number "2" would be represented as "50" or in hex as "32". By subtracting ASCII "0" which is actually "48" or in hex, "30", you can see that the resultant value will be "2" (50 – 48 = 2). So after performing the instructions on Lines 57 and 58, the Accumulator contains the actual numeric value of the first digit.

Line 59 introduces us to the "sta" or "Store the Accumulator in Memory" instruction. This command simply stores a copy of the value in the Accumulator at a certain memory location. This case illustrates the absolute form of the instruction in that the label following is an absolute location in memory. Remember that we defined the label "answer" at the very bottom of the program, so the value will be placed at location hex 0371.

Lines 60–65

Line 60 first gets the next character from the line and stores it in the Accumulator. This character can be the second digit of the number, a comma (if it is a one-digit number), or some other BASIC token.

If the Carry Flag is not set (by routine "chrget") then we know that the character in the Accumulator is the ASCII representation of a number between 0 and 9. The "bcc" or "Branch on Carry Clear" instruction on Line 61 is another type of branch instruction that will branch to another point in the routine (in this case, "ascl", if the Carry is not set). If there is a second digit, this path will be followed.

If the character is not a number between 0 and 9, then we want to check to make sure it is a comma. Note that this check is up to you as a programmer (i.e., I have decided that one digit numbers must be followed by a comma). The "cmp" or "Compare the Accumulator with Memory" instruction on Line 63 compares the value now residing in the Accumulator with the immediate value of "44" ("44" is the ASCII representation of a comma). The "cmp" instruction actually subtracts the value in memory (in this case, the immediate value "44") from the value in the Accumulator, but DOES NOT store the result anywhere. Depending on what the result of the operation was, one of three Status Flags will be set. The Negative Flag is set if the result is less than zero, the Carry Flag is set if the result is greater than or equal to zero, and the Zero Flag is set if the result is equal to zero.

The "beq" or "Branch on Result equal to Zero" instruction in Line 64 will cause a branch to the end of the routine, "ascend" (Line 82), if the present character is a comma. If this is the case, the "sys" command only passed a one-digit number. If the character is anything else, the routine will jump to a routine that prints "syntax error". This is done on Line 65.

Lines 67–69
If the character in the Accumulator is a number between 0 and 9, indicating a second digit, then control continues at Line 67. Lines 67 and 68 will again take the ASCII representation of the second digit, and subtract an ASCII "0" from it. The resultant value in the Accumulator will then be the actual numeric representation of the second digit.

We have now converted the two digits down to their actual numeric forms. The first digit at this point is stored at the memory location labeled "answer" and the second digit is in the Accumulator. Now what? Well I now have to disturb the cobwebs of your mind by referring to some concepts that were taught in elementary school—those concerning the "tens" column and "units" column of numbers. The first digit obviously represents the "tens" column, while the second digit represents the "units" column. To convert this number, we must first multiply the "tens" column by 10, then add the "units" column to that answer.

But before going on to the instructions that do this, we first must save the second digit. I do this in Line 69 with the "pha" or "Push the Accumulator onto the Stack" instruction. What's a Stack? The easist way to explain a stack is to use the analogy of a "plate stacker" used in many restaurants. As clean dishes are brought out, they are placed in the stacker. When a dish is needed, it is taken off the top and the stacker pushes upward. The idea is that the most recently placed dishes are the first to be taken off. The Stack in the PET/CBM works the same way. The values placed on the Stack are the first to be taken off. The only thing that you must be careful of is to make sure that values are taken off in the proper order.

Lines 70–77
How do we perform a multiplication? The way I've chosen here is a technique known as the "successive addition" method. In other words, we start with 0 and add 10 to the result of the previous addition successively for the number of times specified by the "tens" column digit. For example, if our first digit is "3," we perform the operation, 0+10(1st)+10(2nd)+10(3rd), which gives us 30.

To implement this, we first perform the instruction "lda" or "Load the Accumulator with the value in Memory" as in Line 70. Again we have the immediate form of the instruction which in this case loads the value "0" into the Accumulator. Next we load the X register with the value of the "tens" column to be used as the counter for the upcoming loop.

We next enter the loop which starts at Line 72 at label "asc2." First, we set up for an addition operation by performing a "clc" or "Clear the Carry Flag" instruction as on Line 72. This instruction resets the Carry Flag. We do this because the next instruction, the "adc" or "Add Memory to the Accumulator with Carry", will use the Carry Bit (if it is set) as part of the addition. In Line 73, we add the immediate value 10 to the contents of the Accumulator (which at this point is 0). The resultant value will be stored in the Accumulator. As in other operations, certain status flags will be set according to the outcome of the operation. For more information on this, refer to MOS Programming Manual.

Line 74 will decrement the "tens" column value which we initially stored in the X Register. This is done with the "dex" or "Decrement the X Register" instruction. Again, certain status flags will be set according to the outcome of the operation.

If the resultant value in the X Register after the "dex" instruction is performed is 0, the branch instruction on Line 75 will not be executed. If the value is still greater than 0, the loop will be entered again at Line 72. You will notice that the number of successive additions is controlled by the value of the X Register or the "tens" column digit. When we fall out of the loop at Line 75, the value in the Accumulator will be the final result of the successive additions. All we need to do now is add the "units" digit to the value in the Accumulator and the resultant value will be our converted number.

Line 76 stores the "tens" column calculation at location "answer." Note that we have already stored the first digit

→

Machine Language Programming (cont.)

at this location in line 59, so this operation will overwrite that value. The reason for this is simple. If the number to convert was only a single digit, then the value in "answer" is the converted value when we branch to the end of the program in Line 64. If the number to convert is two digits, then the value in answer can be overwritten.

Remember that we originally stored the "units" digit on the Stack in Line 69. Now we need to pull it off, so we do this with the "pla" or "Pull top value off the Stack and place into the Accumulator" instruction. Remember that if you are not careful to pull values from the Stack in the order that they were placed there, you could get yourself in trouble. Notice also that the Stack is a handy place to store values.

Lines 78-82
At this point, we have the "tens" column calculation in location "answer" and the "units" column value in the Accumulator. All we have to do is add these two values together. Lines 78 and 79 do exactly that. After the calculation is performed, we store the result of the addition at storage location "answer", and finally return to the main routine at Line 82.

SUMMARY
To summarize, the program first gets the first digit from our BASIC line (Line 47). Once it has done this, it jumps to the conversion routine (Line 48). This routine first converts the first digit (Lines 57-59) and stores it at "answer." Then it gets the next character from the BASIC line (Line 60) and checks to see if it is numeric (Line 61), or if it is a comma (Line 63). If it is numeric, then it branches to the routine that gets the second digit and converts the number. If it is a comma, then we know it is only a one-digit number and so the routine branches to the end (Line 64). If it is neither of these types of characters, then "syntax error" will be printed (Line 65).

If it is a two-digit number then first we convert it to its actual numeric representation (Lines 67-68). Then we set up to perform our successive addition using the "tens" column digit as the counter (Lines 69-71), do the actual addition (Lines 72-74), add the "units" value to that result (Lines 76-79), and finally, store the converted number at location "answer" (Line 80).

The last thing we do before returning to BASIC is to get the next character (Line 49).

And that's it. Easy, right?? Have fun and experiment!

If you should have any questions or comments, please write me in care of the Editor.■

—Dave Scott

Screen Window for 8032 CBM

by
John F. Krebs

So you have a 80 Column 8032 CBM and want to make use of those fabulous screen editing and graphic features. Of particular interest is controlling the window. Here is a summary of the screen window features both in the direct keyboard mode and under program control.

There are really FOUR different methods of creating or formatting a window on the 80 column screen:

1. USING POKES. Poke location 224 for the top boundary. The screen has 25 lines so poke 224,9 for ten lines from the top. Poke 225 for the bottom boundary. Thus, poke 225,19 for a bottom 20 lines from the top. One may also poke the left window margin this way. There are 80 columns so poke 226,39 for the left margin set at the 40th column. Of course this can be done in either direct mode or in a program.

2. USING TABS AND REVERSE FIELD LETTERS IN QUOTES. The left top margin for the window is set by the reverse field letter "o" in lower case. Thus: print"(7 down cursors)"tab(19)"ESC RVS field small o" to obtain a left upper corner that is 8 lines down and 20 columns from the left. The lower right corner is obtained by using the reverse upper case or shifted O. This also can be done by direct mode (without the down cursors by positioning the statement on the line itself) or in a program. To obtain the reverse character in quotes, use the ESC or escape key to get out of the quote mode.

```
10 print "clr scrn" :rem count down from top of screen
20 print "↓↓↓↓↓↓↓" tab(19)"(ESC key)RVS o" :rem left
top corner
30 print "↓↓↓↓↓↓↓↓↓↓" tab(45)"(ESC key)RVS O"
:rem right botm corner
```

3. USING THE CHR$ NOTATION. Chr$(15) is for the top left corner and chr$(143) is for the lower right corner. Again, use the tab to do it correctly. The previous examples in method 2 will serve if one substitutes the CHR$ number for the reverse field o or O.

```
10 print "clr scrn"
20 print "↓↓↓↓↓↓↓" tab(19)chr$(15)
30 print "↓↓↓↓↓↓↓↓↓↓↓" tab(45)chr$(143)
```

4. USING A COMBINATION OF KEYS IN THE DIRECT MODE. Position your cursor on the exact spot on the screen. Then press simultaneously the lower case three letters z,a,l for the top left corner. Reposition the cursor and use the same three letters, but now upper case or shifted for the lower right corner: Z,A,L.■

REFERENCES: Compute! Nov/Dec 1980, p.93
Compute! Jul/Aug 1980, p.70
Compute! Nov/Dec 1980, p.80
Commodore Magazine Oct 1981, p.2

Specific Line Delete

by
Jack Weaver

There are many programs around for the PET and CBM which delete unwanted lines from a program. This one will act only on the lines you specify without printing number sequences to the screen which will not be used.

This program will accept your input to delete line A through line B. It then uses the function of the Keyboard buffer to store Returns (CHR$(13)), and uses them to act upon the screen after the specific lines to be deleted have been printed to the screen.

This neat little utility routine can be listed to the screen before loading a larger program (the one that needs some deletion). After loading of the subject program has stopped, simply bring the cursor to the first line of the DELETE program, and hit RETURN for the 10 lines. It will then be appended to your program.

Be sure that your subject program does not have lines that are numbered as high as this DELETE program.

To activate the DELETE program, simply RUN 63000. Then specify the lines to be deleted and you will see it list each line number to the screen. When the computer stops itself, LIST your program and you will see that the lines that you specified are gone, and very quickly. There is no time wasted in listing unwanted line numbers to the screen.■

Editors Note: This program can only delete 10 lines at a time.

The Program

Lines 63000 and 63005 accept your input for the range of line numbers to delete.

Line 63010 initializes G$, E$ and variable T.

Line 63030 clears the screen and starts a loop at memory location 1025.

Line 63040 PEEKs locations which give the next coming line number and put that figure into variable X.

Line 63050 increments variable A to the point where only nine line numbers are printed to the screen.

Line 63060 aborts the loop if it reaches Line number 63000, the first line of this routine.

Line 63070 prints Line numbers to the screen and aborts the routine if the program meets the maximum line number you have specified in line 63000.

Line 63080 increments the J loop the necessary amount to point to the next line number, by PEEKing ahead.

Line 63090 prints the value of variable B and the incremented value of variable N as well as the string G$. If the program is to be aborted because it has reached the limit set in line 63000 or if it reaches line 63000 then G$ is changed from ":GOTO 63010" to "END". Be sure that a COLON begins strings G$ and E$ in line 63010.

Line 63100 pokes the index to the Keystroke Buffer, fooling the computer into thinking there have been ten keys struck. Then the J loop POKES value 13 into memory locations 623 through 632. This means a return will be executed to the screen 10 times after the screen has been cleared and printed with new line numbers to delete.

```
63000 INPUT "ENTER STARTING LINE#";B
63005 INPUT "ENTER ENDING LINE#";N
63010 G$ = ":GOTO63010":T = 256:E$ = ":END"
63030 PRINT"▚▚▚": FORJ = 1025T032678
63040 X = PEEK(J + 3)*T + PEEK(J + 2):IFX
        < BTHEN63000
63050 A = A + 1:IFA = 10THEN63090
63060 IFX = 63000THENG$ = E$:GOTO63090
63070 PRINTX:IFX = > NTHENG$ = E$:GOTO63090
63080 J = PEEK(J + 1)*T + PEEK(J) – 1:NEXT
63090 PRINT"B = "B":N = "N;G$
63100 POKE158,10:FORJ = 0TO9:POKE623 + J,13:
        NEXT:PRINT"▚":END
READY.
```

"Two Handed Sketching"

by
Preston F. Marshall

"Two Handed Sketching" is aimed at age groups 5 to 105 years young. The urge to doodle is ageless. Your 8K or larger PET computer is a means to satisfy that urge. With this program, one can move a pen over 4,000 possible screen locations in eight different directions and with five different types of control commands plus the option to print a hard copy screen dump.

After working the program a few times, you suddenly realize you have not scratched the surface of its possibilities. Cubics, curves, figures within figures, dot drawings, faces, machines, chemical, biological, mathematical, graphical, geographic, and just abstract forms that take on meaning as you view them all crowd in upon you.

Six-year-old Jim wants to show you his space ship on the screen. Mary is ready to draw a representation of a stage layout for a play she is working on. Your turn later.

The principles of drawing the screen with two hands is shown in Figure 1. A simplified flow chart is shown in Figure 2. In my opinion, the hour spent punching in the program is well worth the effort, but if you calculate otherwise, send $5 to my address and I will arrange to have a local store forward a verified copy on one side of a 50 foot tape.

A VIC adaptation of "TWO HANDED SKETCHING" is a straight-forward conversion. With a little extra effort, the addition of color would add spice.

Preston Marshall
191 Gould Street
Walpole, MA 02081

Figure 1

LEFT HAND CONTROLS
OPERATION

S STOP

E ERASE AND BLINK

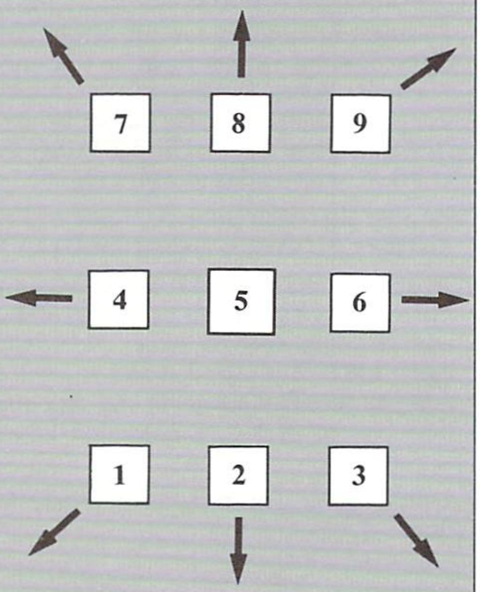
R RETURN HOME

T TRACE

H HOP OVER PRINT AND BLINK

P PRINT HARD COPY USING A SCREEN DUMP

RIGHT HAND CONTROLS
DIRECTION



The display screen is 80 columns by 50 lines.

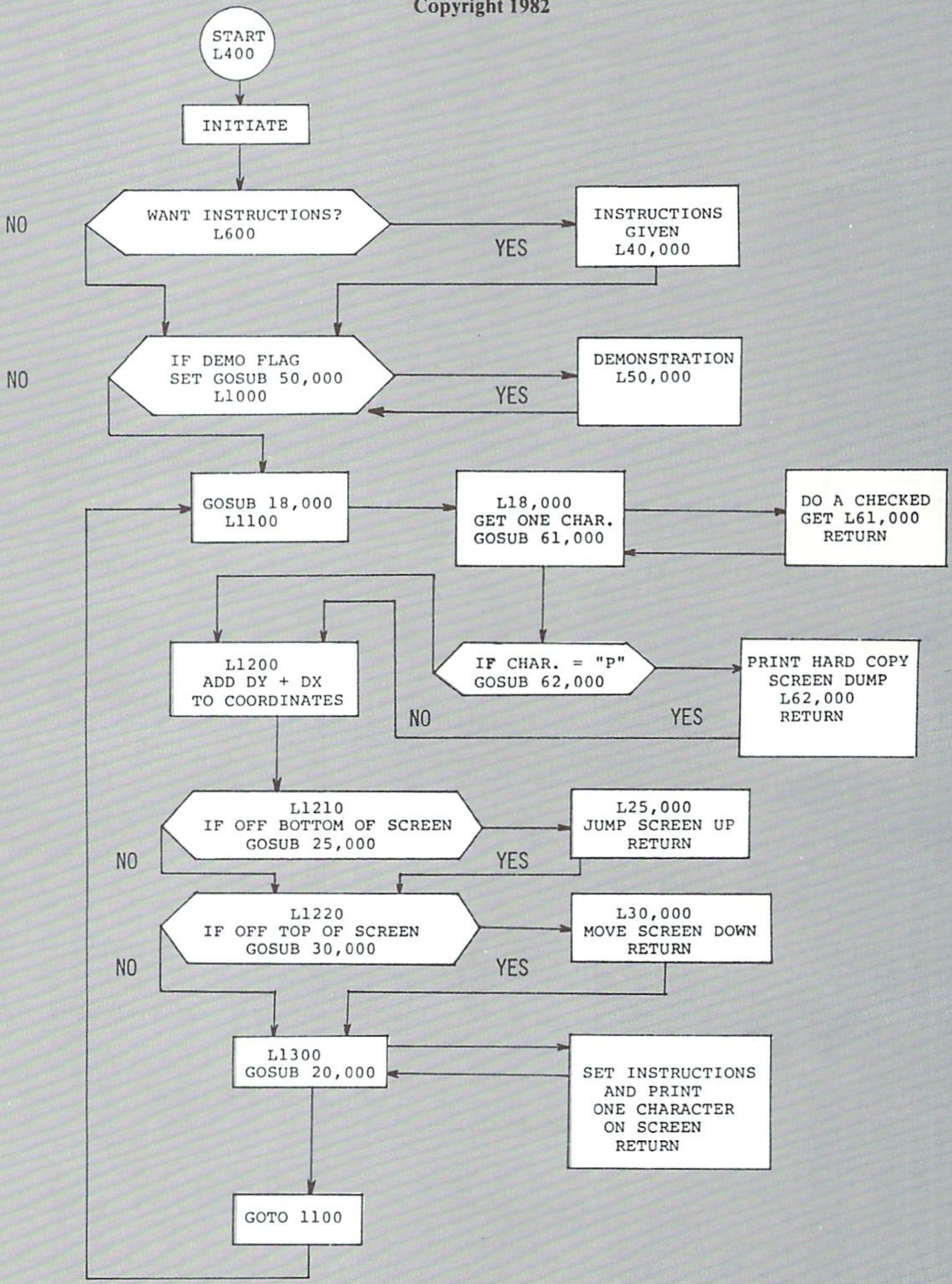
The character printed is a square white dot (4 x 4 pixels) that is 1/4 of the area of the standard characters.

When the right hand directs the dot to move, it will continue moving until the left hand gives the order STOP (S) or the right hand issues a new direction order.

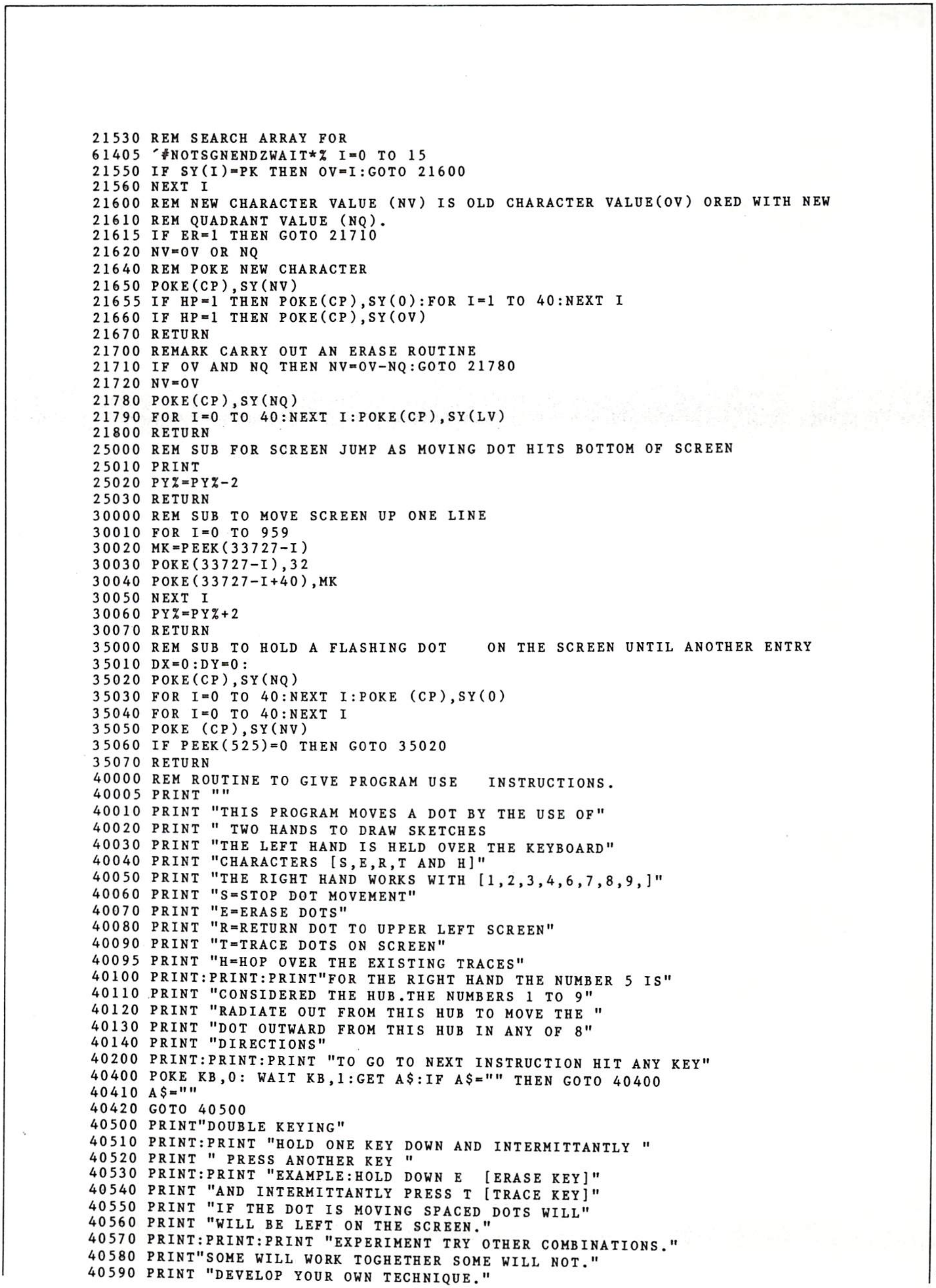
Lines 50005–50015 can be altered so that the demonstration program automatically draws a sketch you have designed without use of the control keys.

Figure 2
Two Handed Sketching

Preston F. Marshall
191 Gould Street
Walpole, MA 02081



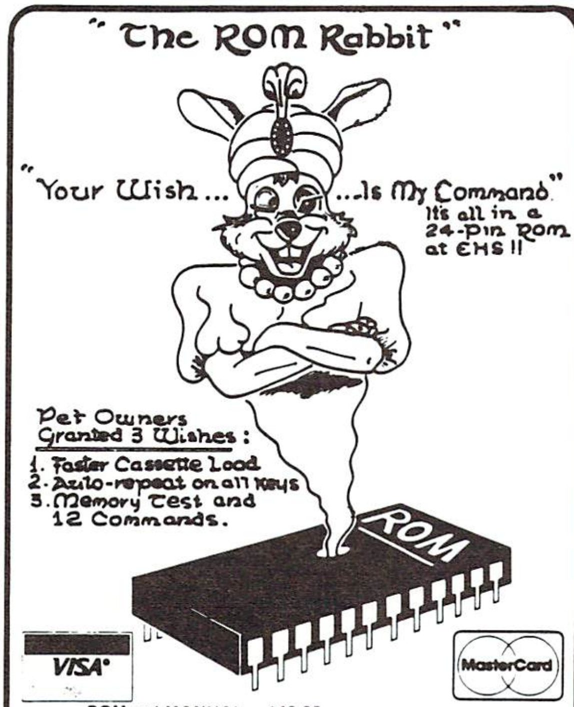
```
400 KB=158:REM KEYBOARD COUNTER NEW ROMS
410 IF PEEK(51234)=0 THEN KB=525:REM OLD ROMS
420 DATA 64,0,128,64,-64,-128,0,-64
430 FOR I=0 TO 7:READ C(I):NEXT I:REM  SETS UP DATA FOR PRINT OUT
500 REM**TWO HANDED SKETCHING******
510 REM**BY PRES MARSHALL**********
511 REM**191 GOULD ST,WALPOLE,MA***
512 REM**02081*********************
513 REM*COPYRIGHT 1982*************
520 REM****************************
600 PRINT"DO YOU WANT INSTRUCTIONS?"
610 PRINT:PRINT "HIT [Y] FOR YES"
620 PRINT:PRINT "HIT [N] FOR NO"
630 POKE KB,0:WAIT KB,1:GETA$:IF A$="" THEN 630
640 IF A$="Y" THEN A$="":DE=1:GOTO 40000
650 IF A$="N" THEN A$="":   GOTO 700
670 A$="":PRINT "TRY AGAIN":GOTO 630
700 DIM SY(15):REM SYMBOL ARRAY
705 DIM DR$(1)
710 PRINT""
720 PRINT""
800 POKE 59468,12
810 DATA 32,123,126,97,108,98,127,252,124,255,226,236,225,254,251,160
820 FOR I=0 TO 15
830 READ SY(I):REM SET UP SYMBOL ARRAY
840 NEXT I
900 PY%=0:PX%=0
1000 REM START OF PROGRAM
1010 IF DE=1 THEN GOTO 50000:REM IF DEMO FLAG IS SET GO TO 50000
1100 GOSUB 18000
1200 PX%=PX%+DX:PY%=PY%+DY
1210 IF PY%>49 THEN GOSUB 25000
1220 IF PY%<0 THEN GOSUB 30000
1300 GOSUB 20000
1400 GOTO 1100
18000 REM SET UP A SET OF DIRECTIONS TO MOVE THE TARGET DOT WITH A GET
18100 VA$="64283719RSETHP":REM VA=VALID RESPONSES TO A SINGLE LETTER  GET
18110 IF PEEK(KB)=0 THEN GOTO 18240:REMIF NOTHING IN KEYBOARD BUFFER GO ON
18120 GOSUB 61000:REM DO A CHECKED GET
18130 IF C$="6"THEN DX=1:DY=0:REMRIGHT
18140 IF C$="4"THEN DX=-1:DY=0:REMLEFT
18150 IFC$="2"THEN DX=0:DY=1:REMDOWN
18160 IFC$="8" THEN DY=-1:DX=0:REM UP
18170 IFC$="3" THEN DX=1:DY=1:REM SE
18180 IFC$="7" THEN DX=-1:DY=-1:REM NW
18190 IFC$="1" THEN DX=-1:DY=1:REMSW
18200 IFC$="9" THEN DX=1:DY=-1:REM NE
18210 IF C$="T" THEN ER=0:HP=0:REM SETS DOT TO LEAVE ITS TRACE
18220 IF C$="R" THEN PX%=0:PY%=0:DY=0:DX=0:REM SET DOT TO ZERO POSITION
18230 IF C$="E" THEN ER=1:HP=0:REM SETS DOT TO ERASE AS IT MOVES
18240 IF C$="S" THEN GOTO 35000:REM GO TO A FLASHIND DOT SUB
18250 IFC$="H" THEN HP=1:ER=0: REMARK SETS FLAG TO HOP OVER EXISTING DOTS
18260 IF C$="P" THEN GOSUB 62000
18900 RETURN
20000 REM START OF ARRAY TO SET QUADRANT SYMBOLS
21000 REM SUB TO PRINT PX% AND PY% ON A 50 BY 80 COORDINATE GRAPH
21050 RO=2:REM RO= ROW OF QUADRANT SYMBOL
21060 IF PY%/2=INT(PY%/2) THEN RO=1
21100 CO=2:REM CO=COLUMN OF QUADRANT SYMBOL
21110 IF PX%/2=INT(PX%/2) THEN CO=1
21200 REM CHARACTER POSITION ON SCREEN=CP
21210 CP=32768+40*(INT(PY%/2))+INT(PX%/2)
21300 REM SET UP NEW QUADRANT LOCATIONS NQ
21310 IF RO=1 AND CO=1 THEN NQ=2
21320 IF RO=1 AND CO=2 THEN NQ=8
21330 IF RO=2 AND CO=1 THEN NQ=1
21340 IF RO=2 AND CO=2 THEN NQ=4
21350 REM IN ABOVE QUADRANTS READ 1,2,4,8
21500 REM PEEK TO FIND OLD CHARACTER VALUE AND THEN ADD NEW QUADRANT VALUE
21510 PK=PEEK(CP):REM PK =PEEKED OLD SYMBOL. OV= OLD ARRAY VALUE
```



```
21530 REM SEARCH ARRAY FOR
61405 ´#NOTSGNENDZWAIT*% I=0 TO 15
21550 IF SY(I)=PK THEN OV=I:GOTO 21600
21560 NEXT I
21600 REM NEW CHARACTER VALUE (NV) IS OLD CHARACTER VALUE(OV) ORED WITH NEW
21610 REM QUADRANT VALUE (NQ).
21615 IF ER=1 THEN GOTO 21710
21620 NV=OV OR NQ
21640 REM POKE NEW CHARACTER
21650 POKE(CP),SY(NV)
21655 IF HP=1 THEN POKE(CP),SY(0):FOR I=1 TO 40:NEXT I
21660 IF HP=1 THEN POKE(CP),SY(OV)
21670 RETURN
21700 REMARK CARRY OUT AN ERASE ROUTINE
21710 IF OV AND NQ THEN NV=OV-NQ:GOTO 21780
21720 NV=OV
21780 POKE(CP),SY(NQ)
21790 FOR I=0 TO 40:NEXT I:POKE(CP),SY(LV)
21800 RETURN
25000 REM SUB FOR SCREEN JUMP AS MOVING DOT HITS BOTTOM OF SCREEN
25010 PRINT
25020 PY%=PY%-2
25030 RETURN
30000 REM SUB TO MOVE SCREEN UP ONE LINE
30010 FOR I=0 TO 959
30020 MK=PEEK(33727-I)
30030 POKE(33727-I),32
30040 POKE(33727-I+40),MK
30050 NEXT I
30060 PY%=PY%+2
30070 RETURN
35000 REM SUB TO HOLD A FLASHING DOT    ON THE SCREEN UNTIL ANOTHER ENTRY
35010 DX=0:DY=0:
35020 POKE(CP),SY(NQ)
35030 FOR I=0 TO 40:NEXT I:POKE (CP),SY(0)
35040 FOR I=0 TO 40:NEXT I
35050 POKE (CP),SY(NV)
35060 IF PEEK(525)=0 THEN GOTO 35020
35070 RETURN
40000 REM ROUTINE TO GIVE PROGRAM USE   INSTRUCTIONS.
40005 PRINT ""
40010 PRINT "THIS PROGRAM MOVES A DOT BY THE USE OF"
40020 PRINT " TWO HANDS TO DRAW SKETCHES
40030 PRINT "THE LEFT HAND IS HELD OVER THE KEYBOARD"
40040 PRINT "CHARACTERS [S,E,R,T AND H]"
40050 PRINT "THE RIGHT HAND WORKS WITH [1,2,3,4,6,7,8,9,]"
40060 PRINT "S=STOP DOT MOVEMENT"
40070 PRINT "E=ERASE DOTS"
40080 PRINT "R=RETURN DOT TO UPPER LEFT SCREEN"
40090 PRINT "T=TRACE DOTS ON SCREEN"
40095 PRINT "H=HOP OVER THE EXISTING TRACES"
40100 PRINT:PRINT:PRINT"FOR THE RIGHT HAND THE NUMBER 5 IS"
40110 PRINT "CONSIDERED THE HUB.THE NUMBERS 1 TO 9"
40120 PRINT "RADIATE OUT FROM THIS HUB TO MOVE THE "
40130 PRINT "DOT OUTWARD FROM THIS HUB IN ANY OF 8"
40140 PRINT "DIRECTIONS"
40200 PRINT:PRINT:PRINT "TO GO TO NEXT INSTRUCTION HIT ANY KEY"
40400 POKE KB,0: WAIT KB,1:GET A$:IF A$="" THEN GOTO 40400
40410 A$=""
40420 GOTO 40500
40500 PRINT"DOUBLE KEYING"
40510 PRINT:PRINT "HOLD ONE KEY DOWN AND INTERMITTANTLY "
40520 PRINT " PRESS ANOTHER KEY "
40530 PRINT:PRINT "EXAMPLE:HOLD DOWN E  [ERASE KEY]"
40540 PRINT "AND INTERMITTANTLY PRESS T [TRACE KEY]"
40550 PRINT "IF THE DOT IS MOVING SPACED DOTS WILL"
40560 PRINT "WILL BE LEFT ON THE SCREEN."
40570 PRINT:PRINT:PRINT "EXPERIMENT TRY OTHER COMBINATIONS."
40580 PRINT"SOME WILL WORK TOGHETHER SOME WILL NOT."
40590 PRINT "DEVELOP YOUR OWN TECHNIQUE."
```

→

```
40595 PRINT:PRINT:PRINT "HIT P FOR HARD COPY"
40600 PRINT:PRINT:PRINT "TO START HIT ANY KEY
40900 POKE KB,0:WAITKB,1:GETA$:IF A$=""THEN GOTO40900
40910 A$="":GOTO 700
50000 REM SUB TO RUN DEMONSTRATION PROGRAMS
50005 DR$="R333333333333666999888777444111222333333366998877444112233666666"
50010 DR$=DR$+"RE22222222222222T333222222222688888889963222222336669988912226"
50015 DR$=DR$+"6669988844426666999889999111122222233669E888877T777"
50020 PRINT "  DEMO      LAST KEY PRESSED="
50030 FOR K=1 TO LEN(DR$)
50040 C$=MID$(DR$,K,1)
50050 DD=ASC(C$)
50060 IF DD>64 THEN DD=DD-64
50070 POKE(32798),DD
50080 GOSUB 18130:REMARK PLOT
50082 PX%=PX%+DX:PY%=PY%+DY
50085 GOSUB 20000:REM PLOT
50090 NEXT K
50100 PRINT"TO GO ON HIT ANY KEY"
50105 POKE(KB),0:WAIT(KB),1:GET A$:IF A$="" THEN GOTO 50105
50120 PRINT "WANT TO SEE IT AGAIN?"
50130 PRINT "HIT [Y] FOR YES OR [N] FOR NO"
50140 POKE(KB),0:WAIT(KB),1:GET A$:IF A$="" THEN GOTO 50140
50150 IF A$="Y" THEN DE=1:REM SET DEMONSTRATOR FLAG TO 1
50160 IF A$="N" THEN DE=0:REM RESET DEMO FLAG
50170 IF A$<>"Y" AND A$<>"N" THEN PRINT"TRY AGAIN":GOTO 50130
50180 PY%=0:PX%=0
50190 PRINT ""
50210 DY=0:DX=0
50220 GOTO 1000
61000 REM SUB TO DO A CHECKED GET
61010 T1=TI
61020 CC$=C$
61030 GETC$:IFC$="" AND (TI-T1)<10 THEN 61030
61040 IF C$<>"" THEN GOTO 61060
61050 IF C$="" AND (TI-T1)>10 THEN C$=CC$:RETURN
61060 FOR G=1 TO LEN(VA$)
61070 IF C$=MID$(VA$,G,1) THEN RETURN
61080 NEXT G: GOTO 61030
62000 REM AXIOM SCREEN DUMP
62010 OPEN 4,4:CMD4:PRINT CHR$(8)+CHR$(15);
62020 R$(0)=CHR$(146)
62030 R$(1)=CHR$(18)
62040 FOR I= 32768 TO 33767
62050 P=PEEK(I)
62060 PRINT R$(INT(P/128)) +CHR$(P+C(P/32));
62070 NEXT I
62075 C$="R"
62080 PRINT#4,"":CLOSE4:RETURN
```



ROM and MANUAL — $49.95
Specify 3.0 or 4.0 Basic (Works with or without toolkit)

EASTERN HOUSE SOFTWARE
3239 Linda Drive
Winston-Salem, N. C. 27106

(919) 924-2889
(919) 748-8446

ROM OR ON CASSETTE

THE PET RABBIT

4.0 OR 3.0 ROMS

The PET Rabbit is a programmers aid which provides 12 additional commands that can be executed in BASIC's direct mode. In addition to the commands, automatic repeat of any key held down for 0.5 seconds is also provided. This will greatly aid inputtings of characters and provide more convenient cursor control. Most importantly, the RABBITs high speed recording technique allows an 8K program to be saved in 38 seconds instead of the normal 2 minutes and 44 seconds in Commodore's format. (Note— The RABBIT cannot be used to store data tapes from BASIC.)

The PET Rabbit is 2K of machine code supplied on cassette or in ROM. The cassette version occupies the top-most portion of memory and can be ordered in one of 5 locations: $1800-$1FFF for 8K PETs, $3000-$37FF or $3800-$3FFF for 16K PETs, and $7000-$77FF or $7800-$7FFF for 32K PETs. The reason for two different versions for the 16K and 32K PETs is to provide room for those programmers who use the DOS Support (wedge) program. (Note— The cassette RABBIT works only with 3.0 ROM PET's.)

The ROM version is a 24 pin Integrated Circuit which plugs into spare socket D4 and occupies memory $A000-$A7FF. Since the ROM version does not occupy user RAM, it will work with any 8K, 16K, or 32 K 3.0 or 4.0 ROM PET. The main advantage of the ROM Rabbit is that it doesn't have to be loaded each time you power up your PET and it does not occupy valuable RAM memory (4.0 ROM version at $9000).

The PET RABBITs high-speed cassette recording feature will not work with some of Commodore's older cassette decks. To be specific, cassette decks with the lift top lid (termed old style) will not work but all other features will work. In addition, we have discovered that some new style cassette decks will not work properly. How do you know if your cassette will work? Simple -open up the cassette deck and look at the printed circuit board components. If there are IC packages for all the active components, it will work with the RABBIT. If there are any transistors on the board, it will not work. Most new style cassette decks will work okay since there are very few of the transistor types. If you wish to purchase ROM RABBIT and a cassette deck, we can offer an attractive discount.

The RABBIT commands are:

SS — Save with short leader	D — Convert decimal # to hex #
SL — Save with long leader	H — Convert hex # to decimal #
L — Load a program	Z — Toggle character set
V — Verify a program	K — Kill the RABBIT
E — Load and then run	* — Go to monitor
T — RAM memory test	G — go to machine language program

RABBIT works with the PROGRAMMERS TOOL KIT.

CASSETTE RABBIT - $29.95
ROM RABBIT - $49.95
(specify memory, 3.0 or 4.0)

EASTERN HOUSE SOFTWARE
3239 Linda Dr.
Winston-Salem, N. C. 27106

U.S. Bank Check or International Money Order in U.S. dollars.

(LABEL), Y (LABEL,X) LABEL + INDX—1

LDA STA LDX STX LDY STY

SET IFM IFP IFN IFE

6502 Assembler/Editor

- APPLE
- ATARI
- PET
- KIM
- SYM

Before you buy that off-brand Assembler/Text Editor, note that EHS is the only company that provides a line of **compatible** ASM/TED's for the **PET/APPLE/ATARI/SYM/KIM** and other microcomputers.

When you make the transition from one of these 6502-based microcomputers to another, you no longer have to relearn peculiar Syntax's, pseudo ops, and commands. Not only that, EHS ASM/TED's are the **only resident 6502 Macro Assemblers** availiable and they have been available for several years. Thus you can be sure **they work**. Our ASM/TED's may cost a little more but do the others provide these **powerful features:** Macros, Conditional Assembly, String Search and Replace, or even up to 31 characters per label? Before you spend your money on that other ASM/TED, **write for our free detailed spec sheet.**

MACRO ASM/TED

- For APPLE/ATARI/PET/SYM/KIM
- Other than our MAE, no other assembler is as powerful.
- Macros/Conditional Assembly.
- Extensive text editing features
- Long Labels
- Designed for Cassette-based systems.

$49.95

MAE ASM/TED

- For APPLE/ATARI/PET
- The most powerful ASM/TED
- Macros/Conditional and Interactive Assembly
- Extensive text editing features
- Long Labels
- Control files
- Designed for Disk-based Systems.

$169.95

EASTERN HOUSE SOFTWARE
3239 Linda Drive
Winston-Salem, N. C. 27106 USA
(Dealer Inquiries Invited)

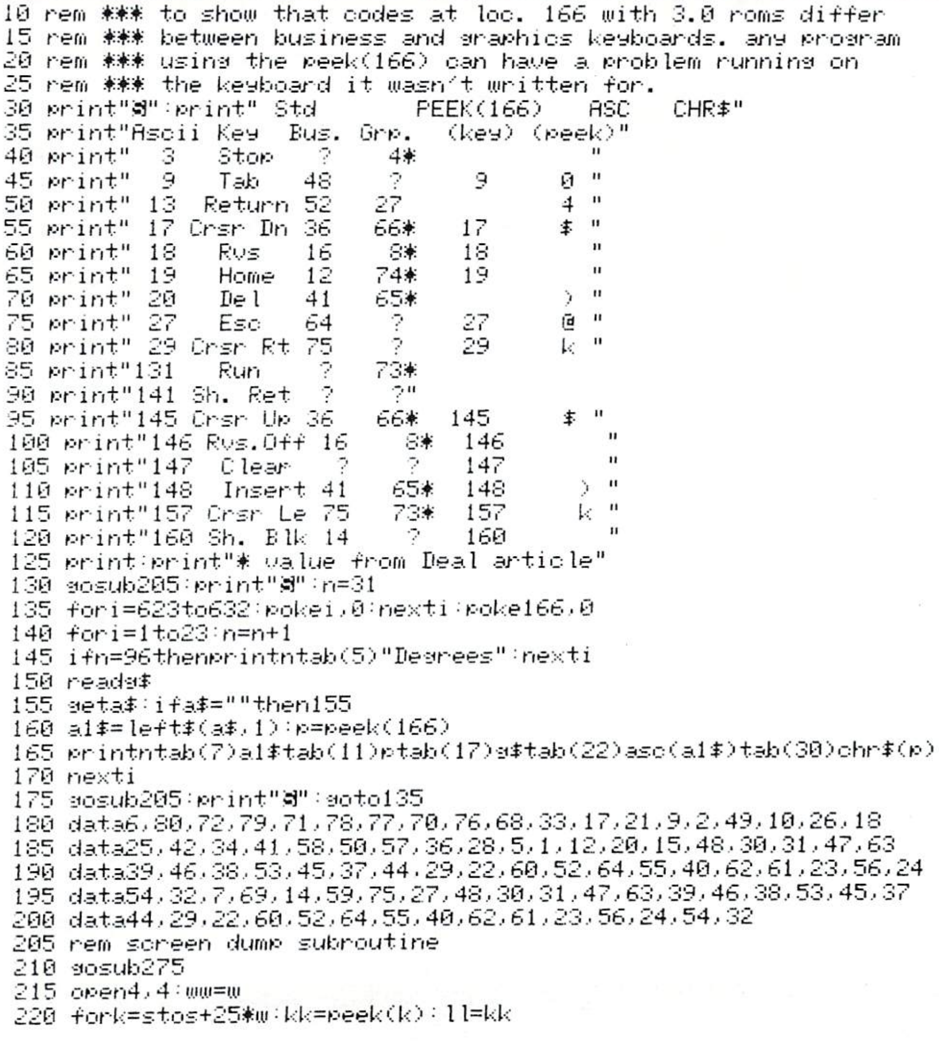
PHONE ORDERS
(919) 924-2889
(919) 748-8446

.EN .BY .OS .BA .DE .CE

Sequel to "Four PET/CBM Keyboards"

by
Louise Schultz
Havertown, PA

The October 1981 issue of Commodore Magazine carried an article by Elizabeth Deal regarding the differences in keyboard decoding among several variants of the Commodore micros. There is often a problem using a program written on a micro with a BASIC ROM that is different from the micro on which one is trying to run the program. The keyboard decoding differences not covered in the October article are found in the "Business" keyboard. Through the courtesy of Dan Herman of the PET Users Group of the Philadelphia Area Computer Society, the comparison of values at loc. 166 (and 151) in the 2001 (or 3032) "graphics" and "business" keyboards was developed. The review shows that the values in the column "Grp." correspond with those listed in the article by Liz Deal. The PEEK(151) is the same regardless of shift key position for the upper row on the business keyboard. Values shown for digits are derived from the numeric keypad. Coding for the listing loaded the "graphics" keyboard values from data lines and is not important. As suggested by Deal, the values from the keyboard in question are displayed by printing PEEK(151) or (166). This listing is organized by the conventional ASCII value, also created by the ASC ("key") on the "Business" keyboard, except for the switch of upper and lower case alphas. The column labeled "CHR$(peek)" points out the impact of the situation.■



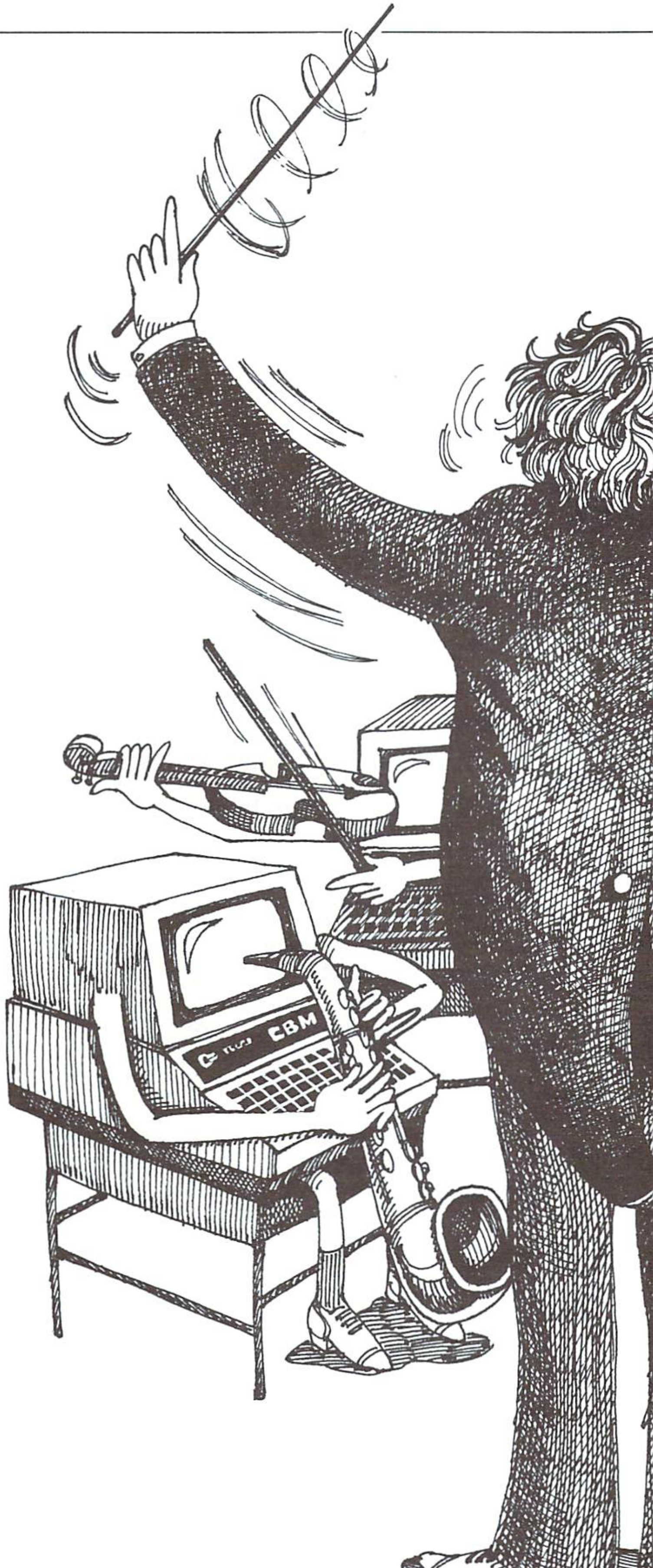
```
10 rem *** to show that codes at loc. 166 with 3.0 roms differ
15 rem *** between business and graphics keyboards. any program
20 rem *** using the peek(166) can have a problem running on
25 rem *** the keyboard it wasn't written for.
30 print"♥":print" Std        PEEK(166)   ASC   CHR$"
35 print"Ascii Key   Bus. Grp.  (key) (peek)"
40 print"  3   Stop   ?    4*             "
45 print"  9   Tab   48    ?     9      0 "
50 print" 13  Return 52   27            4 "
55 print" 17 Crsr Dn 36   66*   17      $ "
60 print" 18   Rvs   16    8*   18        "
65 print" 19   Home  12   74*   19        "
70 print" 20   Del   41   65*           ) "
75 print" 27   Esc   64    ?    27      @ "
80 print" 29 Crsr Rt 75    ?    29      k "
85 print"131   Run    ?   73*
90 print"141 Sh. Ret  ?    ?"
95 print"145 Crsr Up 36   66*  145      $ "
100 print"146 Rvs.Off 16    8*  146        "
105 print"147  Clear   ?    ?   147        "
110 print"148  Insert 41   65*  148      ) "
115 print"157 Crsr Le 75   73*  157      k "
120 print"160 Sh. Blk 14    ?   160        "
125 print:print"* value from Deal article"
130 gosub205:print"♥":n=31
135 fori=623to632:pokei,0:nexti:poke166,0
140 fori=1to23:n=n+1
145 ifn=96thenprintntab(5)"Degrees":nexti
150 readg$
155 geta$:ifa$=""then155
160 a1$=left$(a$,1):p=peek(166)
165 printntab(7)a1$tab(11)ptab(17)g$tab(22)asc(a1$)tab(30)chr$(p)
170 nexti
175 gosub205:print"♥":goto135
180 data6,80,72,79,71,78,77,70,76,68,33,17,21,9,2,49,10,26,18
185 data25,42,34,41,58,50,57,36,28,5,1,12,20,15,48,30,31,47,63
190 data39,46,38,53,45,37,44,29,22,60,52,64,55,40,62,61,23,56,24
195 data54,32,7,69,14,59,75,27,48,30,31,47,63,39,46,38,53,45,37
200 data44,29,22,60,52,64,55,40,62,61,23,56,24,54,32
205 rem screen dump subroutine
210 gosub275
215 open4,4:ww=w
220 fork=stos+25*w:kk=peek(k):ll=kk
```

```
225 ifww=wthenww=0:print#4:ifpeek(59468)=14thenprint#4,"█";
230 pokek,(ll+128)and255
235 ifkk>127andzz=0thenprint#4,"█";:zz=1
240 ifkk<128andzz=1thenprint#4,"█";:zz=0
245 kk=kkand127:ff=0:ifkk>63thenff=128:kk=kk-64
250 ifkk<32thenkk=kk+64
255 print#4,chr$(kk+ff);
260 ifkk=34thenprint#4,chr$(141);tab(ww+1);
265 ww=ww+1:pokek,ll
270 nextk:print#4:ff=45:close4:return
275 w=40:zz=0:ww=40:s=32768:return
READY.
```

Std		PEEK(166)		ASC	CHR$
Ascii	Key	Bus.	Grp.	(key)	(peek)
3	Stop	?	4*		
9	Tab	48	?	9	0
13	Return	52	27		4
17	Crsr Dn	36	66*	17	$
18	Rvs	16	8*	18	
19	Home	12	74*	19	
20	Del	41	65*		)
27	Esc	64	?	27	@
29	Crsr Rt	75	?	29	k
131	Run	?	73*		
141	Sh. Ret	?	?		
145	Crsr Up	36	66*	145	$
146	Rvs.Off	16	8*	146	
147	Clear	?	?	147	
148	Insert	41	65*	148	)
157	Crsr Le	75	73*	157	k
160	Sh. Blk	14	?	160	

* value from Deal article

Std		PEEK(166)		ASC	CHR$
Ascii	Key	Bus.	Grp.	(key)	(peek)
32		14	6	32	
33	!	72	80	33	h
34	"	80	72	34	P
35	#	7	79	35	
36	$	71	71	36	g
37	%	79	78	37	o
38	&	6	77	38	
39	'	70	70	39	f
40	(	78	76	40	n
41	)	5	68	41	
42	*	3	33	42	
43	+	58	17	43	:
44	,	21	21	44	
45	-	77	9	45	m
46	.	29	2	46	
47	/	10	49	47	
48	0	20	10	48	
49	1	9	26	49	
50	2	17	18	50	
51	3	25	25	51	
52	4	33	42	52	!
53	5	57	34	53	9
54	6	49	41	54	1
55	7	68	58	55	d
56	8	76	50	56	l
57	9	65	57	57	a
58	:	3	36	58	
59	;	58	28	59	:
60	<	21	5	60	
61	=	77	1	61	m
62	>	29	12	62	
63	?	10	20	63	
64	@	50	15	64	2
65	A	56	48	193	8
66	B	30	30	194	
67	C	31	31	195	
68	D	55	47	196	7
69	E	39	63	197	'
70	F	62	39	198	>
71	G	54	46	199	6
72	H	61	38	200	=
73	I	43	53	201	+
74	J	53	45	202	5
75	K	59	37	203	;
76	L	51	44	204	3
77	M	13	29	205	
78	N	22	22	206	
79	O	35	60	207	#
80	P	42	52	208	*
81	Q	40	64	209	(
82	R	46	55	210	.
83	S	63	40	211	?
84	T	38	62	212	&
85	U	37	61	213	%
86	V	23	23	214	
87	W	47	56	215	/
88	X	15	24	216	
89	Y	45	54	217	-
90	Z	24	32	218	
91	[	34	7	91	"
92	\	44	69	92	,
93	]	60	14	93	<
94	↑	67	59	94	o
95	←	8	75	95	
96	Degrees				
97	a	56	27	65	8
98	b	30	48	66	
99	c	31	30	67	
100	d	55	31	68	7
101	e	39	47	69	'
102	f	62	63	70	>
103	g	54	39	71	6
104	h	61	46	72	=
105	i	43	38	73	+
106	j	53	53	74	5
107	k	59	45	75	;
108	l	51	37	76	3
109	m	13	44	77	
110	n	22	29	78	
111	o	35	22	79	#
112	p	42	60	80	*
113	q	40	52	81	(
114	r	46	64	82	.
115	s	63	55	83	?
116	t	38	40	84	&
117	u	37	62	85	%
118	v	23	61	86	
119	w	47	23	87	/
120	x	15	56	88	
121	y	45	24	89	-
122	z	24	54	90	



Song Data for Instrument Synthesis Program

by
Elizabeth Deal

Probably the best way to get an idea of the information your PET needs to play music is to look at a coded song. The illustration attached to this article does just that.

Users who already have the system, can, of course, examine any song they have, in a similar fashion. They can also experiment with the parameters of instructions, specifically the F6 and F1 commands, to see what they hear. There may be a discrepancy in the starting address of the song data. I have a program with debugging extensions; my song data begins at $0F00. The older version begins at $0E00. There is no way to describe the available commands as precisely as it has been done in a book. If you have never looked at a coded song, I'd recommend you do it with the manual on hand.

I have placed signposts on the code to help you see what commands are used and what they do. All commands are of the $Ex and $Fx type. You can see them in the first two sections of the code. In the third section, the flagged bytes mark the beginning of a musical "event" which loosely corresponds to a note. Note that these are excerpts. They do show several notes, as well as some of the structure of the song, but there is no continuity in the listing.

The music program was written in machine code by Dr. Frank Covitz. The program interprets your instructions (mostly Ex and Fx commands followed by their parameters). It issues an error message if they don't make sense, or goes on to create instruments and play a song, if they do.

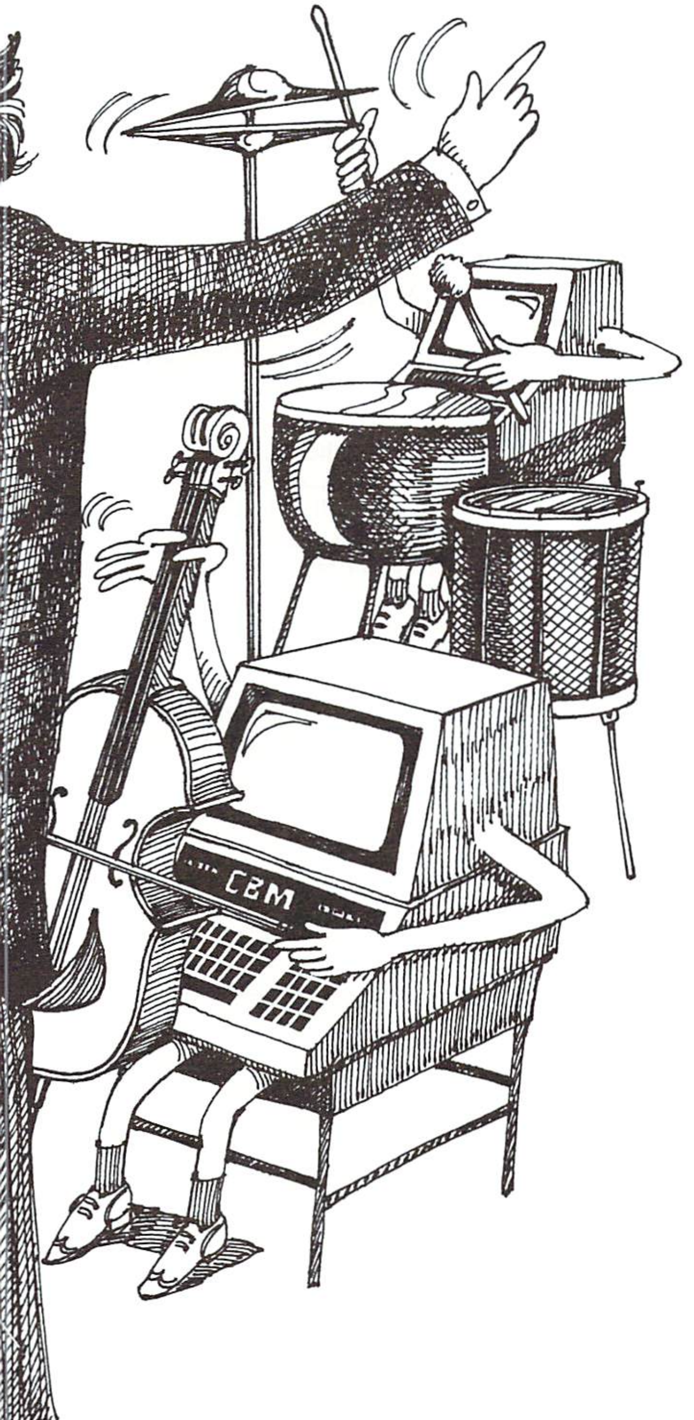
Just as in BASIC, precise syntax has to be observed. The syntax is described in the book. It consists mostly of how many parameters a command is to contain and what information is to be put into those parameters.

The code the user needs to enter consists of three major groups of communication:

1. decide what sort of sounds you would like to hear
2. build instruments and decide on how to play the song
3. provide the score to be performed by the PET.

The score itself is the simplest thing to enter. Music, its notes and durations are transcribed into numbers. That's section 3 in the illustration. Coding is easy, once you

Continued on page 78.



Real Sound in Real Time

by
Elizabeth Deal

If you ever wanted to synthesize instruments in real time on your PET/CBM, then this article will tell you of one marvelous, unadvertised, unpromoted program. It is Instrument Synthesis Software Package (K-1002-6C) from Micro Technology Unlimited. The cost of the program is $50. The required digital-to-analog converter (DAC) is $60.

Hal Chamberlin who is considered the father of real-time microcomputer music, has written several eloquent papers on the principles behind the system (see references). He invented an ingenious software method of producing multi-voice sound on the PET and has recently elaborated that invention to include instrument building.

Hence, with minimal cost, a PET user can have an inhouse synthesizer comparable in quality to some expensive hardware synthesizers. I recently attended a concert in Philadelphia at which Frank Covitz and Cliff Ashcraft, known in the trade as the Diatonic Duo, demonstrated the MTU's system with help of PET's distinguished 6502 ancestors: the Baroque Aim and the Homebrew Kim. The excellent sounds included a super 17th century harpsichord and a $2 ukelele pitch pipe.

Tiny Bit of History

Originally, the Chamberlin-style four-voice music software produced organ-like sound of varied characteristics, achieved through user's complete control of overtones. The sound had a sudden onset, level sustain and a sudden drop. There exist two such systems: AB Computers' Visible Music Monitor, written by Frank Levinson and known for its superb graphic editor, and MTU's four-voice package written by Frank Covitz, known for its lack of an editor. Both systems are currently available.

The State of the Art

Hal Chamberlin elaborated on the idea of his original system by adding an instrument synthesis feature. It is now possible to synthesize just about any sound, real or imagined. Fairly successful imitation of existing instruments has been done. Invention of new instruments via sound analysis permits us to hear, for example, Bach's inventions played by a word "NO" or "Raindrops Keep Falling on my Head", played, quite appropriately, by a straw and, I think, water dripping into a glass.

Continued on page 80.

Song Data for Instrument Synthesis Program (cont.)

enter a measure or two of music, it's just tedious. Mistakes are costly, as room may need to be opened up or closed. Supermon or Extramon are handy; the "human interface," in the form of an editor, will be very helpful, indeed, when it's written.

The second major group is fun to write and is easy to do. That's part 1 in the illustration. It's the key to the whole system. It creates a base of numerous waveforms for each harmonic. You can copy the F5 instructions from the book, from songs you already have or you can invent your own by transferring coordinates of your graph into the F5 command. Needless to say, if you have a way to analyze sounds, you can synthesize them back by entering a larger amount of data into the F5 parameters than shown here. That's apparently what Frank Covitz does when he synthesizes real sounds (wolf, oink-oink, bang!, no, listen . . . dripping water, flutes, trombones and a genuine (?) 17th century harpsichord).

Finally, section 2 completes the requirements for a coded song. It consists of two parts.

The first part builds sound configurations (loosely called "instruments") from the waves calculated in section 1. This is where most of the fun is and most experiments can be performed. There is no limit to a variety of sounds you can create from a finite set of waves. You do it by varying the last three bytes of the F6 command, with the largest and most interesting impact achieved by variations in the third byte from the end. This byte controls the variable speed at which waves are scanned during sounding of a note. F6 command used after the F5 command is the powerhouse of the program. This is where you achieve the distinct sounds of the instruments, or families of instruments, you plan to use. It's really worthwhile to spend some time fooling around with the F6 command in the already coded song and listening to results. I can't think of a better way to learn the system.

The second part tells the interpreter the structure of the song. It includes addresses of the pieces, the order in which they are to be played, etc. as well as how to play a segment: its tempo, pitch offsets from the coded values, instrument assignments to voices, and stereo voice assignments (if you have two DACs). This information can be given once and will then be used for the entire song. Alternately, it can be given for each piece or a group of song segments. It's your ballgame. You divide the song into meaningful segments based on the musical needs you detect (sound characteristics, timing changes, special dynamic changes, etc.) and you tell the PET how to play each segment.

```
C:*
     PC   IRQ   SR AC XR YR SP
.;   C6C3 9053 4D 00 3A 00 FA
.
.:   0F00 EE 10 F5 01 11.01.00 00. —
.:   0F08 02 D8.10 00.FF.02.00 00.
.:   0F10 04 60.08 60.10 00.FF.00.
.:   0F18 F5 12 31.01.00 84.1F 00.
.:   0F20 FF.03.00 44.14 00.FF.04.
.:   0F28 00 38.10 00.FF.05.00 28.
.:   0F30 12 00.FF.07.00 20.10 00.
.:   0F38 FF.08.00 08.04 00.FF.00.
.:   0F40 F0 04 F6 01.1A.12 31 F6  —
.:   0F48 02.00.01 11 F6 73.11.01
.:   0F50 0C.00.0D 11 F6 14.00.01
.:   0F58 02,00.07 11 F1 C1.21 11
.:   0F60 F1 C2.41 11 F1 C3.31 11
.:   0F68 EA.EA.EA.EA.EA.EA F2 B8
.:   0F70 FE 01 06 FE 01 BB FE 01
```

Meaning of the Code	Purpose
EE manages memory. (2)F5 calculates $31 pages of waveforms from which various instruments can be built.	1 Build base for sounds
F0 sets 04 voices. (4)F6 build 4 instruments. (3)F1 assign instruments to voices left to right. F2 sets tempo to $B8. (n)FE play song segments coded at pp bb relative to $0F00.	2 Build instruments & song structure: how to play which part and when

```
.:  0F78 00 FE 01 06 FE 01 CB FE
.:  0F80 01 00 FE 01 EA FE 01 00
.:  0F88 FE 02 4F FE 01 00 FE 02
.:  0F90 B4 FE 01 00 FE 03 14 FE
.:  0F98 03 6A FE 01 00 FE 01 EA
.:  0FA0 FE 01 00 FE 02 4F FE 01
.:  0FA8 00 FE 02 B4 FE 01 00 FE
.:  0FB0 03 14 FE 03 75.EA.EA.EA.
.:  0FB8 F6 01.18.13 31 F2 5C FE
.:  0FC0 03 90 FE 03 95 FE 04 C2
.:  0FC8 FE 06 B6 FE 04 C2 FE 06
.:  0FD0 C1.EA.EA.EA F6 01,1A.12
.:  0FD8 30 F2 4A FE 06 D6 FE 06
.:  0FE0 DC FE 08 13 FE 06 DC FE
.:  0FE8 08 23 FE 08 38 FE 08 38
.:  0FF0 FE 06 D0 FE 06 DC FE 08
.:  0FF8 23 FE 08 38 00 00 00 00
.?
```

— F6 changes sound of Ins#1.
F2 redefines tempo to $5C.
(n)FE play segments.

— F6 again redefines Ins#1.
F2 changes tempo to $4A.
(n)FE play segments.
00 in the last position ends command string.

```
.
.:  1000 00 00 00 00 00 00 20 74
.:  1008 00 00 55 20 70 00 00 15
.:  1010 10 30 00 61 00 10 74 00
.?
```

— *The music has been transcribed into 5 byte events.
The first byte is duration of a note.
Other four are notes translated into numeric code.

3
What to play

```
.
.:  10A8 00 00 00 30 F2 6D 00 5D
.:  10B0 0E 74 00 00 1D 02 00 00
.:  10B8 00 00 00 20 F4 6C 63 5C
.:  10C0 20 34 00 00 1C 1E 34 00
.?
.
.:  1720 2F 53 00 20 EB 00 00 53
.:  1728 23 2B 00 5A 13 26 2B 63
.:  1730 1A 13 30 2B 23 1A 13 00
.:  1738 10 00 00 00 00 10 77 00
.:  1740 00 00 20 79 00 00 00 30
.?
.
.:  1898 2A 23 17 22 73 2A 23 17
.:  18A0 44 F4 00 00 5C 26 34 00
.:  18A8 63 1C 38 34 68 00 1C 00
.:  18B0 00 00 00 00 00 00 00 00
.?
"
READY
```

*The score has been cut into musically meaningful segments. Only excerpts are shown here. A segment ends with a 00 in the duration position. The first segment begins at $1006, ends at $10BA. The second begins at $10BB. Last segment begins at $1738, ends at $18AE. A zero in $18AF and in the command string at $0FFC terminates the song. Segment addresses in FE commands are relative to $0F00.■

Real Sound in Real Time (cont.)

The program permits the user to select from the book and/or create "instruments" by specifying the amplitude and its rate of change over time for each harmonic. This sounds hard, but isn't. Entering X-Y (time-amplitude) coordinates does the trick, the program does the rest. The so defined composite amplitude envelope accomplishes the distinctions between various instruments: plucked, struck and blown instruments have been successfully implemented. The necessity for very high overtones does not yet permit building bowed instruments. There is no limit to the variety of sounds you can make (within the 8100 cps sampling rate), including different touch or attack characteristics, as well as crazy combinations of sound effects.

In fact, one of the features of this software I treasure most is that to the extent of available memory and speed of processing, it is infinitely flexible. Many sounds can be made, and any note-timing protocol can be set, not limiting you to a strict observance of the score. Knowledge of sound theory is not needed. The book gives enough information.

MTU has a demonstration audio tape available for $5. You may hear the sounds before deciding to plunge into the synthesis. For those who already have the program, MTU sells a floppy or two with precoded song data. The program and song data are loaded into the PET and, simply, RUN.

The explanation of the physics behind the system and the instructions on how to use the software are provided. I found only one, insignificant, error in the book. The instructions are complete, but difficult to use at first. The reason is that it is not immediately apparent from reading the descriptions of the available commands what is the commands' purpose. Their utility becomes obvious when one studies the book and a coded song together. The syntax of commands is unambiguous and yields an error message or silly sounding music if not observed. Locating an error is easy if the program's extensions are used.

How Does it Sound?

I like the sound. People who are familiar with the earlier, organ-like, music may be interested to know that through several ingenious programming tricks, Frank Covitz has improved the sound quality of the system, even though this program has to perform many more calculations in real time than it did previously. There are no "clicks" between the notes and the signal-to-noise ratio has been increased by an audible 6db.

Chamberlin and Covitz remind the readers that the sounds are "guitar-like," "horn-like" etc. and not quite "the real thing." What's missing is the "liveliness" and the resonances of the instrument when imitations are built (though piano in the lower range is superb, since it uses 26 harmonics!). The reason is that the 8 kc rate is too slow to permit high harmonics without running into distortion problems at high pitch. Hence, it is not appropriate to compare it with the real thing. But these tones are nevertheless pleasing and the key characteristics of the imitated instruments are clearly present. Invented instruments, of course, are not subject to such comparison, and, therefore sound just terrific. One of my favorites is a wind instrument that gently whistles while it plays, the whistle coming in a bit after the highest amplitude of the rest of the sound.

Musically, many of the instruments in the book and in the already transcribed songs, are beautiful. Their interesting characteristics and their variety compensate for the "smallness" of PET's sound. Many instruments chosen for a particular piece of music sound appropriate to the coded music, even if that piece of music was written for another instrument. It takes some doing to orchestrate things. I like the results produced by Covitz, Ashcraft and Chamberlin.

Educational Possibilities?

Listening to transcribed music, in many instances, makes the music more accessible to the listener. The impact is similar to, for instance, Segovia transcriptions of Bach's keyboard or violin works for the guitar or the Canadian Brass Quintet's transcriptions of all sorts of music for trumpets and things.

The system has a great entertainment value built in. It may also have an educational value. You can study various aspects of music and the physics of sound without fancy sound analyzing gizmos, and without a multitude of instruments at hand. Your ears and the graphing paper tell you just about the whole story.

It is interesting to hear music performed at different speeds, without changing pitch. It's also interesting to study what happens if a line of music is played backwards or by a variety of different instruments. It points out the tie between an instrument, melody, mood, etc. It can be used to study how instruments fit the music. It shows that to be able to utilize the multitude of possible sound configurations, new music may need to be written. These are some of the things that cannot possibly be observed with just a record player or a tape recorder.

A curious result of coding your own music is a realization of a fact, well known to musicians, that the musical score is only a hint and that it is totally inadequate as a means of communicating the composer's wishes (whatever they may have been). Pages of notes are a necessary requirement for music, but by no means sufficient. It takes some doing to tell the PET to not perform like a machine but it can be done since the program places almost no limitations on the user. Your coded music will not sound big and powerful, but it still can have a character of its own. Therein lies the fun of putting music on the PET.

How Easy to Code?

It could be worse if the book was ambiguous, which it isn't. But once you read the book, accept the syntax rules and code one or two measures of a song. It's not hard at all. Just tedious.

MTU considers this product an unfinished work. It is not promoted and as far as I can tell it will not be promoted until the "human interface", or a note and instrument entering editor is written. Writing such an editor is a great challenge, where half-way measures won't do. I can't tell when the editor will be available for the PET. Hence, we are stuck with coding instruments and music via the Machine Language Monitor. No machine code knowledge is required to use the system as it is an interpreter specifically designed for doing music. For instance, two of some 16 commands look like this: 'F2 tt' means tempo, 'FE pp bb' means play a segment that is in memory at location pp bb. Knowing how to use PET's Monitor and a sense of pages in its memory is needed, but one can learn by doing.

You have two options. One is to use the program and the song data for enjoying it and wait with coding your own things until the editor is written. The second option is to jump in now and not miss all the fun. The system, even though tedious in places, is useable, error-free and invites experimentation. Debugging extensions to the program have been provided which help locate a note, or a song segment for easy finding of coding errors.

In case you might be worried that your song data may become obsolete should an editor be introduced, I'd say "don't worry." One of the key policies of MTU has been compatibility. It is unlikely they will obsolete anything. Just as, at the present time, a song coded on an APPLE, for instance, will play, **with no modifications** on any one of PET's many releases and 6502 relatives, I am pretty sure the introduction of an editor will not change a thing.

System Considerations

MTU strongly recommends a 32K PET. Only several available songs will play in a 16K PET. Neither the program (2.5K) nor the song data (varies: ¼K-2K is a good guess) use much memory. The waveforms for the instruments, however, gobble it up pretty fast. A disk drive is not essential, though always very helpful. Monitor extensions, such as the Supermon or Extramon are essential. They permit easier editing of data, specifically, inserting, deleting and transferring code. You can't do without them. Both are public domain programs, available from various sources. The MTU's Visible Memory board can, optionally, be used to see the elements of an instrument. A digital-to-analog converter board needs to be plugged into the User port of the PET. Two DACs can be used for stereo effect (2 voices to each channel). In a concert hall the stereo effect was incredible, but in a home situation it's not needed (I'm biased: I think monaural records are OK).

Other Information

Making this kind of music on your microcomputer is a lot of fun now and the results can be quite musical, even though the sound quality is that of a slightly noisy AM radio if only quiet instruments are picked. Do not let this discourage you. We can reasonably expect true high fidelity sound in a short time. All it will take is a faster processor and larger memories. It's worth taking the plunge now and be ready for progress.

REFERENCES

(1) Hal Chamberlin, A Sampling of Techniques for Computer Performance in Music, BYTE magazine, September 1977.
(2) Hal Chamberlin, Advanced Real Time Music Synthesis Techniques, BYTE magazine, April 1980.
(3) Hal Chamberlin, Musical Applications of Microprocessors (a big book, 653 pages), $25.

References (1) and (2) contain much of the information necessary to understand the PET music system. They are written in simple language while presenting quite complex physical theory of sound generation on a computer. I found both to be invaluable in understanding how to use the programs. Reference (2) deals specifically with instrument synthesis. The description of coding is slightly more difficult than it needs to be, so read it only to grasp the general idea.

The debugging extensions, which, in my opinion, are most desirable, are available for $20 from MTU MUSIC, c/o Keith Sproul, 1368 Noah Road, North Brunswick, NJ 08902.

Micro Technology Unlimited, 2806 Hillsborough Street, PO Box 12106, Raleigh, NC 27605: PET DAC—$59, Instrument Synthesis software $49. ■

PROGRAMMER'S TIPS

An EASY Cursor Positioning Routine

After having programmed in a BASIC with many more extensions, structured constructs, and easy file handling commands, you might easily become frustrated writing code in Commodore BASIC. Of course, when working on the larger machines that support those types of BASICs, you don't ever get a feeling or understanding of how the operating system really works. One of the nice features about the PET is that you can actually look at the operating system, change certain parts of it to fit your own needs, and use some of the ROM routines to interface with your own programming.

One of the features I missed most was the ability to move the cursor to an X,Y location on the screen and print a message there. After a few attempts to position the cursor using POKEs to address 198 (position of cursor on above line) and 216 (line where cursor lives), I decided to write my own routine.

The routine is written in machine language and sits at the top of RAM, so that it can be used by any BASIC program that is loaded. The routine is entered via a SYS 32256, and your BASIC program must pass certain parameters as in the format:

sys 32256,y,x,"your message should appear here"

Where "y" is any line number between 0 and 24 (25 lines), and "x" is any column position between 0 and 79 (80 columns). If the format is not as that listed above, or your X,Y coordinates are not in the correct range, the routine will break with a "syntax error."

I have provided both a BASIC poker program that readjusts the top of memory in order to secure the routine and pokes in the appropriate code for the routine, and the actual source assembly listing for those of you who have an interest or wish to make changes.

Since it may be awkward to run the poker program everytime, you may wish to run it once, then break into the monitor and do a machine language "save" to save it in program format (for subsequent "dloads"). In this case, I have also provided a loader program which first readjusts the top of memory pointers and the start of variable pointers (according to your largest program module in your application), then loads in the "program" version of the routine, and finally loads in your first application program, in this case, the "master menu."

This program will also work on 40 column PETs if you change the value "80" on line 200 of the BASIC poker program to "40".■

—Dave Scott

```
0000            ;*************************************************************************
0000            ;****                                                                 ****
0000            ;**** program name = display.src                                      ****
0000            ;**** author = dave scott, cbm  us                                    ****
0000            ;**** date = 02/23/82                                                 ****
0000            ;****                                                                 ****
0000            ;**** purpose: the purpose of this program is to print a message at   ****
0000            ;****          a specified x,y location on the screen.  it is         ****
0000            ;****          entered via the "sys" command and the basic program    ****
0000            ;****          must pass certain parameters.  the syntax for the sys ****
0000            ;****          command is:                                            ****
0000            ;****                                                                 ****
0000            ;****          sys 32256,15,30,"this is test"                         ****
0000            ;****                                                                 ****
0000            ;****          where 32256 is the sys address, 15 is the line, 30 is ****
0000            ;****          the column, and the string in between quotes is the    ****
0000            ;****          message to be printed at that x,y location.  seal the ****
0000            ;****          routine off from basic by doing a poke 52,126.         ****
0000            ;****                                                                 ****
0000            ;*************************************************************************
0000            ;
0000            ;*******************************************
0000            ;****          kernal routines          ****
0000            ;*******************************************
0000            ;
0000            outchr=$ffd2           ;print char in acc
0000            ;
0000            ;*******************************************
0000            ;****      rom and system routines      ****
0000            ;*******************************************
0000            ;
0000            chrget=$0070           ;char get (in 4.0)
```

```
0000                chrgot=$0076          ;char got (in 4.0)
0000                synerr=$bf00          ;syntax err (in 4.0)
0000                ;
0000                ;****************************************
0000                ;****        storage areas           ****
0000                ;****************************************
0000                ;
0000                txtptr=$0077          ;current char
0000                chkspc=$007f          ;check for space
0000                ;
0000                ;***************************************
0000                ;****        main routine           ****
0000                ;***************************************
0000                ;
0000                       *=$7e00
7e00                ;
7e00   20 13 7e            jsr savcod                         ;save chrget code
7e03   20 2f 7e            jsr movey                          ;cursor to line y
7e06   20 39 7e            jsr movex                          ;cursor to column x
7e09   20 43 7e            jsr prtmsg                         ;display message
7e0c   20 24 7e            jsr getcod                         ;restore chrget
7e0f   20 70 00            jsr chrget                         ;get next char
7e12   60                  rts                                ;return to basic
7e13                ;
7e13                ;
7e13                ;***************************************
7e13                ;****    save chrget code 7f-80     ****
7e13                ;***************************************
7e13                ;
7e13   a5 7f        savcod lda chkspc                         ;these lines save
7e15   8d f6 7e            sta store                          ;the code from chrget
7e18   a5 80               lda chkspc+1                       ;that check for a
7e1a   8d f7 7e            sta store+1                        ;space and replaces
7e1d   a9 ea               lda #$ea                           ;them with a nop
7e1f   85 7f               sta chkspc                         ;instruction "ea"
7e21   85 80               sta chkspc+1                       ;
7e23   60                  rts                                ;
7e24                ;
7e24                ;***************************************
7e24                ;**** put back chrget code 7f-80    ****
7e24                ;***************************************
7e24                ;
7e24   ad f6 7e     getcod lda store                          ;these lines put
7e27   85 7f               sta chkspc                         ;back the original
7e29   ad f7 7e            lda store+1                        ;chrget code
7e2c   85 80               sta chkspc+1                       ;
7e2e   60                  rts                                ;
7e2f                ;
7e2f                ;***************************************
7e2f                ;****     cursor to line y          ****
7e2f                ;***************************************
7e2f                ;
7e2f   20 b8 7e     movey  jsr chkcom                         ;check for comma
7e32   20 69 7e            jsr gety                           ;get value of y
7e35   20 7b 7e            jsr posy                           ;cursor to y
7e38   60                  rts
7e39                ;
7e39                ;***************************************
7e39                ;****    cursor to column x         ****
7e39                ;***************************************
7e39                ;
7e39   20 b8 7e     movex  jsr chkcom                         ;check for comma
7e3c   20 90 7e            jsr getx                           ;get value of x
7e3f   20 a2 7e            jsr posx                           ;cursor to x
7e42   60                  rts
7e43                ;
7e43                ;***************************************
7e43                ;****        print msg              ****
7e43                ;***************************************
7e43                ;
```

➡

```
7e43   20 76 00    prtmsg jsr chrgot                                         ;get current char
7e46   c9 2c              cmp #44                                            ;is it = ","
7e48   f0 03              beq prt1                                           ;  yes, print msg
7e4a   4c b2 7e           jmp err                                            ;  no print syntax
7e4d                      ;                                                       error
7e4d   20 70 00    prt1   jsr chrget                                         ;get next char
7e50   c9 22              cmp #34                                            ;check for quotes
7e52   f0 03              beq prt2                                           ;  yes, go print
7e54   4c b2 7e           jmp err                                            ;  no print syntax
7e57                      ;                                                      error
7e57   20 70 00    prt2   jsr chrget                                         ;get next char
7e5a   c9 22              cmp #34                                            ;check for quotes
7e5c   f0 0a              beq prtend                                         ;  yes, done
7e5e   c9 00              cmp #0                                             ;  end of line??
7e60   f0 50              beq err                                            ;   yes, print
7e62                      ;                                                       syntax error
7e62   20 d2 ff           jsr outchr                                         ;
7e65   4c 57 7e           jmp prt2                                           ;get next char
7e68                      ;
7e68   60          prtend rts                                                ;done
7e69                      ;
7e69                      ;***************************************
7e69                      ;****    get value of y             ****
7e69                      ;***************************************
7e69                      ;
7e69   f0 47       gety   beq err                                            ;invalid char
7e6b   b0 45              bcs err                                            ; print syntax error
7e6d   20 c3 7e           jsr aschex                                         ;conver # to hex
7e70   ad f5 7e           lda temp                                           ;temp=converted val
7e73   c9 19              cmp #25                                            ;is y < 25
7e75   b0 3b              bcs err                                            ;  no
7e77   8d f3 7e           sta ycoord                                         ;store y coordinat
7e7a   60                 rts                                                ;
7e7b                      ;
7e7b                      ;***************************************
7e7b                      ;****    move cursor to line y      ****
7e7b                      ;***************************************
7e7b                      ;
7e7b   ae f3 7e    posy   ldx ycoord                                         ;cursor down using
7e7e   a9 13              lda #19                                            ; ycoord as counter
7e80   20 d2 ff           jsr outchr                                         ; home cursor
7e83   e0 00              cpx #0                                             ;  is y = 0?
7e85   f0 08              beq posyen                                         ;  yes, done
7e87   a9 11              lda #17                                            ; 19 = home
7e89   20 d2 ff    posy1  jsr outchr                                         ; 17 = cursor down
7e8c   ca                 dex                                                ;
7e8d   d0 fa              bne posy1                                          ;
7e8f   60          posyen rts                                                ; done
7e90                      ;
7e90                      ;***************************************
7e90                      ;****    get value of x             ****
7e90                      ;***************************************
7e90                      ;
7e90   f0 20       getx   beq err                                            ;invalid char
7e92   b0 1e              bcs err                                            ; print syntax error
7e94   20 c3 7e           jsr aschex                                         ;convert x to hex
7e97   ad f5 7e           lda temp                                           ;temp=converted val
7e9a   c9 50              cmp #80                                            ;is x > 80??
7e9c   b0 14              bcs err                                            ;   yes
7e9e   8d f4 7e           sta xcoord                                         ;store in xcoord
7ea1   60                 rts                                                ;
7ea2                      ;
7ea2                      ;***************************************
7ea2                      ;****    move cursor to x           ****
7ea2                      ;***************************************
7ea2                      ;
7ea2   ae f4 7e    posx   ldx xcoord                                         ;cursor to column x
7ea5   e0 00              cpx #0                                             ;  if x=0, done
7ea7   f0 08              beq posxen                                         ;  if not,
7ea9   a9 1d              lda #29                                            ;  use xcoord as
7eab   20 d2 ff    posx1  jsr outchr                                         ;  counter
```

```
7eae  ca                 dex                                        ;  29 = cursor ->
7eaf  d0 fa              bne posx1                                  ;
7eb1  60          posxen rts                                        ;
7eb2              ;
7eb2              ;
7eb2              ;************************************************************************
7eb2              ;****                     utility routines                           ****
7eb2              ;************************************************************************
7eb2              ;
7eb2              ;*******************************************
7eb2              ;****    jump to syntax error routine   ****
7eb2              ;*******************************************
7eb2              ;
7eb2  20 24 7e    err    jsr getcod                                 ;restore chrget code
7eb5  4c 00 bf           jmp synerr                                 ;print syntax error
7eb8              ;
7eb8              ;*******************************************
7eb8              ;****     check basic line for comma    ****
7eb8              ;*******************************************
7eb8              ;
7eb8  a9 2c       chkcom lda #44                                    ; 44 = ascii ","
7eba  a0 00              ldy #0                                     ;
7ebc  d1 77              cmp (txtptr),y                             ;compare to current
7ebe              ;                                                     basic char
7ebe  d0 f2              bne err                                     ;if not = ","
7ec0              ;                                                      print syntax error
7ec0  4c 70 00           jmp chrget                                 ;get next char
7ec3              ;
7ec3              ;*******************************************
7ec3              ;**** conv 2 digit ascii number to hex ****
7ec3              ;**** store result in temp            ****
7ec3              ;*******************************************
7ec3              ;
7ec3  38          aschex sec                                        ;set carry for sub
7ec4  e9 30              sbc #48                                    ;1st digit to hex
7ec6  8d f5 7e           sta temp                                   ;store it in temp
7ec9  20 70 00           jsr chrget                                 ;get second digit
7ecc  90 07              bcc asc1                                   ;goto to asc1 if
7ece              ;                                                      btwn 0-9
7ece  c9 2c              cmp #44                                    ;  is it = ","
7ed0  f0 20              beq ascend                                 ;    yes, done
7ed2  4c b2 7e           jmp err                                    ;    no, syntax error
7ed5              ;
7ed5  38          asc1   sec                                        ;set carry for sub
7ed6  e9 30              sbc #48                                    ;2nd digit to hex
7ed8  48                 pha                                        ;save 2nd digit
7ed9  a9 00              lda #0                                     ;
7edb  ae f5 7e           ldx temp                                   ;find tens by mult
7ede  18          asc2   clc                                        ;
7edf  69 0a              adc #10                                    ; 1st x 10
7ee1  ca                 dex                                        ;
7ee2  d0 fa              bne asc2                                   ;
7ee4  8d f5 7e           sta temp                                   ;store tens in temp
7ee7  68                 pla                                        ;get 2nd digit
7ee8  18                 clc                                        ;
7ee9  6d f5 7e           adc temp                                   ;  and add to temp
7eec  8d f5 7e           sta temp                                   ;  save it in temp
7eef  20 70 00           jsr chrget                                 ;  get next char
7ef2              ;
7ef2  60          ascend rts                                        ;done
7ef3              ;
7ef3              ;*******************************************
7ef3              ;****       storage areas               ****
7ef3              ;*******************************************
7ef3              ;
7ef3  00          ycoord .byte 0                                    ;line coordinate
7ef4  00          xcoord .byte 0                                    ;column coordinate
7ef5  00          temp   .byte 0                                    ;work area
7ef6  00          store  .byte 0,0                                  ;storage for code
7ef7  00
7ef8              ;
```

Are You A Programmer?

Can you code programs for the Commodore line of computers? Do you have a company which does this and/or offers system consultation? If so, fill out this form and send it to the Approved Products Manager. We are starting a new section in the Software Encyclopedia that will list people or firms that offer programming services for Commodore users. For instance, if a user has a need for a non-existent application, or has a software product which is ALMOST what he needs, but requires modification, he will probably need the services of an independent programmer. If you can fill that need and you would like to have your firm added to the encyclopedia, get your name, address, and phone number to us immediately, in order to make it into the next edition of the Commodore Software Encyclopedia.

Name______________________________

Company Name________________________

Address____________________________
(street)

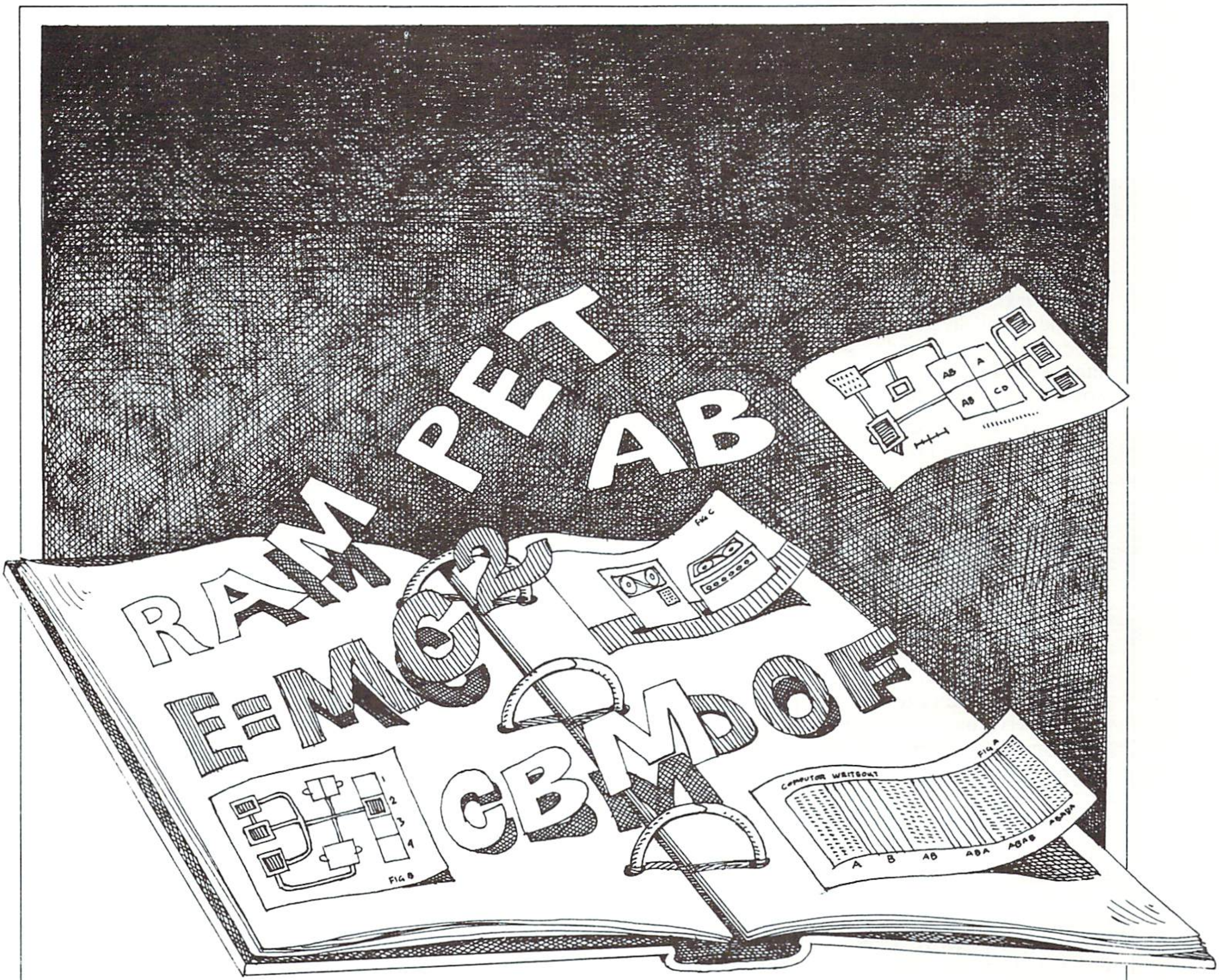
____
(city) (state) (zip)

Phone (_____) _____-__________ Hourly Rate $________ per hour
(area code)

Experience__________________________

____

____



Excerpts from a Technical Notebook

Here is some additional information on accessing the serial port of the SuperPET which was not available at press time for the last issue. The RS-232 connector on the SuperPET is located on the middle circuit board near the right front corner. It is a standard 25-pin, female connector. The signals brought out to the connector are shown in the table below.

Pin Nr.	Signal Description	
1	Protective Ground	
2	Transmitted Data	(TxD)
3	Received Data	(RxD)
4	Request To Send	(RTS)
5	Clear To Send	(CTS)
6	Data Set Ready	(DSR)
7	Signal Ground	
8	Data Carrier Detect	(DCD)
20	Data Terminal Ready	(DTR)

SuperPET RS-232 Connector Pin-outs

Positioning for DATA READS

Everyone knows how RESTORE, READ and DATA statements operate. The first READ gets the first DATA element, and so on. RESTORE sets the READ pointer back to the beginning of text. But there is no command that allows positioning to a particular DATA line.

This could be useful if, for example, a DATA line were part of a subroutine. The only way to accomplish this in strict BASIC is to RESTORE and then issue enough READ commands to position to the desired data. This can be a pain!

→

RUN, CLR or RESTORE sets the DATA Read Pointer (address 62 and 63 decimal) back to $0400; the start of BASIC text. When a READ command is given, this pointer starts advancing through text looking for a 'DATA' line. If the pointer reaches the end of text before finding data, an ?OUT OF DATA ERROR occurs.

PET maintains another pointer that climbs up and down through text. This pointer is part of the CHRGET routine and essentially points at the code currently being executed. If this pointer (addresses 119 & 120) is transferred into the DATA Read Pointer, the next READ command would force a search to the next DATA line.

```
 10 DATA FIRST, SECOND, THIRD
 20 DATA FOURTH
 30 READ A$, B$
 40 POKE 62, PEEK(119) : POKE 63, PEEK(120)
 50 READ A, B
 60 PRINT A$, B$, A, B
 70 DATA 1, 2, 3, 4
 80 END
```

The READ command in line 30 gets "FIRST" into A$ and "SECOND" into B$, leaving the pointer pointing at "THIRD". Line 40 moves the pointer past line 10 and line 20 leaving it at some point in line 40. Since this is obviously not a DATA statement, the next READ causes an advance to line 70.

In summary, the POKEs of line 40 position to the *next* DATA statement in text.■

Spooling Disk Files to Printers

Reproduced from Transactor

In Compute #8, T.M. Peterson published a neat trick for getting the 2040 disk to talk to a printer without PET/CBM supervision of the IEEE bus. I imagine this would work for 4040s, 8050s and any make printer interfaced via the IEEE bus, but naturally I can't be sure for all cases. However, the idea was so incredible that I felt it definitely worth repeating.

Everyone knows how to LIST a program to the printer. But long listings can wear patience thin, especially on a slow printer! Not only that, but while your printer is chugging along, the PET just sits there with everything disabled except RUN/STOP. Wouldn't it be nice if the disk fed the printer while you continue editing or play a quick round of space invaders or Microchess, that is if you can bear some of those arrogant printers. By the way, those wing nuts on the bottom of Commodore 202X printers . . . take them out. They're only shipping screws that hold the mechanism tight. Once removed, the noise level is reduced considerably.

First a file must be created on disk. This could be any SEQ file with any contents that are printer recognizable, but for now we'll create one of a program LISTing.

```
1. Enter some small program
2. OPEN 8,8,8, "0:TEST SPOOL,S,W" : CMD8 : LIST
3. PRINT#8,""; : CLOSE8
4. NEW
5. OPEN 8,8,8,"0:TEST SPOOL"          ;defaults to ',s,r'
6. POKE 165,72 : SYS 61695            ;use SYS 61668
7. POKE 165,104 : SYS 61695              for BASIC 2
8. OPEN 4,4 : CMD4 : POKE 176,3 : POKE 174,0
```

On hitting return the printer should fire up and continue at full speed to the end of the file. At this point your cursor might be acting funny (try hitting return on a blank line). To stop this, POKE 14,0 for BASIC 2.0 or POKE 16,0 for BASIC 4.0. If you're lazy like me, invoking a couple of ?SYNTAX ERRORS (e.g., '=' and Return) will do the same thing. However, to restore normal cursor operation under program control (yes program control!), you would have to use the POKE. More on this in upcoming paragraphs.

Once the printer starts, don't try using the IEEE bus or the spool will abort. When it's all finished you can CLOSE the open disk file by sending an Initialize command or with:

```
OPEN 1,8,8 : CLOSE 1
```

A filename isn't necessary, but use the same secondary address as in step 5.

The NEW command at step 4 is only for clarity. Instead you might load another program or just leave the current one in for further editing. You can RUN the program in memory and even use the cassettes, but they're still as slow as before.

The example here uses all direct commands but they could just as easily be put in a program. Think of the applications! In a user oriented system, a report could be output to the disk and immediately spooled to the printer while the operator continues working on the next task. Of course the user might inadvertently try a bus operation which would kill everything. Fortunately this busy state can be detected using the following "trap":

```
 100 IF (NOT(PEEK(59456))) AND 64 THEN 100
 110 OPEN 1,8,SA: CLOSE 1
 130 . . . and continue
```

If a spool is in progress, line 100 will loop back to itself until the bus is free. Line 110 is for closing the disk files and also turns off the active LED. SA is a variable containing the secondary address which might be used in coding the OPEN command that starts the operation.

Also note that line 110 causes no disk activity so there's no need to go around it to save time.

Theory and Variations

This example used device number 8 right through. However, you might have more than one disk on line which would mean a different device number. In step 6, address 165 (the IEEE output buffer) is POKEd with 72. This number (72) is derived from 64 + 8, where 8 is the device number. For versatility, this '8' might be replaced by a variable like DV.

The following SYS activates the ATN line on the bus telling all devices to 'pay attention'. The contents of 165 are then sent to the bus but only the device that has a matching 'talk' address responds, in this case device 8; the disk.

The disk is ready to start sending but from where? All it needs now is the secondary address of the OPENed file. Step 7 sets up the output buffer with the secondary address plus 96. Since 8 was chosen, the result is 104. This step might also be modified to read POKE 165,96 + SA.

Nothing happens yet because the ATN line isn't released by the PET. When CMD4 is executed (step 8), the printer becomes the output command device. PET releases ATN, the disk starts talking and the printer listens.

POKE 176,3 tricks the PET into thinking the output command device is the screen and POKE 174,0 simulates no files OPEN. In a program you would have to re-open files (e.g., command channel, modem, etc.) at spool completion. By the same token, you might want to CLOSE any open write files before starting.

Digital Voltmeter Program

This program continuously reads a Hewlett Packard Model 3455A DVM. The "talk only" switch must be in talk only mode. This short program might be included as a subroutine in a larger data collection program. ■

```
50 DIMA(14)
90 A = 59424
91 B = 59425
95 F = 59456
100 FORI = 0TO14
105 POKEF,253:REM**SET NRFD LOW
107 POKEB,52:REM**SET NDAC LOW
130 POKEF,255:REM**SET NRFD HIGH
132 IFPEEK(F) > 128THEN132:REM**IS DAV LOW
134 POKEF,253:REM**SET NRFD LOW
140 A(I) = 255 - PEEK(A):REM**GET DATA
160 POKEB,60:REM**SET NDAC HIGH
170 IFPEEK(F) < 128THEN170:REM**IS DAV HIGH
180 IFA(I) = 10THEN210
200 NEXTI
210 FORK = 0TOI
212 PRINTA(K);
215 A$ = A$ + CHR$(A(K))
216 NEXTK
217 PRINTA$:A$ = ""
220 GOTO100
READY.
```

VIC and PET are registered trademarks of Commodore Business Machines.

Enter The UMI Dimension of Mind

The Dimension of Mind is an extension of sight and sound. A dimension whose only boundary is imagination. These same characteristics are inherent in the extraordinary line of hardware and software products manufactured by UMI for Commodore's VIC, PET, and other micro-computers.

UMI offers micro-computer users unique, innovative entertainment, educational, and professional programs on tape cassette or on UMI's own durable cartridges. UMI also provides memory expansion devices, communications programs, as well as programmer and hobbyist aids.

To begin your journey into The Dimension of Mind, please mail attached coupon to receive UMI's 1982 Catalog or call (714) 594-1351.

Please send me the
UMI 1982 CATALOG
UNITED MICROWARE IND.,INC.
3503 TEMPLE AVE., SUITE C
POMONA, CA 91768
NAME ____________
ADDRESS ____________
CITY ________ STATE ________
ZIP ________ COMPANY ________

umi

united microware industries, inc. • 3503 temple avenue • suite c • pomona, california • 91768

Hardware Review
Madison Computer's Z-RAM Card

What would you say if you were offered a simple way to multiply the number of applications available for your Commodore system ten-fold, have access to a number of high-level computer languages, plus have expanded user memory and a number of advanced hardware interfaces . . . all at the same time?

You would probably answer that all this is pushing things a bit too far, especially on a PET or CBM computer. Well, we're not dreaming and all this is possible using a simple hardware add-on for your system which allows you to use CP/M® (Control Program for Microprocessors).

As a bit of background let's take a look at what CP/M is, in general, and then how you can use its power on Commodore equipment. One of the problems with microcomputers, in fact any computer, is that each has a unique set of commands that control the system.

But at the lowest level, each computer must perform the same tasks: get data from the keyboard, print information and handle disk activity. These tasks are usually handled by a "manager" referred to as an operating system.

On the standard PET, a combination of BASIC and DOS (Disk Operating System) perform these activities. If you're just using PET programs everything's great, but what happens if you want to use a program designed for some other system? For the most part—tough luck. Without extensive modifications to the program—the time and expense is hardly worth it—there is no way an alien program will run on a different system than it was designed for because each operating system was produced for a unique piece of hardware.

Fortunately, in the early days of microcomputers this problem of incompatibility was addressed, and CP/M was born. The unique thing that makes CP/M so popular is that in the design of the system all the hardware dependent parts of CP/M were put in one part of the program. This way, a developer can change just those portions of CP/M that involve specific hardware, without touching the main application. Since no changes are required to the main program, it can be used by any system capable of running CP/M.

CP/M On The PET

CP/M has two hardware requirements that are not met by standard Commodore equipment. The first is: CP/M was originally designed around an 8080 microprocessor. Since the PET contains a 6502, whose instruction set is not compatible, a different microprocessor must be added to the PET. The second requirement is that the system must have at least 48K of user memory, which the PET normally doesn't contain.

The way around this is a hardware add-on that provides a Z-80 microprocessor (8080 compatible) and additional user memory (RAM). The Z-RAM card from Madison Computer (1825 Monroe St., Madison, WI 53771 and distributed by Computer Marketing Services in Cherry Hill, NJ) opens up the world of CP/M to your PET.

Z-RAM Card

Physically, Z-RAM is a separate card containing a Z-80 microprocessor, a 6502 processor and 64K of additional RAM. The card is designed to fit inside the top part of the PET enclosure, directly under the monitor. Four mounting screws make the physical installation a snap.

Z-RAM is designed to work with either a 40-column PET or 80-column 8032 system. The advantage of using the 8032 is that most CP/M programs were originally designed to support 80-column terminals. In fact, the 8032 is looked upon as a terminal by the Z-RAM card.

The only electrical connections involve unplugging the power cable to the main PET motherboard, and connecting this cable to the Z-RAM board. Another cable then brings power from the card back to the PET. The final connection involves removing the 6502 microprocessor from the PET's main logic board and attaching a 40-conductor cable from the Z-RAM card to the 6502 socket.

All of this only takes a few minutes to complete. The cables are "keyed" allowing them to be inserted only one way. Then just close the case and go on.

The construction of the Z-RAM card is excellent, with all critical circuits socketed. The card also shows no signs of last minute "fixes" which normally appear as external point-to-point wiring. The RAMs are state-of-the-art 64K low power devices, thus keeping overall board size to a minimum. You have the full 64K RAM work space also.

In using this extra RAM from the PET, you can split memory to accommodate three programs simultaneously. The first bank contains 26K RAM with the second and third banks each containing 32K RAM. This is advantageous for large programs since the wait normally associated with going to the disk to bring in another part of the program is eliminated.

In operating the card in this mode the thing you must watch for is programs that leave disk files open. Switching to another program without locking things up properly will probably cause loss of data.

Z-RAM supports printers either through the standard Commodore IEEE-488 interface or through Madison Computer's McTERM RS 232 serial port. McTERM is their comprehensive communication package which uses a connection to the PET's

→

user port to supply a standard RS 232 signal.

To use CP/M with Z-RAM, just boot the supplied CP/M disk. After a short wait—CP/M is a small program—the opening message will be displayed along with the CP/M ready prompt. Another side note: CP/M is not the most efficient with disk space.

Using Commodore's 8050 disk with a total of 500K on a diskette will give you a most effective operating environment. Though using a 4040 or 2031 single disk is certainly acceptable, just a bit inconvenient.

CP/M Operation

The nice thing about CP/M is that once you learn it, the operation is the same for all CP/M based systems, not just the Commodore implementation. Remember, we mentioned compatibility earlier!

There were no surprises in using version 2.20B of CP/M supplied with Z-RAM. Each operation provided the expected results, from getting a directory of the disk to using a standard CP/M application.

To get general again for a moment, you may be interested in exactly how CP/M differs from standard PET operation.

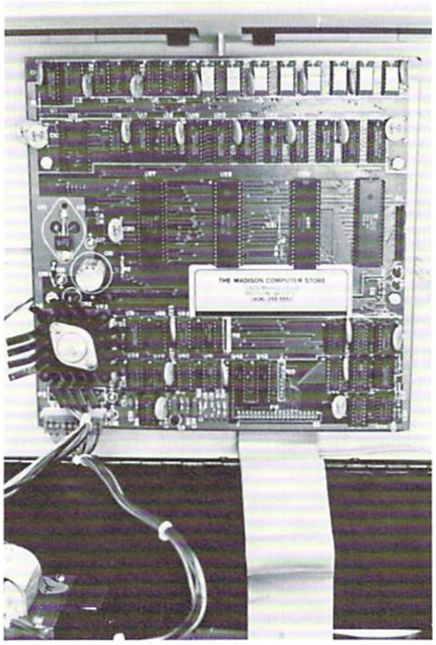
Rather than naming the disk drive units '0' and '1', which is standard with CBM disk units, CP/M names disks 'A:' and 'B:'.

The CP/M system prompt will be 'A>' or 'B>', depending on the drive you select. This is called the "logged" drive. From this point you can get a directory of the programs on the disk, inquire about specific information regarding a file or the entire disk, perform housekeeping duties such as file transfer and disk formatting, and, of course, run specific programs.

The standard CP/M disk supplied with Z-RAM contains the support programs to perform the functions mentioned above. In addition, the standard Microsoft BASIC language is included.

A CP/M directory will look quite different from a standard CBM disk catalog:

```
A>DIR
A: FORMAT   COM : COPY      COM : MBASIC COM
A: PIP      COM : STAT      COM : ED     COM
A: ASM      COM : DOWNLOAD  COM : WS     COM
A: PR MENU  BAS : PR PGR    BAS :
```



Looking at the directory, you can usually tell the type of file by the suffix, i.e., 'COM', 'BAS', etc. A COM file is a command file, which requires you to just type the file name and it will be executed immediately.

A 'BAS' file is a BASIC program that requires the loading of the Microsoft BASIC language first. Once that's accomplished—by typing MBASIC—you can type "RUN FILENAME" and the BASIC program will be loaded and executed. Remember that in using BASIC, memory is taken up by CP/M, then BASIC, and finally the application.

But the real power of CP/M, again, is in the number of applications available, and the portability of those programs.

In testing the Z-RAM card, a number of standard CP/M applications were run through their paces, and each performed flawlessly. In fact, to get some of the applications on CBM formatted disks, a CP/M communication program was used to transfer CP/M programs from another system to a CBM 8032.

Because CP/M based programs are designed to run on various sytems, most programs are supplied with an "install" utility. This does the final set-up of the CP/M application so it will run on a specified piece of hardware.

Install usually configures the program for a particular type of terminal. Since the Z-RAM card can make the PET look like a number of standard terminals, there was no problem configuring a CP/M program like WORDSTAR, the premier word processing package, to run properly.

In the seven years that CP/M has been around, thousands of programs have been created that run under it, written by over 100 companies. Applications range from languages including business BASIC, FORTRAN, COBOL and Pascal, and development utilities like assemblers, to application programs such as accounts payable/receivable, data bases, financial planning, and word processing.

The Z-RAM card is an ideal way to gain access to this wealth of software for your PET. ■

—Mike Heck

CP/M is a registered trademark of Digital Research, Inc.

BACKPACK™

Battery Backup System for Commodore Computers

FLOPPY BACKPACK™

Battery Backup Systems for Commodore Dual Disk Drives

BACKPACK and FLOPPY BACKPACK are total, rechargeable battery backup systems for the COMMODORE line of computers and disk drives.

BACKPACK was designed specifically for the COMMODORE COMPUTER SYSTEM. It supplies a minimum of 15 minutes (maximum of 25 minutes) of battery backup power to either the CBM/PET 4000 series or the CBM 8000 series (dealer installation is required for the SuperPET). BACKPACK supplies reserve power to the total computer, including monitor, cassette drive, memory, operating system, and keyboard. BACKPACK installs within the cabinet of the COMMODORE COMPUTER, has its own off/on switch and is constantly recharged, during normal operation, from the computer's own internal power supply. Life expectancy is approximately 3 to 5 years. Installation is simple, requires no wiring changes or alterations and will take even the novice user approximately 10 minutes to accomplish.

FLOPPY BACKPACK is the COMMODORE DUAL DRIVE FLOPPY DISK battery backup system. It too supplies a minimum of 15 minutes (maximum of 25 minutes) reserve power to either the CBM 4000 or CBM 8000 dual drive disk units. FLOPPY BACKPACK will supply power to both drives and the operating system. FLOPPY BACKPACK installs within the cabinet of either unit, has its own off/on switch, and is constantly recharged, during normal operation, from the disk's own internal power supply. Life expectancy is approximately 3 to 5 years. Installation is simple, requires no wiring changes or alterations and will take even the novice user approximately 10 minutes to accomplish.

When both units are installed within a system, power outages, and gliches have no effect on either the CPU or disk drives because BACKPACK and FLOPPY BACKPACK automatically cut in to supply power to all phases of the system. BACKPACK also eliminates the problems that occur because of power surges and spikes. It is the best line filter available for the COMMODORE. Disk crashes, loss of valuable data and valuable time, etc. are totally eliminated. The 25 minutes (maximum) of reserve power gives the user time to store data to disk, properly close all files, power down in an organized manner, or continue collecting information and never drop one bit of data. And since both units are recharged during the normal operation of the system, you are never without reserve power.■

Suggested Retail Price—$225.00 per unit
Available from: ETC Corp., Apex, NC (919) 362-4200

Ticker Tape Information Processing System

Here is an interesting program for those of you who are seriously interested in the stock market, i.e., you make a good part of your living in the stock market. The Ticker Tape Information Processing System (or TTIPS for short) brings the Big Board right into your office or living room. You rent a Western Union ticker tape line from the N.Y.S.E. on a monthly basis and the program will display the following data for all stocks, right on your CBM's CRT:

- opening price
- latest price
- previous price
- accumulated volume
- previous high
- previous low

Any 16 stocks can be displayed simultaneously in an uncluttered and easy to read format. There is also a real-time clock continuously displayed.

Additional displays include: the last 16 trades—price and volume of any 96 stored stocks, any 2 simultaneously; most active stocks—continuous display of leaders; Dow Jones Average—net change continuously updated; total market volume—continuously updated; tic-sum—continuous display of combined up and down tics for every transaction.

The special tic-sum feature is a running total, from the opening bell, of the combined up and down tics of every transaction of every stock. This feature has been indispensible in spotting early trend changes.

It should be mentioned that there are two other options available which do not require the expense of an instantaneous dedicated line. The first is a much less expensive dedicated line from the N.Y.S.E., which is delayed by 15 or 20 minutes. This should make little difference to all but the most serious user. The second option is a dial-up to Western Union, which again will give a delayed tape. It will only be useful for giving a delayed Big Board reading across the screen. None of the other functions of the program will work since the data must be accumulated from the opening bell. This is even less expensive and can be used at will, much like making a phone call.

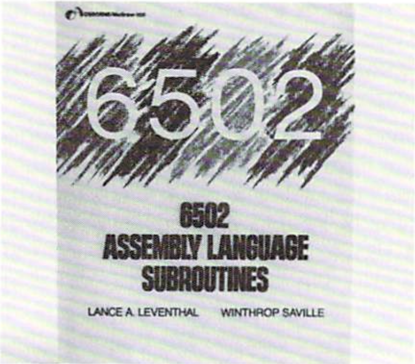
So if your business requires a ticker tape or instant and continuous access to the Big Board display on the floor of the N.Y.S.E., this seems to be the package for you. To our knowledge there is nothing else like it.

This product is available from AQR Products, 53 Clifford Rd., Sudbury, MA 01776. Their phone number is (617) 443-2906.■

NEW PRODUCT DEVELOPMENTS

In the fast paced microcomputer industry, new products are developed at an incredible rate – sometimes too quickly for even the most informed computer person. To do our part in keeping you abreast of Commodore-related product developments, this section will often appear in Commodore Magazine. *Its sole purpose will be to announce NEW developments in hardware and software products that are compatible with Commodore equipment. Publication of these new product announcements does not necessarily mean that Commodore approves or recommends the products; we only wish to provide timely updates on the far-reaching world of Commodore microcomputers. It is a service that our readers – and all users of Commodore products – deserve.*

If you have a new product or service that's worth mentioning, don't hesitate to send the information to the NEW PRODUCTS DEPARTMENT in care of this magazine. And, please, to expedite the processing of your submission, send the information in a format similar to the product listings below.



Company:
A Osborne/McGraw-Hill
630 Bancroft Way
Berkeley, CA 94710
(415) 548-2805

Product:
6502 Assembly Language Subroutines — Osborne/McGraw-Hill has just released the first title in a new series of Assembly Language Subroutines.

6502 Assembly Language Subroutines by Lance A. Leventhal and Winthrop Saville presents an overview of assembly language programming for the 6502 microprocessor and a collection of more than forty useful subroutines. These routines can be used as subroutines in actual applications and as guidelines for complex programs.

Leventhal and Saville provide code for common routines including: code conversion, array manipulation arithmetic, bit manipulation, summation, sorting and searching. Also included are examples of I/O routines, interrupt service routines, and initialization routines for common family chips such as parallel and serial interfaces, and timers. These subroutines will run on 6502-based micros, including Commodore.

This book, as well as forthcoming books in the series, is aimed at the user who wishes to utilize assembly language immediately. All routines have been debugged, tested and documented.

Price:
$12.99 (paperback —
496 pages)

Company:
Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

Product:
VIC Piper — Turn your VIC into a music machine with VIC PIPER. This program allows you to compose, save, recall and playback music using a standard VIC without any additional hardware!

You enter music by using alpha notation:

AF# C G D

Rests and note duration are easily entered. You can vary the volume and tempo,̀ play harmony, print pictures of text to accompany your music and automatically load and run additional compositions from cassette or diskette.

Includes manual and sample compositions.

Price:
$25.00 (U.S. or Canada)
and $30.00 (foreign)

Company:
Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

Product:
VIC Vigil — VIGIL is a new language for programming interactive games on the VIC. The language includes a set of commands that allow you to use all of the features available on the VIC (color, sound, light pen, game paddles and joysticks).

With VIGIL you can create action packed games that rival machine language coded games in speed, but in a fraction of the time. The commands are easily learned with the 80+ page manual.

You can start out using VIGIL immediately because it comes with these 9 full-length programs — Breakout, SpaceWar, AntAircraft, U.F.O., SpaceBattle, Concentration, Maze, Kaleidoscope & FortuneTeller.

Requires 3K VIC memory expander.

Price:
$35.00 (U.S. or Canada)
and $40.00 (foreign)

Company:
Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

Product:
VIC HiRes — Explore the HiResolution and Multi-Color modes of your VIC with these two useful utilities.

The HiRes utility shows you how to use VIC's HiRes graphics giving you 104 × 152 plot positions without any additional hardware! Plot points, draw lines and boxes and display ASCII text in HiRes.

The Multi-Color utility gives you the same capabilities and an additional color on a 52 × 76 size screen, also without any additional hardware.

Both utilities add the graphic commands to your standard VIC BASIC with minimum overhead but full function.

Includes sample programs and documentation.

Price:
$20.00 (U.S. or Canada)
and $25.00 (foreign)

Company:
Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

Product:
PET Tiny BASIC Compiler — TINY BASIC COMPILER (TBC) supports a floating point subset of the PET BASIC programming language. The compiler reads your program and writes out a file containing 6502 object code that you then load and execute.

The TBC supports all floating point arithmetic and functions that are available in the full PET BASIC language. In fact, you can write, test and debug your program using the built-in interpreter before using the TBC to compile it.

The TBC package will run on all 40-column model PET/CBM's with a minimum of 8K of memory. If you have at least 16K of memory, then there is also a version (included in package) which will give you a full assembly listing of the compiled code.

Price:
$25.00 (U.S. or Canada)
and $30.00 (foreign)

Company:
Briley Software
P.O. Box 2913
Livermore, CA 94550

Product:
RNAV3 Navigator — An air navigational aid is now available to pilots from Briley Software. The computer program uses the Commodore PET/CBM Microcomputers and is called "RNAV3 NAVIGATOR."

It obtains true straight line courses for aircraft having DME equipment and VOR-TO-VOR courses for aircraft without the DME ability.

It performs hundreds of trisometic functions per flight, handles the spherical nature of the earth (polar coordinates), and always searches for the closest radio signal while the flight path is being calculated.

As complicated as the program is, the pilot provides only four items: Flight Option, Waypoint Interval (if DME used), the Departure Coordinates, and the Destination Coordinates. The resulting calculated flight path is then displayed upon the computer's screen and optionally sent to a printer. Each waypoint of the flight contains the closest VORTAC code-name, radio frequency, radial, and distance from path. Each point also displays the magnetic compass bearing, nautical miles traveled, and miles left to complete.

The program currently comes in two versions. One, which fits within the 8K sized PET microcomputer, covers the three Pacific States, Idaho, Nevada, and Arizona. A 16K version covers the Eleven Continental States west of the 102nd Meridian. Other versions covering the rest of the United States are being planned.

The program is available from local Commodore computer dealers or direct from BRILEY SOFTWARE, Box 2913, Livermore, CA 94550-0291.

Price:
$25.00 (8K version)
and $30.00 (16K version)

Company:
Electronic Courseware Systems, Inc.
P.O. Box 2374, Station A
Champaign, IL 61820
(217) 359-7099

Product:
A Planning Guide To Successful Computer Instruction — Electronic Courseware Systems, Inc., has developed a publication to aid teachers and administrators in planning and evaluating the use of computers in the classroom. The new publication, authored by G. David Peters and John M. Eddins, is entitled, *A Planning Guide to Successful Computer Instruction.* The book offers criteria and guidelines for assessing the available computer and microcomputer hardware and software for instructional use. Educational discounts to schools for multiple-copy purchases.

Price:
$19.95

Company:
Electronic Specialists, Inc.
171 South Main Street
Natick, MA 01706
(617) 655-1532

Product:
Self-Retest Power Line Interrupter — Electronic Specialists expands their AC Power Line Interrupter series to include automatic reset models. Should AC Line Voltage be disrupted or exceed pre-set safety limits, the POWER INTERRUPTER disconnects AC power from

→

controlled apparatus. A 4-minute time delay, followed by automatic self-reset, helps avoid wide voltage fluctuations associated with Power Line malfunctions. An optional Line Voltage Monitor is available.

Intended for installations operating unattended for long periods, the SELF-RESET POWER INTERRUPTER provides safety and protection for equipment and personnel.

Connecting to the AC line with a standard 3-prong plug, the SELF-RESET POWER INTERRUPTER can accommodate a 15 amp resistive load or a 10 amp inductive load.

Model PI-SR-15
Self-Reset Interrupter ... $185.95

Self-Reset & Voltage
Monitor Interrupter $205.95

Price:
See description

Company:
Human Engineered Software
3748 Inglewood Blvd. Room 11
Los Angeles, CA 90066

Product:
HESCOM — HESCOM is a machine language program that can transfer data and programs between two PETs, two VICs or a PET and a VIC. You could load into the PET/CBM a program from a disk and transfer it to the VIC at 7000 bytes per second — three times the speed of the disk! After modifying the program on the VIC, you could send it back to the PET for saving to a disk or listing on a printer.

Price:
$49.95 (tape) and
$52.95 (disk)

Company:
Human Engineered Softwaretems, Inc.
3748 Inglewood Blvd. Room 11
Los Angeles, CA 90066

Product:
HESCAT — HESCAT is a complete and fast diskette cataloging system for a PET/CBM, comprised of five programs in BASIC and machine language. You can catalog and uncatalog diskettes and print different reports. Use HESCAT to organize your diskette library. Using a full or partial name, in a few seconds, you can find on what diskette a certain program is on. Sorting is done in machine language. All programs are menu-driven with excellent human engineering.

Price:
$39.95 (disk)

Company:
Info Designs
6905 Telegraph Road
Birmingham, MI 48010
(313) 540-4010

Product:
Order Entry/Point-of-Sale — A module that interfaces with Info Designs Inventory Management and Accounts Receivable systems to provide fully automated invoicing, stockroom picklists, on-line inventory tracking, and automated backorder control and reporting. **A COMMODORE APPROVED PRODUCT.**

Price:
$750

Company:
Info Designs
6905 Telegraph Road
Birmingham, MI 48010
(313) 540-4010

Product:
Time Management/Client Billing System — A comprehensive system to keep track of professional staff time and chargeable expenses by client and matter. The system produces work-in-progress reports by matter, a series of management reports, and provides for automated client billing. **A COMMODORE APPROVED PRODUCT.**

Price:
$2,000

Company:
INI, Inc.
4013 Chestnut Street
Philadelphia, PA 19104
(215) 386-7994

Product:
Client Write-Up System — Journal entries can be edited, deleted, or added to at any time. They are entered and edited utilizing INI's full screen-editing capabilities, and the system checks to make sure that account numbers are valid and that transactions balance.

The user can create reports utilizing any monthly, quarter-to-date, or year-to-date balances from this year, last year, or the budget, including ratios and/or variances, subtotals, account groupings, and many other options. Examples of reports which may be created are any comparative balance sheet or income statement, sales reports, payroll reports, skeletal accounts receivable and payable, and changes in working capital. The system also provides a detailed general ledger report, cash disbursements and receipts journals, and a general journal listing.

The accountant can have the system transfer budgets created with VisiCalc™ (Personal Software) to the system's budget files, in order to produce budget and budget comparative reports.

INI's Client Write-Up System is currently based on the Commodore 8032 CBM microcomputer with 8050 dual disk drive. Two disks are used for each client.

Price:
$850

Company:
Insurance Guidance Corporation
1509 Locust St., P.O. Box 2086
Elkhart, IN 46515
(219) 264-1380

Product:
Insurance Rating Programs — Now insurance agencies can have the ability to figure quotations of

insurance premiums by using a desk-top micro-computer. It contains the rates of the particular agency, customized to agency needs. The turn-key program is useable without any special training in either computers or insurance.

When insurance companies change rates, a program is included so the agency can change their computer by modifying data statements. If other changes are made, the price is $50.00 per company.

Programs are designed for a Commodore Business System. A minimum of 32K memory and disk drive is required, and an 80-column printer is suggested. Prices include research, training and program installation by the hardware dealer. Sales taxes are not included.

Base Price (One-Time Charge): ..	$600.00
Per Company Per Program	$125.00
Each Rating Territory Over 1	$25.00

A deposit of $600.00 is required,

Rating is based upon the manuals provided by the agency. Delivery date will be confirmed upon receipt of order and deposit. Usual delivery period is 60-90 days.

Price:
See description

Company:
Insurance Guidance Corporation
1509 Locust St., P.O. Box 2086
Elkhart, IN 46515
(219) 264-1380

Product:
Micro Accounting — Created fbr independent agencies, this accounting package includes: invoicing, check writing, accounts payable, accounts receivable, customer files, payments, management statistics, general ledger and word processing.

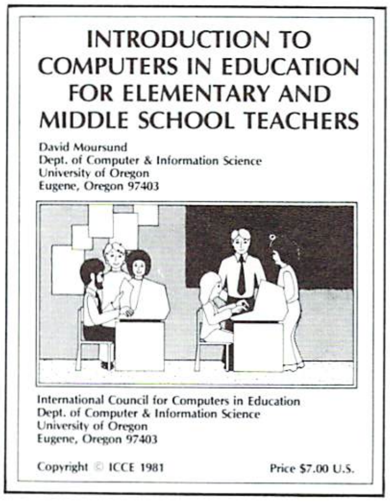
Two packages are available. The first, with a capacity of 1,500 accounts, is $6,585 plus tax. The second, for up to 4,000 accounts, is $8,385 plus tax. This includes a turn-key package of hardware/software, using a Commodore Business System. No customizing need be done, as the packages are complete in themselves. The program is entirely written in Assembler language, so speed of retrieval is excellent.

Software Only: .... (Optional)	$2,000.00
Custom Insurance Rating:	$600.00 Base
Per Company Plan Charge	$125.00
Per Extra Territory	$25.00

Taxes are not included. Installation and training are included at no charge.

Note: Prices are subject to change without notice.

Price:
See description


INTRODUCTION TO COMPUTERS IN EDUCATION FOR ELEMENTARY AND MIDDLE SCHOOL TEACHERS

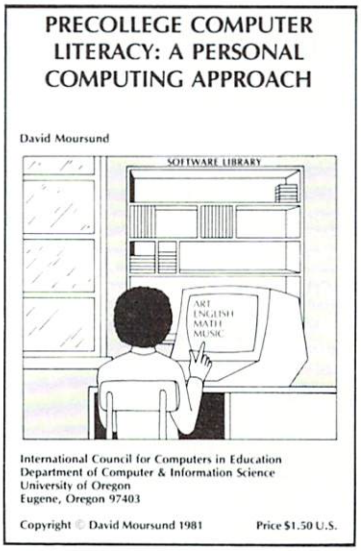
David Moursund
Dept. of Computer & Information Science
University of Oregon
Eugene, Oregon 97403

International Council for Computers in Education
Dept. of Computer & Information Science
University of Oregon
Eugene, Oregon 97403

Copyright © ICCE 1981 Price $7.00 U.S.

Company:
International Council for Computers in Education (non-profit)
Department of Computer and Information Science
University of Oregon
Eugene, Oregon 97403

Product:
Introduction To Computers in Education For Elementary and Middle School Teachers — is a textbook designed for preservice and inservice courses. It is designed to be used in an introductory course for teachers who have had no previous exposure to computers. It is not a computer programming book — indeed, it does not contain any instruction in computer programming. The book suggests that an introductory course would contain considerable hands-on experience and some computer programming, and that there are a number of suitable texts for that part of the course. The book contains a large number of activities for teachers, as well as many activities for elementary and middle school students.


PRECOLLEGE COMPUTER LITERACY: A PERSONAL COMPUTING APPROACH

David Moursund

International Council for Computers in Education
Department of Computer & Information Science
University of Oregon
Eugene, Oregon 97403

Copyright © David Moursund 1981 Price $1.50 U.S.

Price:
$7.00

Company:
International Council for Computers in Education (non-profit)
Department of Computer and Information Science
University of Oregon
Eugene, Oregon 97403

Product:
Pre-college Computer Literacy: A Personal Computing Approach — discusses what is meant by computer literacy and outlines goals for computer literacy instruction. This booklet is designed for teachers and

curriculum leaders who support the idea that all students should become computer literate. It defines computer literacy as a functional, useful level of knowledge and skill, and it points out a number of ways in which being computer literate can make a significant difference in the life of a precollege student.

Price:
$1.50

Company:
Lawrence Hall of Science
University of California
Berkeley, CA 94720
(415) 642-3167

Product:
Creative Play: Problem Solving Activities With The Computer — Now teachers everywhere can introduce students as young as eight years old to computers. And teachers don't even have to be familiar with programming themselves! U.C. Berkeley's Lawrence Hall of Science has announced the publication of *Creative Play: Problem Solving Activities with the Computer,* a software package inspired by the lessons learned in the Hall's after-school "Creative Play" class, which has introduced more than 2,000 eight to eleven year olds to computers since 1974.

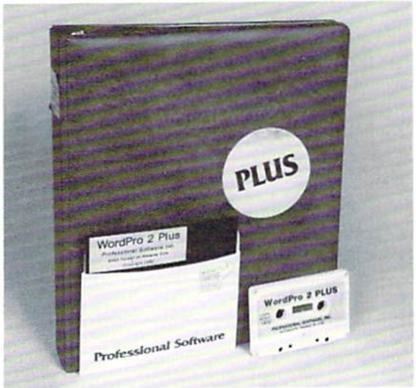
The activities, strategies, and computer programs that make this class so effective have been edited by Batya Friedman and Twila Slesnick into a 67-page book with an accoming twenty-five programs for use with the PET microcomputer. *Creative Play: Problem Solving Activities with the Computer* now makes the techniques developed over the last eight years available to educators everywhere.

The discovery-oriented activities are organized into eight 1½-hour sessions. Emphasis is on the problem-solving skills: logical reasoning, data collection and evaluation, creative thinking, spatial visualization, and pattern identification. The activities in the software package also encourage the children to develop important social skills — cooperation, collaboration, communication, and peer teaching.

Copies of *Creative Play: Problem Solving Activities with the Computer,* including one copy of the diskette for the PET microcomputer, are available for $48 plus California taxes when applicable. (Please add $3 postage and handling per item.) Additional copies of the diskette are available for $10 per copy to those ordering the software package. Orders should be sent to the Math and Computer Education Project, Lawrence Hall of Science, University of California, Berkeley, CA 94720.

Creative Play: Problem Solving Activities with the Computer is just the latest piece of educational computer courseware published by the Math and Computer Education Project. For a free brochure describing courseware available for purchase from the Hall, write to MCEP at the above address or call (415) 642-3167.

Price:
See description



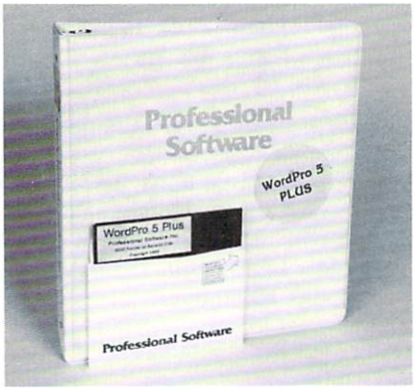
Company:
Professional Software, Inc.
166 Crescent Road
Needham, MA 02194
(617) 444-5224

Product:
WordPro 2 Plus — Professional Software's new WordPro 2 Plus word processing package has now joined the highly successful WordPro family.

One of the major benefits of WordPro 2 Plus is its flexibility. WordPro 2 Plus is compatible with just about any CBM computer available. Almost any current or prospective owner of a CBM system can benefit from WordPro 2 Plus.

WordPro 2 Plus requires a minimum of 16K and is sold complete with *both* cassette and diskette versions and is fully compatible with most CBM computers WordPro 2 Plus will also operate on all Commodore disk drives. A wide range of popular dot matrix and letter quality printers are supported *directly from the program*. Full operational flexibility.

Price:
$199.95



Company:
Professional Software, Inc.
166 Crescent Road
Needham, MA 02194
(617) 444-5224

Product:
WordPro 5 Plus — WordPro 5 Plus creates a word processing system with separate text areas to make word processing easier. Multi-user capability (up to 8 workstations) is available via the addition of a multi-user interface device. WordPro 5 Plus operates with many business oriented programs including DataPlus™, Professional Software's new Information Management program.

WordPro 5 Plus is designed for use on Commodore's 8032 computer with Commodore's 64K Memory Expansion Board installed. Any CBM Disk Drive may be used for document storage and any properly interfaced ASCII letter quality or dot matrix printer may be used.

Price:
$450.00
panying computer diskette contain-

Company:
Retcom Systems, Inc.
61-B Mountain Boulevard
Warren, NJ 07060
(201) 561-3112

Product:
The Retailer — The Retailer provides three main functions: cash register, inventory manager and report generator. As a cash register it prompts the user for: Quantity, Stock numbers, Price, etc. It calculates Sales Tax and/Discount, if applicable. The system also displays an Error Message if the item being entered does not exist in inventory. Daily sales transactions are stored on diskette for subsequent inventory updates.

The Inventory Manager allows for the entry and maintenance of approximately 3,000 stock units. Each stock unit contains information such as Quantity Ordered/Received, Selling Price, Cost, Reorder Level and Stock Balance. The daily sales records are run against the inventory data base updating inventory items and generating daily sales reports.

The Report Generator Module produces a series of reports which provide sales analysis and tracking capabilities.

The Retailer™ has been in actual field use for over one year. It was initially written for a clothing store, but designed with flexibility in mind and is easily adapted to most retail environments.

Price:
N/A

Company:
SLED Software
P.O. Box 16322
Minneapolis, Minnesota 55416
(612) 926-5820

Product:
Special Learning Ed Software — The program consists of individualized computer-based instructional activities in spelling for students with specific language learning disabilities. However, this program will improve the skills of any persons with spelling deficits. The benefits of using the tapes include teaching average students, slow learners, and those who are studying English as a second language. The program can be used by anyone who can read.

The spelling program includes spelling rules, exceptions, and generalizations which provide repetitive exercises and reinforcement as well as motivation to the learner. The drill and practice which persons with specific language learning disabilities require to learn to spell can be provided through tapes to be used on the PET. All tapes work with any 40-column PET, old or new.

The program consists of 24 tapes covering five basic spelling rules essential to encode words. The complete program is also available on disk. The copyrighted program includes:

F-L-S-or Z rule	
4 Tapes	$38.00
Doubling Rule or One, One, One	
4 Tapes	$38.00
Silent E Rule	
5 Tapes	$47.50
The Y Rule	
5 Tapes	$47.50
Extension of the Doubling Rule	
6 Tapes	$57.00

You can SAVE by purchasing the entire series, tapes or disk, for $199.00.

Minnesota residents add 5% sales tax.

Individual spelling rules can be purchased. ■

Price:
See description.

Skyles Electric Works

BASIC Programmer's Command-O™

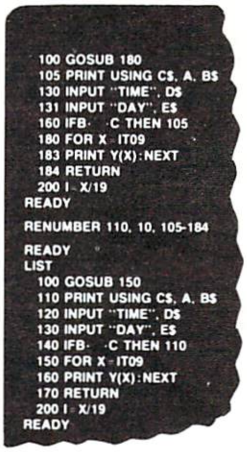
For PET/CBM™ Owners Who Want More Fun And Fewer Errors with Their Programs

Here are nineteen commands you'll need, on a single chip you can install in two minutes without tools, **on any PET or CMB system.** 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off.

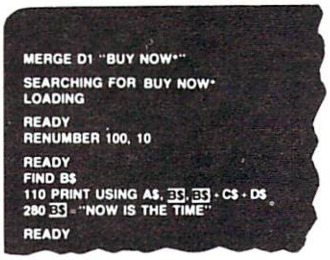
For PET 4016, 4032, CBM 8032, SuperPET or BASIC 4.0 upgrade

BASIC Programmers Command-O™

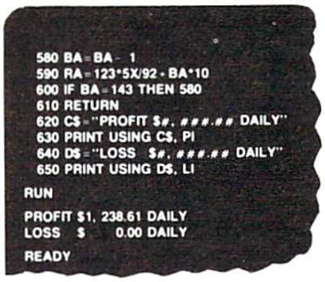
AUTO DUMP DELETE FIND (improved) HELP KILL OFF TRACE (improved) RENUMBER (improved) INITIALIZE MERGE MOVE EXECUTE SCROLL OUT SET SEND PRINT USING BEEP



```
  100 GOSUB 180
  105 PRINT USING C$, A, B$
  130 INPUT "TIME", D$
  131 INPUT "DAY", E$
  160 IFB·   ·C THEN 105
  180 FOR X = IT09
  183 PRINT Y(X):NEXT
  184 RETURN
  200 I = X/19
READY

RENUMBER 110, 10, 105-184

READY
LIST
  100 GOSUB 150
  110 PRINT USING C$, A, B$
  120 INPUT "TIME", D$
  130 INPUT "DAY", E$
  140 IFB·   ·C THEN 110
  150 FOR X = IT09
  160 PRINT Y(X):NEXT
  170 RETURN
  200 I = X/19
READY
```



```
MERGE D1 "BUY NOW*"

SEARCHING FOR BUY NOW*
LOADING

READY
RENUMBER 100, 10

READY
FIND B$
110 PRINT USING A$, B$, B$ + C$ + D$
280 B$ = "NOW IS THE TIME"

READY
```



```
  580 BA = BA - 1
  590 RA = 123*5X/92 + BA*10
  600 IF BA = 143 THEN 580
  610 RETURN
  620 C$ = "PROFIT $#, ###.## DAILY"
  630 PRINT USING C$, PI
  640 D$ = "LOSS  $#, ###.## DAILY"
  650 PRINT USING D$, LI

RUN

PROFIT $1, 238.61 DAILY
LOSS  $    0.00 DAILY

READY
```

➧ NOTICE ➧
COMMAND-O IMMEDIATELY AVAILABLE
FOR
ALL MODELS PET/CBMs
PLEASE SPECIFY !!
➧ NOTICE ➧

INSTALLATION:

Installs into socket UD12 ($9000) in "fat forty's", 8032's or SuperPET's. Installs into socket UD3 ($9000) in BASIC 4.0 upgrades.

PRICES:

For PET 4016, 4032, BASIC 4.0 upgrade COMMAND-O CO-40N	$75.00
For CBM 8032, SuperPET (SP9000) COMMAND-O CO-80N	$75.00

Shipping and handling $2.50 USA/Canada, $10.00 Europe/Asia

California residents please add 6% or 6-1/2% sales tax as required

*order your **Disk-O-Pro,** today*

AVAILABILITY: Immediately from your LOCAL DEALER

or

VISA, MASTERCHARGE ORDERS CALL (800) 227-9998 (except California residents)
CALIFORNIA ORDERS PLEASE CALL (415) 965-1735

Skyles Electric Works

231E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

The Print Mint

by
Jim Butterfield, Toronto

The usual point of programs is that they produce output. The normal way of producing output is by using the PRINT command or its cousin, PRINT#. We can abbreviate PRINT with a question mark (?), but oddly enough, PRINT# can't be shortened that way: typing ?# will produce a program line that lists as PRINT#, but doesn't work.

Other ways

We can generate output without using PRINT. It's not always good practice, but we can POKE to the screen memory area. This can be good for graphic games and animations, but there are several bonus things we get by using PRINT. First, PRINT keeps track of the line position for us, and starts a new line as necessary. Secondly, PRINT doesn't limit output to the size of the screen. When the screen fills up, scrolling is automatic. Finally, and most important, PRINT can easily be changed to PRINT# to allow output to be directed to other devices such as printer, modem, disk or cassette tape. In contrast, screen POKEs are absolutely limited to the size of the screen, and can't be easily redirected anywhere else.

Punctuation

If you say 'PRINT X' you will print the value of X and start a new line. The absence of punctuation at the end of the PRINT command signifies "That's the whole thing, print it and wrap the line up." In contrast, if you say 'PRINT X;' you will print the value of X but you won't go to the next line. The invisible cursor will wait behind the printed value. This sounds a little backwards: you do something extra if you have no punctuation, but you do nothing if you have a semicolon.

There's one other form of "formatting" punctuation—the comma. If you type 'PRINT X,' you will print the value of X and then skip ahead to the next "column." Columns are considered to start at positions 11, 21, 31 and so on up to position 71. They exist only on the screen; saying 'PRINT#4, X,' to send to printer or other device won't set up columns properly. The comma will produce quick and convenient output to the screen, but it may be a bad habit since you can't use it anywhere else.

We can use this punctuation within a BASIC PRINT statement as well as at the end. 'PRINT A;B$;C;' will generate the values of variable A, string B$, and variable C one behind the other and will leave the cursor positioned behind the value of C.

Neat input

We can use this punctuation to generate prompting for INPUT statements. For example, if we wanted to add ten numbers, we might code:

```
100 PRINT "INPUT EACH NUMBER:"
110 FOR J = 1 TO 10
120 PRINT J;
130 INPUT X
140 T = T + X
150 NEXT J
160 PRINT "TOTAL IS";T
```

We prompt for the ten numbers with 1? . . . 2? . . . 3? . . . and so on. The prompting number is printed by line 120—J is stepping from 1 to 10—and the question mark from the INPUT statement appears behind it because line 120 ends in a semicolon; after printing the number we wait on the same line so that the question mark will appear there. Question: What would happen if line 120 ended with a comma instead of a semicolon? Try it and see.

Number formats

Numbers are printed in a special format. First, there is either a space for positive numbers or a minus sign for negative numbers. Then the number appears, as many digits as required plus a decimal point if needed and perhaps even "E" notation. (Never heard of E notation? Try PRINT 3E2 and see if you can figure it out.) Finally, the number is followed by a cursor-right on the screen.

This seems at first to give you two spaces between numbers, but there are one or two fine points that are useful to know. If you type PRINT 2;3;4 you will see two spaces appear between each set of digits. Now try this: Type a bunch of x characters over the answer (a row of xxxxxxx . . .) and then cursor back to the PRINT statement and press RETURN again. Some of the x's don't go away. That's because a cursor-right skips over that part of the screen without writing there.

There are a couple of ways to eliminate this difficulty if it bothers you. If you change a value to a string before printing, the cursor-right won't be performed. You could type PRINT STR$(2);STR$(3);STR$(4)—the same numbers will print with at least part of the problem solved. If you happen to have an 80-column or Fat-40 4.0 system, you may type: PRINT CHR$(16);CHR$(22);2;3;4 and you'll discover the problem is solved quite elegantly.

Here's another exception to the two-spaces rule: Type PRINT 2;-3;-4;5 and look at the result. The minus signs take up one of the two positions, and now there's only one space between some numbers.

Summary

PRINT is handy and versatile. It takes a little while to get used to the formatting of the PRINT statement, but you'll soon have good control over your output.

There are some fascinating things you can PRINT which cause the screen to do unusual things. More about them another time.

Pretty printing

When you are producing output, it's good to make it neat. The computer is there to help its human readers, and the more you can do to improve the information, the better job you'll be doing.

BUTTERFIELD ON COMMODORE

Printing in columns

Beginners often arrange values in columns by using the screen tabulation functions: putting a comma into the PRINT statement, or using the TAB function. These methods work, but they have a pitfall—they won't behave properly if the output goes to other devices. The problem is that the computer always knows exactly where the screen cursor is, but it never knows on what column the external devices are located. It doesn't even try to keep track, so a TAB or a comma directed to the printer or other device won't behave properly.

It's my feeling that almost everything that goes on the screen can be usefully directed to the printer, or written to a disk file with a view to transferring to the printer later. Once you have a report looking nice on the screen, you don't want to have to reprogram to get it looking nice in print. So . . . stay away from TAB and commas—there's a better way.

Redirecting output

While I'm on the subject of switching output from the screen to the printer, I'd like to share a little coding trick with you. Most programmers know that you can direct output to a printer by performing an OPEN to device number 4 (the printer) and then using PRINT# . . . That's fine for a finished program, but you can waste a lot of paper while you're checking out a program if you do everything to the printer.

Here's the trick: We can OPEN to device number 3 (the screen) and PRINT# to the screen, checking our program and fixing it up. When it's ready to go, all we need to do is to change the OPEN statement so that it names device number 4, and output goes to the printer. We save time and paper. Let's try it. Code:

```
100 OPEN 1,3
110 FOR J=1 TO 10
120 PRINT#1,J;SQR(J)
130 NEXT J
140 CLOSE 1
```

When we run this program, output is delivered to the screen. If everything looks good, we can now change line 100 to OPEN1,4 . . . and output is redirected.

It's not really a trick; it's good coding. We could allow the user to specify what output he wanted by coding something like: 100 INPUT"DEVICE NUMBER";N:OPEN 1,N so that the user could type in 3 or 4 to select the type of output he wants.

Neatness counts

If I'm sternly discouraging TAB and the comma, how can you arrange things in columns? A few simple answers, but first some ground rules. The best way to arrange stuff in columns is to make sure that each "field" is always the same length. Then, each item will be printed neatly in the same place across the page.

How can we rechop two numbers as different as 3 and −32768 so that they occupy the same space? For that matter, how can we take two names as different as BUTTERFIELD and NG and make them the same length?

Let's take the names first. These "strings" could be neatly chopped down to a fixed length by means of the LEFT$(function . . . if they were long enough. For example, we could slice out the first eight characters of string X$ with LEFT$(X$,8); but it won't work if X$ is less than eight characters long in the first place. So—pay attention—we must first pad out the name by adding spaces to the end. Sticking extra characters onto the end of a string is called "concatenation"—pronounced with emphasis on the cat—and is done with a plus sign. If we had a short name like M and wanted to tack eight spaces on the end, we'd do it by writing "M"+"   " which would create a new string nine characters long. A name like BUTTERFIELD treated the same way would end up nineteen characters long, but this doesn't matter; we're going to chop them both down to the same length with LEFT$(.

Let's put it all together. If the name is held in variable N$, we code PRINT LEFT$(N$+"   ",8); with a semicolon at the end. First we concatenate, adding the spaces, then we chop (or "truncate"), cutting to a fixed length. Finally, we print. Both long and short names will be printed as exactly eight characters. The next thing we print will be neatly lined up behind it. We might want to make the field more than eight characters long, since a splendid name like BUTTERFIELD would end up chopped to BUTTERFI—if we do increase the length we must remember to add more spaces, of course.

The above procedure is called left justification, since the strings are lined up neatly on the left with spaces filling out the right hand side. We can go the other way and produce right justification with a small adjustment. Try PRINT RIGHT$("   "+N$,8); and you'll see how the left side fills with spaces and names line up on the right. This is the kind of alignment you will want with numbers. We'll deal with that in a moment. Remember that if you don't allow enough space you'll end up with chopped-off names like TERFIELD, and there's no justification for that.

If the numbers you are using are integers, you'll usually want to line them up with right justification. Once again, this is easy to do once you know the function that changes numbers to strings. If your value is held in variable X, we can change it to a string with STR$(X). Now we can do the right justification with PRINT RIGHT$("   "+STR$(X),6); everything will work out neatly. Study this statement and see how X builds up into a neatly justified string of length six.

If your numbers contain fractional values, you may want to try to line up the decimal points. That's much more challenging. Perhaps you'd like to try your hand at it. We'll tackle it here another time. ■

THE COMPUTER CAN BE THE MOST DIFFICULT OF ALL EMPLOYEES TO TRAIN

We Train Micro Computers To Work For Business and Professional Firms

We believe we do it best as far as the folks who manage organizations are concerned. At least more business and professional firms come directly to us for management information systems than any other micro computer company in America. We produce quality products that are management oriented in a cost effective manner. We actually support what we produce and we conduct training. We even meet deadlines. Perhaps most important, we listen carefully and we do our best to meet your needs.

We have developed business and professional programs for firms with two or three employees to firms the size of Lenox Industries. Geography presents few problems for us. We have customers in all fifty states and Canada.

May We Train A Micro Computer For You!

MAG

Merging computer technology with the practical worlds of business and government

Corporate Central
Post Office Box 346
493 East Clayton Street
Athens, Georgia 30603
404/353-8090

Atlanta Division
4325 Memorial Drive
Suite H
Atlanta, Georgia 30032
404/294-8000
(Just inside I-285)

BITDIDDLING

Be It Ever So Humble...

A Commodore dealer in Stanwood Washington takes the term "home computing" very seriously. It seems that Kirk and Mary Shroyer, who own Computer Corner, have made their "home" into one of the top Commodore dealerships in the entire Northwestern region in only a few months. A spare bedroom serves as a storage room and an extra living room is their show and demonstration area.

Kirk, citing the biggest advantage of low overhead, said he would recommend selling from the home to anyone just starting out. The Shroyers do all the work themselves including most service and all training. Recently, Kirk sold six CBMs and three PETs to a local high school, then set up a timesharing system, so all the computers could share an 8050 disk drive storage device and the teachers unit could "electronically look over the shoulder of every student in the class."

But alas, success is spoiling the Shroyers, and they will probably move into a "real" store soon. Still, with this hardworking innovative couple it probably wouldn't matter where they sold Commodore Computers.They could probably sell VICs from a broom closet.

VANTASTIC!

In Albany Oregon there is a dealer who has computers to go! Sirco Business Computers has two vans with Commodore business systems set up on computer tables. Sirco will drive to a prospective customer's office or home and demonstrate the system right there. Jim Cotton of Sirco said that "hands-on demonstrations are very important, and it really impresses customers to see the systems in their own environment."

There have been *almost* no problems with these "stores on wheels." On one occasion, an improperly assembled computer table fell apart in the van, resulting in a slightly damaged Commodore Business System.

Sirco by the way stands for the Second Industrial Revolution Company. Cotton said that he believes that computers represent a revolution as important as the industrial one. "Commodore is the flagbearer in the revolution that is changing the way the whole world is doing business." he said. And it's innovative dealers like Sirco who are helping Commodore lead the way.

If We Could Talk to the Animals...

On the big screen, actor Rex Harrison portrayed Dr. Doolittle and was known for talking to the animals. Commodore Magazine, not to be outdone by Hollywood, has found a subscriber who lives on Doolittle Road in Preston, Connecticut. His name? Dr. Walter Doolittle. His occupation? A veterinarian, of course. This Dr. Doolittle may not talk to the animals but he's using Commodore computers to help manage his practice.

Doolittle said he takes a lot of kidding about his name but it's great for business. He is so popular that his clients not only bring their own children to meet Dr. Doolittle but also bring their neighbor's children. "We have a pony named Cocoa to amuse the children so that they won't be in the way," said Doolittle.

Doolittle is extremely pleased with his 8032 and peripherals. He said that the whole family enjoys using them including his son in first grade. Using his 8032, Doolittle has prepared a form letter to inform hundreds of clients that a new vaccine for cats is available, pending government approval. "The Commodore is ready but now we have to wait for the government," said Doolittle.

Computer Widow?

The Football season's over, tis true,
But my gender may as well be neuter,
My husband discovered his best Christmas present,
Is his fascinating PET home computer.

Thanks to Marilyn (Mrs. William) Cohen of Evanston, IL for sharing her "computer widow's" lament.

Unsung Heroes

Ever wonder where all those Commodore microcomputer innovations get their start? Shown here are just some of the Commodore's Research and Development group, who have been responsible for many of the most important advances in the entire computer industry. Some of their recent accomplishments include the VIC 20, the COMMODORE 64, the ULTIMAX and other new projects that are bound to shake up the entire industry. Pictured here, from left to right are: (back row) Yash Teradura, Tim Martin, Rich Lawrence, Bob Yannes, and Mike Angelina; (front row) Jim Redfield, Steve Wyzykiewicz, and Design Manager Al Charpentier.

COMMODORE RETAIL PRICE LIST

Effective April 1, 1982

PRODUCT	DESCRIPTION		RETAIL PRICE
COMMODORE CONSUMER COMPUTER RANGE — VIC SERIES			
VIC 20™	Full-featured, expandable color computer system. 5K RAM. Includes RF Modulator & TV switch box.		$ 299.95
C2N DATASETTE RECORDER	Cassette Storage For PET/CBM/VIC.		$ 75.00
VIC 1515 GRAPHIC PRINTER	VIC dot matrix printer. 30 CPS, 8" paper; prints full VIC character set; tractor feed.		$ 395.00
VIC 1540 SINGLE DISK	Single-drive intelligent 5¼" floppy disk system for the VIC 20. Media compatible with CBM 2031 and 4040 diskettes. (170K).		$ 599.00
VIC 1600 VICMODEM	Telephone Interface Cartridge. Lets VIC owners communicate by telephone, access telecomputing and timesharing services. Includes VICTERM I program on tape. Plus FREE Compuserve Subscription.		$ 109.95
COMMODORE PERSONAL COMPUTER RANGE — 4000 SERIES			
PET®	System includes graphic keyboard and numeric keypad; 12" display/40 characters.	4016N 4032N	$ 995.00 $1295.00
COMMODORE BUSINESS COMPUTER RANGE — 8000 SERIES			
CBM 8032B™	Typewriter-style keyboard, numeric keypad, 80 column x 25 line display. 32K RAM, BASIC 4.0.		$1495.00
CBM 64K™ MEMORY EXPANSION BOARD	64K Add On-Memory. Expands CBM 8032 to 96K RAM.		$ 500.00
COMMODORE ADVANCED COMPUTER RANGE — 9000 SERIES			
SuperPET COMPUTER SP9000	Enhanced 8032 with additional 6809 microprocessor; total 134K memory with 96K RAM. Languages include Waterloo microBASIC, Waterloo microAPL, Waterloo microPascal, Waterloo microFORTRAN interpreters and Waterloo 6809 Assembler.		$1995.00
COMMODORE SYSTEMS PERIPHERALS			
DISK DRIVES			
CBM 2031™ SINGLE DRIVE	Single drive intelligent 5¼" floppy disk system. 170K (DOS 2.6).		$ 695.00
CBM 4040™ DUAL DRIVE	Dual drive intelligent 5¼" mini-floppy disk system. 340K (DOS 2.1).		$1295.00
CBM 8050™ DUAL DRIVE	Dual drive intelligent 5¼" mini-floppy disk system. 1 Megabyte (DOS 2.5).		$1795.00

PRODUCT	DESCRIPTION		RETAIL PRICE

COMMODORE SYSTEMS PERIPHERALS

DISK DRIVES (Continued)

PRODUCT	DESCRIPTION		RETAIL PRICE
CBM 8250 DUAL DRIVE	Dual drive intelligent 5¼" floppy disk system. 2.1 Megabyte (DOS 2.7).		$2195.00
D9060 D9090 HARD DRIVE	Fast, high capacity; single 5¼" "Winchester" drive. Intelligent hard disk system; 5.0 or 7.5 Megabyte capacity. (DOS 3.0).	5.0 MB 7.5 MB	$2995.00 $3495.00

All Commodore disk units feature compatible DOS command language and file handling capabilities to allow upward expansion of systems as required.

PRINTERS

PRODUCT	DESCRIPTION	RETAIL PRICE
CBM 4022 PRINTER	Dot Matrix printer with tractor feed, 80 column, prints full PET graphics, variable line spacing, bi-directional, 80 CPS.	$ 795.00
CBM 8023P PRINTER	Dot Matrix printer with tractor feed, 136 column. Fully programmable graphics, bi-directional, pseudo letter quality, 150 CPS.	$ 995.00
CBM 8300P PRINTER	True letter quality, daisy wheel printer with 10, 12, 15 pitch or proportional spacing and optional tractor feed. Bi-directional; 40 CPS; supports APL & foreign language print wheels.	$2250.00

ADDITIONAL PERIPHERALS

PRODUCT	DESCRIPTION	RETAIL PRICE
CBM 8010 MODEM	High Performance 300 BAUD IEEE interfaced modem.	$ 280.00
CBM 4010 VOICE SYNTHESISER	Features phoneme syntheses for vocabulary construction. User port interface.	$ 395.00

CABLES

PRODUCT	DESCRIPTION	RETAIL PRICE
IEEE-IEEE	Designed to connect more than one peripheral to any PET/CBM computer.	$ 49.95
PET-IEEE	Use for connecting one peripheral to any PET/CBM computer.	$ 39.95

SPECIAL DOCUMENTATION

PRODUCT	DESCRIPTION	RETAIL PRICE
COMMODORE ENCYCLOPEDIA	A catalog of over 1000 software programs, books, and interfaces from Commodore and third parties.	$ 9.95
EDUCATION MARKETING RESOURCES BOOK	185 Public Domain Programs for Education including 4 disks of software. Details of Users' clubs, educational vendors, and educational applications. (380 pages).	$ 26.95
VIC 20 PROGRAMMER'S REFERENCE GUIDE	VIC 20 reference manual with information on VIC BASIC, 6502 Machine Code Programming, Input/Output ports, VIC microprocessing chips, and tips for all levels of programmers.	$ 16.95

PROJECTIONS & REFLECTIONS

Well it's that time again when I get to jot down a few notes on what's happening on the software scene here at Commodore. ALOT! We have just put the finishing touches on our quarterly software bulletin to the regions and our dealers, and find that we now offer 22 products to the market under our name. The newest additions to the list are some more professional vertical market products along with two development tools. Our Approved Products program is expanding as well, and we expect to publish a separate catalog this summer giving these products the recognition that they deserve among the software community. Currently, we are getting our public domain software offerings together. This has proven to be quite a task, as we are using our best efforts to first qualify that the software is indeed 'public domain' and then we must clean, catagorize and collate these programs into the appropriate volumes for distribution to the public. I expect this offering to cause quite a stir among the current Commodore owners, and hopefully will encourage some new friends as well. How would you loke to be able to go into one of our dealers, purchase a new CBM and be able to receive several hundred free programs, utilities, games and general what-not? Pretty neat, huh? (or for our friends in Canada, Pretty neat, eh?)

Software generation for our new products is underway as well. We have also been holding seminars for outside software houses who want to support our products. I am very excited about these machines because they represent a quantum leap for Commodore in hardware and software integration. The new ULTIMAX is literally a new era in household game machines, that is actually a three-for-one deal. A complete game machine, (including 16 colors, grey for shading, 8 levels of sprites with priority and collision detect), a music generator, (only the best sound chip in the industry), and it's programmable too! BUY, BUY ULTIMAX — BYE, BYE, BYE ATARI. The new COMMODORE 64 is again quite a machine, with all the color and sound type things from the ULTIMAX and 64K to boot. This product is truly a cross-over product in that it will accommodate the cartridges from the ULTIMAX *and* offer the small businessman a cost effective solution as well. In the native mode the machine offers approximately 52K work space, and in BASIC the user has a 40K program area. To accompany the new COMMODORE 64, we will have a companion processor board to support CP/M as an operating system, PET program emulators to accept and execute existing software, UCSD Pascal as a second language, and several other little things that have us all very busy. The COMMODORE 64 will also utilize VIC peripherals and current PET/CBM peripherals via an IEEE cartridge.

Software generation for both machines will permit us to bring to market with the hardware some very versatile, ready to use products in the game and personal areas. Imagine the new Commodore hard disks attached to the COMMODORE 64. How's that for upward compatibility? It has been said about some of our hardware competitors that they have recently brought things to market that were both new and good. But unfortunately for them what is good is not new, and what is new is not very good. We are able to say, that with these announcements what is new is really unique and truly great. ■

—Paul Goheen
Software Product Manager

FILL OUT AND MAIL TODAY

THE MICROCOMPUTER MAGAZINE

commodore

Name____

Address ____

City________________ State ________________ Zip __________

Address Change □ Enter new address above & enclose present mailing label

Renewal Subscription____________New Subscription____________

Enclosed is a check or money order for $ ____________ ($15.00 for six issues)
(Canadian and foreign $25.00)

for____________ issues

Make Check payable to: **COMMODORE BUSINESS MACHINES, INC.**
681 Moore Road, King of Prussia, PA 19406
Attn: Editor, Commodore Magazine

JINSAM

Data Manager selected by NASA, Kennedy Space Center

SAVE TIME! SAVE MONEY!
JINSAM saves taxpayers
100's of thousands of $

OFFICIALLY APPROVED BY COMMODORE

"Much more powerful than you can imagine!"
Robert Baker
Kilobaud Microcomputing

JINSAM is an integrated system. It makes it easy to use your information to its fullest. No more will hundreds of valuable hours be spent searching or analyzing needed information nor re-entering information for various reports.

JINSAM transforms your desk-top computer into the "state of the art" data processing machine with features and accessories found nowhere, even at 10 times the price. NASA, Kennedy Space Center selected JINSAM 8.0 and saved approximately $95,000 over other software/hardware costs. Riley County, Kansas also selected JINSAM 8.0 and saved approximately $90,000 over other software/hardware costs.

JINSAM is designed for you. It is forgiving. It has help commands for every option, available at the touch of a button. The amount of information you store, its structure and/or your hardware can change but your data won't have to be re-entered. Recovery utilites are included even for catastrophes, security passwords are built in for privacy, simple editing and entry includes auto recall, and deleting records is easy and the space is reclaimed. JINSAM includes TWO FREE accessories for reports and labels. You have unlimited report formats with summing and lined up decimals and the label printer prints up to 5 across – any size and even allows single envelopes or index cards.

JINSAM 1.0 allows fast and easy file handling, manipulation and report generation for any CBM computer with CBM 2040 disk drive. It features a menu for ease, has encrypted passwords, 3 deep sorts, .5 to 3 second recall.

JINSAM 4.0 for CBM 4000 series adds JINSORT, user accessible machine sort of 1000 records in 15 seconds; compaction/expansion of information, automatic list maintainance, unlimited number of fields, unlimited record length and much more.

JINSAM 8.0 for CBM 8000 series has all 4.0 features plus unlimited sort, horizontal format, and search by key or record number.

JINSAM 8.2, NEW FOR '82 expands 8.0 capabilities by adding information search by word, key or record number and machine language print, format and manipulation routines.

★ CUSTOM DATA FILES
★ CUSTOM REPORTS/LABELS
★ KEYED RANDOM ACCESS
★ FAST/EASY/MENU DRIVEN
★ MULTIPLE SEARCH KEYS
★ PRIVACY ACCESS CODES
★ WILD CARD SEARCH

JINSAM EXECUTIVE version (soon to be released) is our most powerful professional system for the CBM 8000 and 9000 series. Executive will have 8.2 extended features plus allow multiple users with in-use lockout protection, executive command files, automatic math relations, join, merge or link files, greatly increased record capacity and machine information search by word, as well as by key or record number and many, many more features.

There are currently 7 more interfacing modules – and more under development, including independent interfaces between JINSAM and business packages for your G/L, A/R, A/P needs. We announce the availability of modules and enhancements in JINSAM's quarterly newsletter.

WORDPROPACK - Intelligent interface for WordPro 3, 3+, 4, 4+, creates lists of information from JINSAM files. It allows up to 10 conditions based on each item of information. Produce individualized letters, report cards, special reports, checks, invoices, etc.

MULTI-LABEL - Prints multiple labels per record with up to 2 lines for messages and consecutive numbering. Produce inventory, caution labels, bulk mail labels, etc.

MATHPACK - global calculator/editor +, −, *, /, by another field or constant; null (remove contents) of a field or replace contents of a field with any word, number or phrase. Sum multiple fields in each record or running sum of single field in all records. Extract information or effect permanent change. Replace in the same field or place in a waiting field.

DESCRIPTIVE STATPACK - Determine MEAN, MEDIAN, MODE, STANDARD DEVIATION, VARIANCE, RANGE. Generate HISTOGRAMS from 1 to 25 steps, and produce Z-SCORE reports.

ADVANCED STATPACK - (You must also acquire DESCRIPTIVE STATPACK) Generate one, two or three way CROSSTABS (number of occurance) CHI SQUARE, LINEAR REGRESSION, with graphic representation and prediction, LINEAR CORRELATION and SIMPLE ANALYSIS OF VARIANCE.

CALCPACK - 2 way interface to VisiCalc or any user program. It lets you use VisiCalc for complex manipulation, editing, placing results in JINSAM for sorting, storing or moving data to WordPro as well as giving the ability for exchange with your own applications.

INTERAC - Interface which can read VisiCalc files, WordPro files and almost any sequential files to build JINSAM databases automatically. For example: You could "download" information on holerith cards to sequential files and INTERAC would place them into JINSAM files.

All accessories are accessed thru the JINSAM menu and require security password to gain entrance.

JINSAM gives you FREEDOM OF CHOICE. Start with JINSAM 1.0 and upgrade hardware and data at any time. Choose from accessories at any time. The JINSAM Newsletter brings the latest updates, user input and uses and keeps an eye on the future.

JINSAM stands alone by placing "a lot of potential computing power in one integrated program package" (Fred Klein, Ferson div. of Bausch & Lomb). "The JINSAM package is justification for buying a system no matter what the hardware, be it Vector or Commodore or whatever the system" (Larry Colvin, Micro Computer Systems). It is a "grandfather" in this young field since the first JINI MICRO database system has been marketed since 1979. JINSAM EXECUTIVE will be the third generation in development. All JINSAM systems are sophisticated and flexible yet easy to use. JINSAM is saving its users valuable time and money in government, educational and research institutions, business and industry worldwide.

JINSAM is a Commodore approved product. See your local dealer for a demonstration.

JINSAM is a trademark of JINI MICRO-SYSTEMS, Inc.
WordPro is a trademark of Professional Software, Inc.
VisiCalc is a trademark of Personal Software, Inc.
CBM is a trademark of Commodore Business Machines, Inc.

JINSAM Data Manager

.... Additional Information and nearest dealer
.... Newsletter Subscription ($5 US/$8 Worldwide)
.... JINSAM Demo Disk ($15, plus shipping & tax)
.... User's Guide ($40.00, plus shipping & tax)

Please send to:

Name ____________________

Position ____________________

Company ____________________

Address ____________________

City, State, Zip ____________________

Phone (   ) ____________________

Computer, Disk ____________________

Application ____________________

JINI MICRO-SYSTEMS, Inc.
Box 274-CI • Kingsbridge Station
Riverdale, New York 10463
(212) 796-6200

Dealer Inquiry Welcome

Commodore Business Machines, Inc.
681 Moore Road
King of Prussia, Pa. 19406

address correction requested