OCTOBER, 1982
ISSUE NUMBER 50

PRICE U.S. $2.95
£1.50

THE ORIGINAL MAGAZINE FOR TRS-80™* OWNERS

H&E COMPUTRONICS INC.

Cover Photo by Harry Peterson

Tired Of Your GENERAL LEDGER?

VERSA-LEDGER II

★ THE ULTIMATE PERSONAL CHECK REGISTER

★ A PROFESSIONAL ACCOUNTING SYSTEM

★ A PERSONAL FINANCIAL MANAGER

★ A SMALL BUSINESS ACCOUNTING SYSTEM

★ A COMPLETE GENERAL LEDGER

FOR YOUR TRS-80* MODEL I, II, III or 16 • APPLE II* • I.B.M.* • XEROX* • ATARI* OSBORNE* • ALL MICROSOFT BASIC COMPUTERS

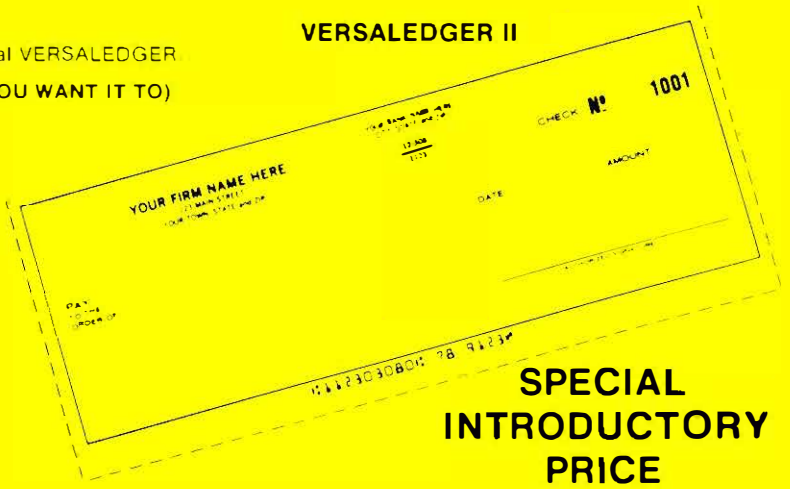
*TRS-80 is a trademark of Tandy Corp. - *APPLE is a trademark of Apple Corp. - *I.B.M. is a trademark of I.B.M. Corp. - *XEROX is a trademark of Xerox Corp. *ATARI is a trademark of Atari Inc. - *OSBORNE is a trademark of Osborne Corp.

HOW IT WORKS . . . .

VERSALEDGER II is a complete accounting system that grows as you or your business grows. To start, your VERSA-LEDGER II acts as a simple method of keeping track of your checkbook. Just enter your check number, date and whom the check is made out to. As you or your business grows, you may add more details to your transactions . . . . . account number, detailed account explanations, etc.

- VERSALEDGER II is an expanded and enhanced version of our original VERSALEDGER.
- VERSALEDGER II gives you an instant cash balance at anytime. **(IF YOU WANT IT TO)**
- VERSALEDGER II can be used as a small personal checkbook register. **(IF YOU WANT IT TO)**
- VERSALEDGER II can run your million dollar corporation. **(IF YOU WANT IT TO)**
- VERSALEDGER II prints checks. **(IF YOU WANT IT TO)**
- VERSALEDGER II stores all check information forever. **(IF YOU WANT IT TO)**
- VERSALEDGER II can distribute one check to multiple expense accounts. **(IF YOU WANT IT TO)**
- VERSALEDGER II can handle more than one checkbook. **(IF YOU WANT IT TO)**
- VERSALEDGER II can be used to replace a general ledger. **(IF YOU WANT IT TO)**
- VERSALEDGER II prints balance sheets and income statements. **(IF YOU WANT IT TO)**
- VERSALEDGER II can interact with any of the other four modules of the VERSABUSINESS SYSTEM.

VERSALEDGER II



SPECIAL INTRODUCTORY PRICE

$99.95

($199.95 after Dec. 31, 1982)

- VERSALEDGER II comes with a professionally written manual (created for the first-time computer user), which includes sample report print-outs.
- VERSALEDGER II HAS AN ALMOST UNLIMITED CAPACITY . . . .
  - (300 checks per month on single density 5¼" disk drives such as the TRS-80 Model I)
  - (500 checks per month on the Apple II)
  - (2400 checks per month on the TRS-80 Model III)
  - (6000 checks per month on the TRS-80 Model II)
  - (3000 checks per month on single density 8" CP/M)
  - (almost unlimited capacity on hard disk drive)
- VERSALEDGER II OUTPERFORMS ALL OTHER COMPETITIVE SYSTEMS NOW AVAILABLE TO MICROCOMPUTER USERS, AT A FRACTION OF THE COST (AND WE OFFER A 30-DAY MONEY BACK GUARANTEE TO BACK UP OUR CLAIM!)

— CAN BE USED WITH 1 or MORE DISK DRIVES with 48K —

ALSO AVAILABLE: VERSARECEIVABLES, VERSAPAYABLES, VERSAPAYROLL, and VERSAINVENTORY

COMPUTRONICS

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

★ ADD $3.00 FOR SHIPPING IN UPS AREAS
★ ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
★ ADD $5.00 TO CANADA AND MEXICO
★ ADD PROPER POSTAGE OUTSIDE U.S., CANADA & MEXICO

MasterCharge VISA AMERICAN EXPRESS

24 HOUR ORDER LINE

(914) 425-1535

NEW TOLL-FREE ORDER LINE

(OUTSIDE OF N.Y. STATE)

(800) 431-2818

ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE
DELIVERY SUBJECT TO AVAILABILITY

PUBLISHER
Howard Y. Gosman

BUSINESS MANAGER
Steven M. Kahan

EDITOR-IN-CHIEF
Hubert S. Howe, Jr.

BUSINESS EDITOR
Peter Shenkin

MANAGING EDITOR
Martin Leffler

CONTRIBUTING EDITORS
Leo M. Conrad
Richard Kaplan
Spencer Koenig
Joseph Rosenman
Gordon Speer
A. A. Wicks
Steven M. Zimmerman, Ph.D.

ADVERTISING DIRECTOR
Kevin Rushalko

SALES MANAGER
Cathleen McGillicuddy

ART DIRECTOR
Edmund Khaleel

OFFICE MANAGER
Beatrice Kahn

SOFTWARE MANAGER
Darlene Bell

CUSTOMER SERVICE
Robert Williams

INVENTORY CONTROL
Michael Bernstein
Michael Wiseltier

SHIPPING
Joan Gentry
Al Pizzo

PRODUCT DEVELOPMENT
Steven Kaplan

PRODUCTION
Adele Damiano
Louise Ann Kerins
Sheryl Streim

MARKETING MANAGER
Andrew Hofer

PROGRAMMING MANAGER
Nancy Rhodes

OCTOBER 1982 **ISSUE NUMBER 50**

CONTENTS

FEATURES

14 Program Previews ..... A. A. Wicks
This Month: E/Z Screen

17 Array of Hope for Basic Programmers (Part 2) ..... Arne Rohde
Searching techniques

20 Four BASIC Programs ..... Gordon Speer
Kaleidoscope, Gascost and Permile, Balloon

24 Assembly Language for Beginners (Part 8) ..... Joseph Rosenman
This month: Let's print the alphabet!

26 Practical Business Programs ..... S. M. Zimmerman and L. M. Conrad
This month: Cash flow analysis for real estate and other investments

42 Program Conversion (Part IX) ..... Richard Kaplan
Alternative operating systems for the Model I/III

48 Cottage Computery Comes of Age ..... Michael Herbert Shadick
How far has the computer age *really* come?

50 Build with the Electric Pencil ..... Jerome I. Weintraub
Use the Electric Pencil to create BUILD files

52 How to Get Rich and Famous! ..... C. Brian Honess
Solve some famous mathematical problems with your computer

57 Epson MX-80 Graftrax Initialization Program ..... Joseph Rosenman
Enable yourself to use the powerful features of GRAFTRAX

REGULAR DEPARTMENTS

2 Bits and Pieces ..... Howard Y. Gosman
Publisher's Remarks

4 The Crystal Ball
News and rumors of interest to TRS-80 owners

9 Letters to the Editor
Readers tell us what's on their minds

32 Beginner's Corner ..... Spencer Koenig
Lend an ear to the Big Boys (part 2)

40 Color Computer Corner ..... Joseph Rosenman
This Month: TRS-80 Color Basic, Color Berserk, Sir Eggbert Jumper

60 Pocket Computer Corner ..... S. M. Zimmerman and L. M. Conrad
This Month: Solving Simultaneous Equations

63 Questions and Answers ..... Hubert S. Howe, Jr.
Answers to questions posed by readers

68 Advertising Directory

All correspondence should be addressed to The Editor, H & E Computronics, Inc., 50 North Pascack Road, Spring Valley, NY 10977. Unaccepted manuscripts will be returned if accompanied by sufficient first class postage. H & E Computronics will not be responsible for the return of unsolicited manuscripts, cassettes, floppy diskettes, program listings, etc. not submitted with a self-addressed, stamped envelope. Opinions expressed by the authors are not necessarily those of H & E Computronics, Inc.

For information about receiving copies of *COMPUTRONICS* in quantity contact:

Worldwide Media Service, Inc.
386 Park Avenue South
New York, New York 10016
Attenion: Sandra A. Joseph
Cable: WORLDMEDIA
Telex: 620430 (WUI)
Tel.: (212) 686-1520

U.S. and Canadian Distributor

H & E Computronics, Inc.
50 North Pascack Road
Spring Valley, New York 10977
Attention: Steven M. Kahan
Tel.: (914) 425-1535

BITS AND PIECES

Howard Y. Gosman

ON THE COVER

On the cover this month is the book *Learning TRS-80 BASIC for the Model I, II/16 and III.* You may consider this to be our plug of the month, but everyone who has a TRS-80 should have a copy of this book. *Learning TRS-80 BASIC for the Model I, II/16 and III is a revised (actually massively revised) version of David Lien's Learning Level II* (first introduced in 1977).

David A. Lien, the author, is the **best** computer book writer anywhere ever! His best selling book is owned by over 300,000 Radio Shack computer owners. As a matter of fact, I believe that Dr. Lien is the major factor that made the TRS-80 a successful computer and perhaps, the one person most responsible for the success of the small and personal computer. How did he do it? David Lien wrote the original manual that accompanied the orginal TRS-80 computer. At that time, no one understood anything about computers (at least if we limit our discussion to "regular people"). Dr. Lien wrote an illustrated, step-by-step, easy to follow, non-intimidating manual that accompanied every TRS-80 Model-I computer. If you walked into a Radio Shack store five years ago, no one at Radio Shack knew anything about computers. Many of the stores had a Radio Shack computer sitting on the shelf. Just ask the salesman about the computer and the reply was, "It's a great computer . . . Just take a look at the manual." Well, I took a look at the manual . . . and sure enough, it did look easy.

Why am I making such a big point about the manual? Many years back, in the beginning, RADIO SHACK released their first computer. Their original computer contained a watered down version of the present BASIC in use (called LEVEL-I BASIC).

continued on page 7

The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE is published by H & E Computronics, Inc., 50 North Pascack Road, Spring Valley, New York 10977. *The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE* is not sponsored, nor in any way officially sanctioned by Radio Shack, a division of Tandy Corporation.

The purpose of the *H & E COMPUTRONICS MONTHLY NEWS MAGAZINE* is to provide and exchange information related to the care, use, and application of the TRS-80™ computer systems. H & E COMPUTRONICS, Inc. does not take any financial responsibility for errors in published materials. Users are advised to check and edit vital programs carefully.

The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE encourages comments, questions, and suggestions. H & E COMPUTRONICS will pay contributors for articles and programs published in the magazine.

The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE is typeset by Photonics, Ltd., 188 Highwood Ave., Tenafly, NJ 07670, and is printed by Kay Offset Printing Service, Inc., 154 Grand Street, New York, NY 10013.

SUBSCRIPTION RATES

$24 per year	SURFACE MAIL	U.S. Only
$36 per year	FIRST CLASS MAIL	U.S.
$36 per year	AIR MAIL	Canada and Mexico
$48 per year	AIR MAIL	Outside U.S., Canada and Mexico
$3 per copy	Single Copies	U.S., Canada and Mexico
$4 per copy	Single Copies	Outside U.S., Canada and Mexico

Foriegn subscriptions and sales should be remitted in U. S. funds drawn on a U.S. bank.

YOUR SUBSCRIPTION HAS EXPIRED IF . . . THE NUMBER ABOVE YOUR NAME AFTER THE DASH ON YOUR MAILING LABEL IS 50 (OR LESS). THE NUMBER FOLLOWING THE DASH TELLS YOU THE LAST ISSUE THAT YOU WILL RECEIVE. For example, if your subscription number is 16429-50, your subscription expires with this issue (issue #50).

II + II =

(BUSINESS DYNAMICS)

DOSPLUS II and the TRS-80 model II. The Dynamic Duo.

The TRS-80 model II was created to help you take care of your business. DOSPLUS II was created to help your TRS-80 help you and your business even more.

DOSPLUS II replaces your O.E.M. dos with an ultra-sophisticated operating system that is 5 or more times faster, more reliable and is totally compatible with all standard business oriented software designed for use with the model II. The result is a computer/operating system that far exceeds manufacturer's specs—with more potential and more capacity to be of service to you.

DOSPLUS II operating systems are available for either your floppy or hard disk drives. The hard drive operating system is compatible with Radio Shack, QuCeS, Corvus Micro Mainframe, VR Data and other popular hard drive sub-systems.

DOSPLUS II...All the quality and dependability you'd expect from DOSPLUS.

For the dealer nearest you call
1-800-327-8724
Dealer inquiries invited.

$249.95

For product information (305) 983-3390

DOSPLUS DOSPLUS first in quality! First in the industry!

To order call toll free 1-800-327-8724

POWERSOFT
A Division of Breeze/QSD, Inc.
11500 Stemmons Fwy., Dallas, Texas 75229
(214) 484-2976 (800) 527-7432

MICRO SYSTEMS SOFTWARE, INC.
4301-18 Oak Circle,
Boca Raton, FL 33431

THE CRYSTAL BALL

(News and Rumors of Interest to TRS-80 Owners)

There have been some fantastic new devices and features for microcomputers introduced in the past few months. It's now possible to have your computer read any ASCII text and speak it out loud, you can use your computer to access individual pictures from a Videodisc system, you can buy almost anything (from stereo and TV equipment to linens and china) over network systems (like SERVE on THE SOURCE), you can control lights and appliances when you're not home, and you can even choose from several makes of robot kits for your TRS-80. Here's a few more inventions that are now being mass-produced for microcomputers:

STRANGE NEW KEYBOARD FROM ENGLAND

The evolution of the typewriter from the first full-keyboard manual machines, through electric typewriters to today's word processors has not been matched by an evolution in the practice of touch-typing. The original "QWERTY" keyboard was designed not to speed up typing, but rather to **slow it down.** The first typewriters would simply jam if you typed two letters in succession too quickly. On the QWERTY keyboard, only one of the eight "home keys" is a vowel—the key under your left pinky, and due to the frequency of use of different letters, the left hand does far more work than the right hand. Consequently, we are left today with a keyboard that defies both logic and reform. To date, no alternative keyboard has ever been accepted, because hundreds of thousands of typists already can type on a QWERTY, and see no reason to learn a whole new keyboard now. A novice typist could be taught an alternative keyboard layout, but it would be hard to find a teacher, and harder still to find a job where the keyboard is used (although reprogramming of keys may soon allow the customizing of any keyboard through software). The most famous alternative keyboard is the "DVORAK" keyboard (named for its inventor) which was designed according to ease of learning and operation and the average frequency of use of each letter in the alphabet. This keyboard has attracted much attention and acclaim, but has never had enough followers for it to catch on in a big way. Now a British company has suddenly advertised a keyboard with a totally new concept in text typing. The MICROWRITER first appeared in a full page ad in the Wall Street Journal on July 20, 1982, where the company solicited American distributors to take on their new product. It's hard to believe, but this is a one-handed keyboard for typing text. The MICROWRITER has only five main keys (one under each finger) and a control key, and yet the Microwriter company says that it is more efficient than an ordinary keyboard for typing text. This small rectangular unit (about the size of a large scientific calculator) is a good example of modern "ergonomic" design—it is molded to the shape of your hand and fingers so that it's nearly impossible to hit the wrong key. You just place the palm of your hand on the keyboard, and your fingers naturally fall right into place. This keyboard could cause a revolution for in-the-field researchers and reporters: a reporter working outside in freezing weather could probably put the MICROWRITER into a coat pocket where it's nice and warm, then put his hand in his pocket and start typing! Back at the hotel room, the MICROWRITER gets plugged into a TV set for reading and editing of the text, then the phone is used to send the story to the home office.

Their theory is that the QWERTY keyboard is the reason that word processing remains in the domain of the secretary—because few executives will learn to touch-type on a QWERTY keyboard. They maintain that the "coding" used by the MICROWRITER operator is so simple that anyone can learn to use it in about an hour. They hope that executives will not hesitate to learn its use, the way they frequently balk at the idea of learning operating a computer or word processor

continued on page 6

A Sweet Deal...

Buy an Alpha Joystick and games together and deduct up to $14.00 from your order.

(Model I version shown)

The Alpha Joystick adds arcade-style control to TRS-80 action games. Simply plug it in and begin playing joystick compatible games. No modification, wiring or batteries are required, and the Alpha Joystick is compatible with other TRS-80 accessories. The instructions are clear and complete. We even show how easy it is to experiment in BASIC (A=INP(0) reads stick) and convert BASIC programs to Joystick control.

MODEL I - Plugs directly into any Level II Keyboard (card edge on rear) or expansion interface (left side, next to printer port).

MODEL III - Works with any 'Model III BASIC' system and plugs into the 50 pin I/O bus (largest edge connector underneath).

THE ALPHA JOYSTICK

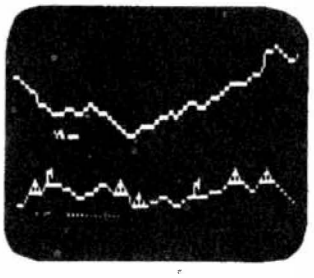
Joystick + 1 game Deduct $ 6.00
Joystick + 2 games Deduct $12.00
Joystick + 3 games Deduct $14.00

ONLY $39.95

MODEL I OR III, SPECIFY WHEN ORDERING. PRICE INCLUDES ATARI JOYSTICK + ALPHA INTERFACE + INSTRUCTIONS + DEMO PROGRAM LISTING.

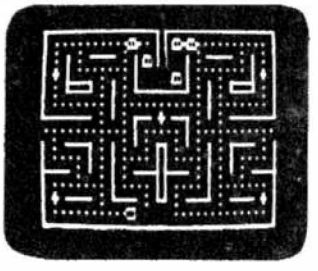
14 DAY MONEY BACK GUARANTEE - if you are not delighted, return it within 14 days for a prompt and courteous refund.

Choose from any of the Joystick Compatible games below. Be sure to mention the "Sweet Deal" discount when ordering.



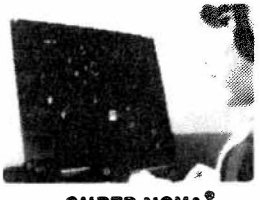
PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missile attacks. With miles of wild terrain and tunnels to penetrate, you're well armed with bombs and multiple forward missile capability. From Melbourne House. Features sound, trainer mode and customizing program.



• • SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You control Scarfman around the maze, gobbling up everything in your path. Try to eat it all before nasty monsters devour you. Excellent high speed machine language action game from the Cornsoft Group. With sound.

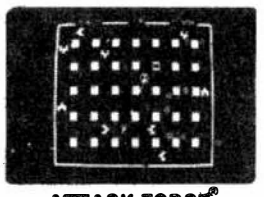


SUPER NOVA®

Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones.) Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine

LUNAR LANDER

As a vast panorama moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored. —if you can land safely. You control LEM main engines and side thrusters. Absolutely the best use of TRS-80 graphics we have ever seen! From Adventure International. With sound.



ATTACK FORCE®

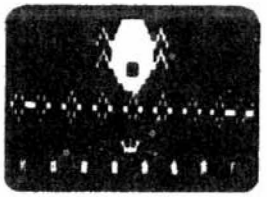
As your ship appears on the bottom of the maze, eight alien ships appear on the top, all traveling directly at you! You move toward them and fire missiles. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the "Flagship". With sound effects!

COSMIC FIGHTER®

Your ship comes out of hyperspace under a convoy of aliens. You destroy every one. But another set appears. These seem more intelligent. You eliminate them too. Your fuel supply is diminishing. You must destroy two more sets before you can dock. The space station is now on your scanner. With sound!

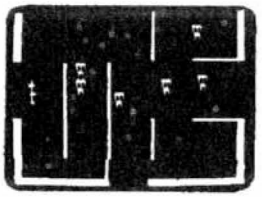
GALAXY INVASION®

The sound of the klaxon is calling you! Invaders have been spotted warping toward Earth. You shift right and left as you fire your lasers. A few break formation and fly straight at you! You place your finger on the fire button knowing that this shot must connect! With sound effects!



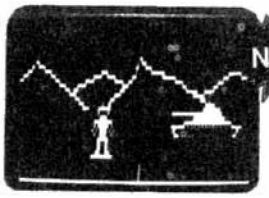
DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from thieving aliens who attack repeatedly. An alien passes your guard, snatches up a cannister and flys straight off. Quick! You have one last chance to blast him out of the sky! With sound and voice.



TALKING ROBOT ATTACK

This game TALKS without a voice synthesizer, through the cassette port. With just a hand laser in a remote section of the space station, you encounter armed robots. Some march towards you, more wait around corners. Careful, the walls are electrified. Zap as many robots as you dare before escaping to a new section. More robots await you.



ARMORED PATROL

A realistic tank battle simulation. Your view is a 3—D perspective of an alien landscape. Manuever your T-36 tank to locate and destroy enemy tanks and robots that lay hidden, ready to assault you. Clever graphics create the illusion of movement and dimension. From Adventure International. With sound.

GAME PRICES

"Penetrator" Disk or 2 Cassette Package	$24.95
"Armored Patrol" Cassette $19.95, . Disk	$24.95
ALL OTHER GAMES:	
16K Level 2, Model 1 + Model 3 Cassette	$15.95
32K Level 2, Model 1 + Model 3 Diskette	$19.95

These games are "Alpha Joystick Compatible." They may be played with or without Joystick (using arrow keys).

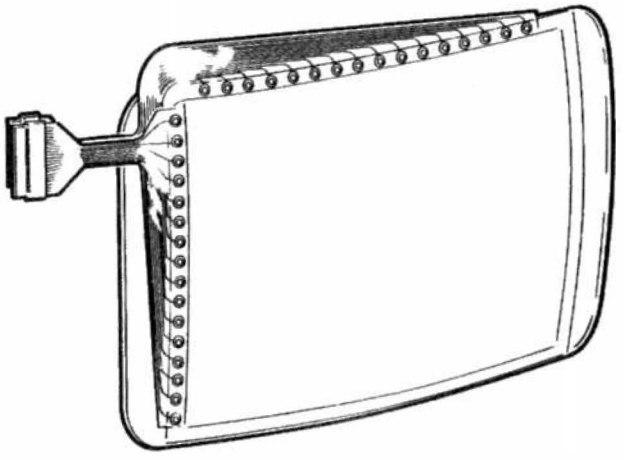
Toll Free Order Line

800-221-0916

ORDERS ONLY. HOURS: 9 AM - 5:30 PM. E.S.T.
FOR INFO CALL: (212) 296-5916

ADD $2.00 PER ORDER FOR SHIPPING AND HANDLING
WE ACCEPT VISA, MASTERCARD, CHECKS, M.O.
C.O.D. ADD $3.00 EXTRA.
NY RESIDENTS ADD SALES TAX.
OVERSEAS, FPO, APO: ADD 10%.
DEALER DISCOUNTS AVAILABLE

ALPHA Products
79-04 Jamaica Ave., Woodhaven, NY 11421 (212) 296-5916



Detector Electronics' Touch Sensitive Panel

THE CRYSTAL BALL

continued from page 4

now. They say that "the learning system is just an extension of normal handwriting, which is why we refer to Microwriting as "electronic handwriting." However, the advertisement carefully avoids direct clues as to the nature of the coding method.

The MICROWRITER is more than just a keyboard, it's actually a complete, battery operated, portable, personal word processor with an 8K memory, a 16 character LCD display, and resident software that provides extensive editing and communication facilities. Through its RS232 port, it can communicate with "practically any" computer or word processor, display 16 lines on a monitor or ordinary TV, interface with a portable cassette recorder, telex, or acoustic coupler, and more.

The advertisement also seems to aim itself at U.S. computer manufacturers, apparently in hopes of making the MICROWRITER a standard option available for popular U.S. microcomputers. The ad predicts that this keyboard will become THE standard keyboard for text typing, and seems to imply that any manufacturer who does not make the MICROWRITER keyboard a standard option soon may get left behind in the rush to convert to the new keyboard.

The actual method of typing with one hand is still an enigma—how can they get all 26 letters, 10 digits, and punctuation symbols without two or more keystrokes per character? And if there are two or more keystrokes per character, how can you ever type as fast as typists do on a standard keyboard? We have sent for more information (and photos) and hope to answer these questions about one-handed touch typing in our next issue.

"TOUCH SCREEN" FOR MICROS NOW AVAILABLE

One of the most exciting developments in interactive computer systems is the development of the touch screen. This touch screen is a curved transparent panel which attaches to the video screen of a computer or videodisc system. This screen has invisible electrical conductors in it, 16 horizontal conductors on a glass panel, opposite 16 vertical conductors on a mylar sheet. When you press any area of the screen, you cause a pair of conductors, horizontal and vertical, to come into contact. This gives the computer the coordinates of the point that was touched.

This device allows you to write a program that displays "buttons" on the screen for various selections or functions, then sense which "button" is pushed by the operator, giving you the ability to create unlimited specialized keyboards, printed right on the screen. In fact, a system designed around this type of device would not need a physical keyboard at all—after an automatic boot-up, all functions of

a specialized system could be carried out by touching "buttons" displayed on screen. Since switch positions are located wherever two of the 32 conductors cross, the touch screen has 256 different positions—that's a big keyboard—and plenty of room for both program prompts and lots of "buttons." An additional application which is of very great potential is in interactive video training and demonstration systems (which we have described at some length in an earlier issue).

The screen is available in 12, 15, 17, and 19 inch models that can be clamped or bonded to video screens. Users must also purchase an interface that allows connection to most microcomputers. Prices range from $840 to $900 for the screens, and the interface costs $200 (or $300 including its own power supply). For more information write or call Detector Electronics, 6901 West 110 St., Minneapolis, MN 55438; (612) 941-5665.

SOME PREDICTIONS . . .

We're going to pass on a few interesting predictions and inside-industry rumors (we heard about them in *INFOWORLD*):

Perhaps because of portables like OSBORNE and imitators, Radio Shack will introduce a portable computer (a real computer, not just a terminal) this fall.

Radio Shack may release a color computer that looks like a Model III, and a Model II without any disk drives—for use as a terminal in a network.

Heathkit is working on a ROBOT KIT that will have amazing features like mobility (it will climb stairs), speech synthesis, voice recognition, and VISION. All this for less than $200.

Xerox has apparently developed a LASER PRINTER with 2400-DOTS-PER-INCH resolution! That is a fantastically fine resolution, far better than can be had with present laser-scanning typesetters (which can produce type as good as the type you're reading now). The most amazing part is that they could probably sell such a printer for $5000-$6000! But rumor has it that they will SUPPRESS the technology rather than make it available to the public! Perhaps because they already have a competitive 380-dots-per-inch printer, they may not want to compete with it. This may not be a good idea, because any innovation that American industry suppresses is bound to come to us soon from the Far East.

Ok, here's our own wild prediction. Within 3 years we will see a computer with many or all of the following features:

Portability (briefcase-sized, battery powered)

Full color flat LED display with med. resolution graphics capability

Touch Screen

More than a Gigabyte (1000 Megabytes) of internal non-volatile RAM

Optional portable 3" hard disk unit with some unbelievable capacity

Built-in four-color printer/plotter

Full size keyboard with standard numeric and editing keypads and programmable keys

Standard software in ROM includes Word Processor, Database, Communications

Optional ROM (plug-in) software (can also store downloaded software in non-volatile RAM)

Speech synthesis and voice recognition

Communicates with *any* other computer or peripheral device

Costs less than $5000 ■

BITS AND PIECES

continued from page 2

At that time, no novice understood or would dare to purchase a computer. Radio Shack had the vision and the luck to hire David Lien to write the manual for their first computer. The results were a huge success. Dr. Lien wrote a step-by-step, well-illustrated learner's manual for the TRS-80. Anyone visiting a Radio Shack store (including myself) was very impressed with the manual that accompanied the computer—this manual was destined to go down in history as the best tutorial manual ever written for beginners. The step by step approach looked so simple that you couldn't go wrong. Of course, the computer with the manual became an instant success and Radio Shack soon released their Level II BASIC. Unfortunately, Radio

continued on page 8

BITS AND PIECES

continued from page 7

Shack and Dr. Lien were unable to agree on financial terms, so Radio Shack had someone else write the manual for the more advanced BASIC (but Radio Shack did continue to include Dr. Lien's manual with every computer, since all the commands and approaches in it also apply to Level II BASIC). Radio Shack never did come out with a good training manual for Level II BASIC—their manual is strictly a reference work. So, Dr. Lien went on to form his own company and fill the gap by selling his own book, *Learning Level II BASIC*, without the help of Radio Shack. He was able to sell thousands of copies at $16.95 each.

That's the story of David Lien. Now, Dr. Lien has published a greatly expanded and updated version of his book called *Learning TRS-80 BASIC for the Model I, II, III and 16.* This is the ultimate tutorial work for anyone wishing to learn TRS-80 BASIC. It also includes plenty of aids, hints and ideas for experienced programmers at any level. Written in a relaxed and amusing style, this book leads the begnning user step-by-step through the many spects of BASIC programming. Sample programs and ideas for writing software abound.

Easy-to-understand directions guide the reader confidently to programming proficiency. Questions and answer sections in most chapters reinforce learning of the material covered.

EVERY TRS-80 OWNER NEEDS AND SHOULD HAVE A COPY OF *Learning TRS-80 BASIC for the Model I, II/16 and III.* New computer owners will be delighted with the TRS-80 BASIC TUTORIAL section. It is the best TRS-80 tutorial anywhere. Intermediate TRS-80 users will find every TRS-80 command described within the manual. Not only will you find an easy to read description of each command, each command described comes with one or more simple sample programs that illustrate its use. And even the "know-it-all" programmer who doesn't need this book will find a wealth if precious information not available elsewhere, including information on PEEKs and POKEs, sorting and sorting techniques, Disk BASIC usage, error handling and more. Educational institutions using computers will find the question and answer approaches with exercises to be extremely helpful to students (and virtually eliminate the need for the teacher all together).

Do I like the book? If you check the past 50 issues of Computronics, you'll find that no other computer product available has received such high praise. I certainly do believe that this is the one product that I would recommend as essential to every TRS-80 owner. I do attribute most of my early computer knowledge Dr. Lien's original manual and find that I use his latest book is even more valuable.

Learning TRS-80 BASIC for the ModelS, I, II/16 and III (written by David A. Lien and published by COMPUSOFT PUBLISHING) is available from H & E COMPUTRONICS, INC. for $19.95 (plus appropriate postage—$3 within the U.S.).

COLOR COMPUTER SOFTWARE

RADIO SHACK has decided to allow non-RADIO SHACK stores to sell the TRS-80 Color Computer (although it won't be called TRS-80). I've always believed that the TRS-80 Color Computer is Radio Shack's most advanced computer, but know one knows it yet. Radio Shack has decided to market the color computer at a very low price. The operating system that Radio Shack released for the disk drives is substandard (compared to the operating systems available for their other computers). Hence, very little good software has become available for the color comptuer. Perhaps, with Radio Shack's new policy, non-Radio Shack vendors will come up with the software that will really allow the color computer to take off.

TANDY WILL SELL COMPUTERS OUTSIDE OF RADIO SHACK

Tandy has made another strange marketing move: they're going to sell a non-Radio Shack version of the COLOR COMPUTER (to be named the TDP SYSTEM 100) to independent computer dealers. Apparently, Tandy feels that it is missing a major part of the market. With thousands of computer stores now popping up everywhere, and dozens of computer manufacturers fighting over shelf space, Tandy wants some of that shelf space too. This may not turn out to be a good idea for Tandy, and will definitely be bad news for Radio Shack.

The problem is this: the new Tandy TDP 100 is identical with Radio Shack's COLOR COMPUTER. The only difference is the color of the case and the brand name printed on it. Although this computer is aimed basically at the home market (competing with Atari, VIC-20, etc.), it has always been about the highest-priced computer in its class (even considering a recent $100 price cut on the Radio Shack version). The Tandy version will sell for exactly the same price as the Radio Shack machine. Since the TDP 100 and the Radio Shack COLOR COMPUTER

are to have identical characteristics, and the same high price, then Tandy's **only** real competition will be Radio Shack! Employees of the Radio Shack division, especially Radio Shack store owners, will undoubtedly have some complaints for their parent corporation. If more low-cost, powerful color computers become available soon (like the Sinclair SPECTRUM, with its $95 disk drives and 64K), then both versions of Tandy's computer will be in big trouble.

If Tandy were a bit wiser, they'd concentrate on getting some good software available for the COLOR COMPUTER—it's basically a very good machine, but the lack of available software may drive most of its owners crazy. And Radio Shack has enough trouble keeping up in the market without competition from within. ■

LETTERS TO THE EDITOR

Sorting Things Out

I'm a recent subscriber to *Computronics* and am very favorably impressed by your editorial content. Perhaps you or your readers could help me.

In developing programs on the Model III, I have found the CMD"O" string sort to be very useful, and fast compared to BASIC. But I have wished for a similar sort for numeric arrays.

OK, so I can read a double precision list off disk as a string array, then sort then save back to disk and later read in as a numeric array. I've used this technique, but a double precision number requires a string length of 17 characters, as opposed to 8 bytes handled as a number. Thus, the size of the array is limited to less than 2000 numbers. If sorted as double precision numbers, the array could be twice as large. Similar arguments can be made for single precision and integer numbers.

What I need is a machine-code sort that can be called from BASIC as easily (almost) as CMD"O". The routine on page 16 (August 1982) is the idea, but for numbers. (I wonder why the author didn't avail himself of CMD"O" for the Model III version.)

Perhaps you have published such a sort in past issues. Perhaps one of your readers would share one with us. I'm not an assembly programmer so it will have to be handed to me on a silver platter as clearly as in Mr. Wicks's article referred to above. If I use it in a commercial program, I'm willing to pay (a little!).

Royal Dossett
Datawriter Corporation
2793 Pheasant Road
Excelsior, Montana 55351

First, a machine-language sort for double-precision numbers wouldn't be nearly as fast as you may expect, because the Z80 instruction set does not include double-precision calculations. Calls would have to be made to the ROM subroutines, and there would be little savings over simply doing the sorting in BASIC.

We are currently publishing a series of articles on sorting called "Array of Hope for BASIC Programmers" by Arne Rohde. These articles may give you some ideas on how you can handle sorting more efficiently in BASIC.

If you want to try a commercial utility package, we would suggest that you look into GSF (Generalized Subroutine Facility) from Racet Computes or FASTSORT from Howe Software, but check the documentation to see if these packages do what you require before ordering.

Adjusted Patches

I saw in the "Model 3 Corner" (July '82) that you suggest a patch to TRSDOS that gives full error messages instead of cryptic error numbers. Bravo!

However, please note that the patch as give only applies to version 1.3. Under the 1.1 and 1.2 versions the same code comes two bytes earlier.

That is, substitute ADD=4E26 in the parameter list, for disks of older vintage. Typically these will be disks with Radio Shack software packages using lots of disk file manipulation, which do not work under the newer 1.3 version. Reason: in order to fix one bug (killing open files would destroy the disk directory) they accidentally aranged it so that *all* files were closed by BASIC's CLOSE command, regardless of file numbers given

continued on page 10

LETTERS TO THE EDITOR

continued from page 9

in the command. Any program that needs to close one file while leaving others will not work under TRSDOS 1.3.

Peter Dyson
510 Rutgers
Swarthmore, Pennsylvania 19081

Misinformed Merge

We received a mountain of replies to a statement in the August issue. Mr. William Byberg of Salem, New Hampshire, wrote to ask whether we had ever heard of a method of merging programs loaded from cassette. We said "No". We were wrong. We rarely, if ever, use cassette systems here in the office, but we should have remembered that we SELL a cassette merge program! Here's a few of the letters on this and other subjects:

I assume that you received several dozen letters about the statement you made to Mr. Byberg, since I, for one, purchased my merge program from Computronics. It is, of course, Re-Model/Pro-Load from Racet Computes.

In addition to that, for Model I users in your October 1979 issue, you published a renumbering program the first three steps of which made a good merge program. For those turning back to that program: line 20310 needs a colon in it.

Since very little is said about us poor but happy tape users anymore, I would like to pass on a couple of short cuts that made me quit thinking of disks for a while yet.

I found that it was not necessary to bulk erase tapes to reuse them, all that is necessary is to have an erased portion to start the CSAVE. The Radio Shack recorder does about as good a job of erasing as the bulk erasers. By simply recording a small section at the beginning of the tape at no volume, the job is done. The recorder will take care of the rest of it if you start your CSAVE in this blank space.

Rather than keep pulling plugs and turning down the volume, I installed a toggle switch in the tape recorder which overrides the remote control. I also cut the wire leading to the built-in Microphone. The switch saves a lot of time and makes it easier to put a space between programs on the same tape.

One night when I had more than an hour to kill waiting for a data tape to load, I remembered what Dr. Howe said when he answered complaints about long data tapes for his data base program. In the February 1980 issue I re-read the article and started to work, only going much further with the same method.

The results were dramatic, to say the least. The very same hour-long data tape was taped off in 3 minutes.

For you non-believers, here are 2 lines for the data base program. As written, it limits your number of fields to 10. But it will tape a mailing list of 100, using all 10 fields, in 5 minutes.

```
510 INPUT#-1,NI,NF,D(Ø),D(1),D(2),D(3),
D(4),D(5),D(6),D(7),D(8),D(9),AT(Ø),
AT(1),AT(2),AT(3),AT(4),AT(5),AT(6),
AT(7),AT(8),AT(9)
520 FOR J = Ø TO NI:INPUT#-1,A(Ø,J),
A(1,J),A(2,J),A(3,J),A(4,J),A(5,J),
A(6,J),A(7,J),A(8,J),A(9,J),X(Ø,J),
X(1,J),X(2,J),X(3,J),X(4,J),X(5,J),
X(6,J),X(7,J),X(8,J),X(9,J):NEXT J:
GOTO 41Ø
```

If you would like to change your tape routine without re-keying all the data, change only the PRINT#-1 routine. Load the data using the old input routine, then save it to a new tape, which will be done by your new save routine. Then you can change your input routine and you have saved all that typing.

I have tried not to go into detail to make this short, but I think that most readers can tell by example what I have done.

George K. Snell
Rt. 1, Box 75
Honor, Michigan 49640

I just received the August issue of *Computronics*, and read William Byberg's letter requesting a cassette merge for your tape drive Model III.

I've included the BASIC listing of a

merge that I have used successfully for a year now, and found it to be very handy. The program was published in the Radio Shack Newsletter dated September, 1981, on page 8, and was written by David Lemley of Savannah, Georgia.

Although the Memory is protected at 32594 for a Model I 16K machine, it appears to be fully relocatable, and you shouldn't have any trouble running it on your Model III.

```
1  ' BASIC MERGE PROGRAM
2  ' ENTER 32594 FOR MEM SIZE ON POWER-UP
10 FOR X = 32595 TO 32712
20 READ Y:POKE X, Y
30 NEXT
40 DATA 221,42,249,64,221,43,221,43,221,
43,221,43,175,205,18,2,205,150,2,6,4,205,
53,2,16,251,205,53,2,221,119,2,183,32
50 DATA 8,58,63,60,238,10,50,63,60,175,
221,182,0,221,182,1,221,182,2,221,35,32,
22,5,205,248,1,42,249,64,229,221,225,17,
235,66
60 DATA 237,82,229,209,221,43,221,43,
221,110,0,221,102,1,174,124,181,40,13,
175,25,229,221,117,0,221,116,1,221,225,
24,232,221
70 DATA 229,225,35,35,34,249,64,34,251,
64,34,253,64,195,25,26
80 POKE 16526,83:POKE 16527,127
90 NEW
```

After you've keyed in the merge program in and saved it to tape, power up again, and set your MEM size. Then load the merge program like any other BASIC program.

After a successful load, type "RUN" and the screen will clear for a few seconds while the merge pokes itself into memory. Once this is done, the computer will return to the "READY" prompt.

Now you're all set to put it to work. Put your first BASIC program in the cassette, and ready it to load. But don't type "CLOAD". Type A=USR(0). This will activate the cassette recorder, and you will see the asterisks flash in the upper right corner of your screen just as if you were "CLOADing" a tape. When the program is loaded, you'll get the usual "READY", and you're now set to begin loading the second BASIC program. Again, don't type "CLOAD", instead activate the tape recorder

continued on page 12

LETTERS TO THE EDITOR

continued from page 11

with A=USR(0) and the second program will load very nicely.

Obviously, you want your first BASIC program to have line numbers lower than those of the second program, and if you don't have Radio Shack's "RENUM" program, I suggest you pic+ one up (catalog # 26-2004, about $10 on tape).

Without RENUM, you'll end up retyping the line numbers on so many programs that the advantage of the Merge program will be lost.

That's it. Hope I've been able to help, and enjoy your "Urge to Merge".

Buddy King
201 Weber Street
South Bend, Indiana 46637

In the August 1982 issue of *Computronics*, you indicate that you are not aware of any Merge programs for cassette systems. I am surprised at that. I am not sure where I got it from, but I have been using a merge procedure for a long time. Essentially what it does is to set the start pointer to the end of the program in memory. Then you load the program to be merged, and finally reset the start pointer back to where it was—the beginning of usable memory. The line numbers of the program to be merged must be higher than those of the program already in memory. The procedure is as follows, with instructions typed in at the BASIC command mode:

1. CLOAD the first program.
2. PRINT PEEK(16633), PEEK(16634)
3. Call the first number returned X, the second, Y.
4. If X is greater than or equal to 2, then POKE 16548,(X-2):POKE 16549,Y
5. If X = 1 or zero, then POKE 16548,(X+254):POKE 16549,(Y-1)
6. CLOAD the program to be merged
7. On Model I POKE 16548,233: POKE 16549,66
   On Model III POKE 16548,233: POKE 16549,67
8. LIST, SAVE, or RUN the program

Raymond A. Somers
Professor of Chemistry
University of Wisconsin
Stevens Point, WI 54481

Many readers pointed out the above method for cassette merging, which was printed in the July 1979 issue of the Radio Shack Microcomputer Newsletter.

Model II Screen Printing

Regarding "Program Conversion (Part VII)", on SYSTEM "SCREEN" (issue 48, August 1982). For the Model II, if the CRT display is linear, a good substitute for the statement SYSTEM "SCREEN", is to use SYSTEM "DUAL ON" and SYSTEM "DUAL OFF", which may be initiated from a BASIC program by PRINT CHR$(14) (for "ON"), and PRINT CHR$(15) (for "OFF"). No message will be put on the screen.

```
100 PRINT @ L(15,5),"H & E COMPUTRONICS"
  ' SCREEN DISPLAY
200 LINE INPUT "<LP>RINT /CONTINUE/";A$
  ' PRINT OPTION
300 PRINT @ L(16,0),CHR$(23)
  ' CLEAR LINE NOT TO BE PRINTED
400 GOSUB 7000 ' TEST INPUT
500 IF LP$="ON" THEN GOTO 100
  ' TO PRINT RERUN DISPLAY
600   ' REST OF PROGRAM
7000 LP$="OFF"
  ' TO INITIATE "DUAL OFF" AFTER INPUT
7010 IF A$="LP" THEN LP$="ON"
  ' TO INITIATE "DUAL ON" FOR PRINT
7020 IF LP$="OFF" THEN 7040
7030 PRINT CHR$(14);:RETURN
  ' SWITCHES TO "DUAL ON"
7040 PRINT CHR$(15);:RETURN
  ' SWITCHES TO "DUAL OFF"
```

A. O. Wright
51 South Street
Rhinebeck, New York 12572

Richard Kaplan's reply:

Mr. Wright's observation is, indeed, correct; however I cannot see how the DUAL function can be a "substitute" for the SYSTEM "SCREEN" command.

The DUAL command can ony output information to the printer as it is displayed on the screen. SYSTEM "SCREEN", on the other hand can output an entire screen at once. The only disadvantage of SYSTEM "SCREEN" is that it can only be executed from within a program or as an immediate command, i.e., one cannot execute this function while a program is running unless it is actually written into the program. Mr. Wright's technique does not resolve this difficulty. Although his program does ask if line printer output is desired, the same effect could be achieved by writing a SYSTEM "SCREEN" command into the program. Actually, these two methods could serve separate purposes: use SYSTEM "SCREEN" to get a printout of just a selected screen display from your program; and use SYSTEM "DUAL ON/OFF" to get a permanent line-by-line record of ALL screen outputs in the program.

H & E Computronics welcomes letters on any subject. If you wish,a personal reply, please enclose a self-addressed, stamped envelope.

H & E Computronics also welcomes readers to submit programs, articles, or reviews for publication. Please address correspondence to:

The Editor
H & E Computronics
50 North Pascack Road
Spring Valley, New York 10977

Please submit programs (and articles, if prepared on a word processor) on media (cassettes or diskettes). Also please indicate the system it was prepared on, and include any necessary instructions. ■

BeaLin Corp. *OFFERS YOU...*

HARDWARE

TRAX-SW

RETAIN YOUR *TRS-80 BLOCK GRAPHICS CAPABILITY WHILE ADDING ALL OF THE FEATURES OF **GRAFTRAX-PLUS SUCH AS UNDERLINE MODE, SUBSCRIPT/SUPERSCRIPT, LINE DRAWING GRAPHICS, ETC. PLUG IN BOARD SUPPLIED WITH OR WITHOUT **GRAFTRAX-PLUS.

(Not available for serial MX-80 Printers.)

TRAX-SW ..... ($49.95)
TRAX-SW WITH GRAFTRAX-PLUS ..... ($109.95)
GRAFTRAX-PLUS ..... ($69.95)
GRAFTRAX 80 ..... ($65.00)

*Trademark of Tandy Corp.
**Trademark of Epson

PRINTER CABLE FOR EPSON PRINTERS
10 FEET ..... ($24.95)

TEAC DISK DRIVES 40TK/SS/DD .. ($250.00)
(one year warranty)

PRINTER SWITCH

Mod I, III $79.95 Mod II, XVI $89.95

Switch between two printers, use your computer to produce draft copies on one printer. Then make final copies on letter quality printer without switching cables.

REAL WORLD INTERFACE

The Card Electronics REAL WORLD INTERFACE system is for the control and monitoring of machines and real world systems (robotics, heating, air conditioning, lighting, motors, solenoids, etc.). With the use of A.C. and D.C. Solid State Relays and status indicating devices, this system will permit TRS-80 users to connect their computer to the real world.

Serial RS-232 version of REAL WORLD INTERFACE ideal for remote control and monitoring through modems.
24 Outputs, 32 Inputs.

(Call or write for information and prices.)

TRS-80 ROBOTICS CONTROL

PAGE-IT $29.95

A disk utility that prints basic programs in easily readable format.

- COMPATIBLE WITH MOST OPERATING SYSTEMS
- HIGHLIGHTS REM STATEMENTS
- INDENTS CODE — LINE NUMBERS STAND OUT
- UNPACKS BASIC CODE
- MENU DRIVEN — EASY TO USE
- DOUBLE SPACE OPTION
- SUPPORTS SPECIAL OPTIONS OF 15 PRINTERS

INVOICE PROGRAM	$39.95
PURCHASE ORDER	$39.95
STATEMENT PROGRAM	$39.95

(BUY ALL 3 FOR $99.95)

uses Nebs's universal form 9050-1 thru 5
*Requires ***NEWDOS 80 2.0*
****Trademark of Apparat Inc.*

CHECK80 ..... $49.95
Use your computer to produce invoices, purchase orders, statements, and print checks with check record maintained on disk.

INVOICE-PLUS $69.95
Print invoices and store all transactions on disk.

USE OUR DATA LINE TO BROWSE THROUGH THE MANY ITEMS WE HAVE AVAILABLE. YOU MAY PLACE ORDERS OR HAVE SPECIFIC QUESTIONS ANSWERED. DATA LINE FORMAT IS 300 BAUD, 7 BIT, NO PARITY. SATISFY ALL OF YOUR COMPUTING NEEDS, COMPLETE SYSTEMS, DISK DRIVES, PRINTERS, EPSON RIBBONS & REPACKS, PAPER PRODUCTS, PLUS MUCH MORE. CALL OR WRITE FOR FREE CATALOG.

Software shipping and handling — $3.50

PHONE VOICE (301) 490-2477 MODEM (301) 730-2229

TRS-80 SOFTWARE

BeaLin Corp. 9335 Old Scaggsville Rd. Laurel, Md. 20707

VISA OR MASTERCARD ACCEPTED — DEALERS INQUIRIES WELCOME

PROGRAM PREVIEWS

A. A. Wicks

This Month: E/Z SCREEN

One of the most irksome tasks a programmer faces, and frequently a difficult one, is the preparation of programs that utilize screen interaction by the user.

Typically, a program of this type would be one that displays blocks of type as "Name — Address — Social Security Number — etc.," and awaits operator input to fill in this type of data. Because of the sometimes varied nature of screen format, such programs often require that the "query line" be placed anywhere on the screen — not just line-by-line justified-left. This usually means many "print-at" statements, for one thing. And if the hard copy printout is to follow the screen display, this must be considered. All of this entails the tricky programming work referred to, and much of this can be deadly repetitive, not conducive to "go-subs," and error-prone, requiring a great amount of time for troubleshooting.

SOURCELiB Systems has now addressed these problems for the programmer in their programmer's program — E/Z SCREEN, and have provided a commendable worksaver. My first impression from their advertising was that this was a user program for graphics displays and operation. This was erroneous, as a detailed study of their wordy advertisement disclosed. E/Z SCREEN is definitely a screen design and code generation tool. Much of the coding required to get data into and out of a program has been eliminated for the programmer. The user also benefits from E/Z SCREEN techniques, because input errors made prior to current input may be corrected without restarting sometimes complex and difficult screen entry data.

Furthermore, the programmer may prepare screen displays that are professional in appearance, at the same time as they may be complex in their utility. The screen may be formatted in any way, such as in the manner that a word processor allows changes and entries. E/Z SCREEN considers all data on the video display as a single logical unit called a "screen." The screen is composed of literals and fields. A literal is a fixed text string that appears on the screen exactly as it is keyed in, and is used to convey non-variable information to the user. Literals are ommonly used to identify the fields on a screen as well as the screen itself. Literals may indicate user options on a menu-type screen, or list information on a "help" screen.

During output the contents of a named variable are moved to the screen according to the mask specified for the field in the definition program. During input the cursor is positioned to indicate where characters will be placed. The cursor may be moved from field to field using various control keys. The cursor is non-destructive, that is, it may be placed on characters, tabbed, etc. without altering data under it. Overtyping is therefore possible. All keys repeat if held down continuously.

Once the format has been established to your satisfaction, E/Z SCREEN will immediately generate all the BASIC subroutines required to process these screens into your program. It should be noted here that when I speak of the "programmer" and the program being produced, I do not mean that the user of E/Z SCREEN need be a professional programmer. Far from it, E/Z SCREEN may be used by practically anyone with a knowledge of writing simple programs in BASIC — although the person creating complex programs is going to gain the greatest utility from this unusual program.

You may create as many as 12 different screen displays in a single program, with up to 100 fields per screen, at the same time using up to nine different functions if desired. One GOSUB establishes the communication between the created program and the E/Z SCREEN code.

One feature of this program that I like is that no USR calls are required for screen processing. Granted, this may slow down operation, but the complexity of coding these functions is eliminated. Some of the other attractive features are the writing of literals to the screen, the writing of data from variables to the screen, and the reading of data from the screen with the keyboard under program control. Also, all fields from the screen may be read into your variables without operator intervention. Input and output fields in memory and on screen may be cleared and screen messages may be displayed, with a wait-for-reply format.

Some of the user operations that are available are a non-destructive cursor, character insert and delete, overtype, and complete cursor control using all arrow keys. Six tab functions are possible. Numeric validity checks may be made, as well as numeric formatting with extended masks.

Programmers will find useful functions such as the centering of literals and fields on the line, changing of field lengths, replication of literals, fields or lines, and the automatic arraying of fields. The cursor character may be specified by a simple command. A Cross Reference utility permits the printing of the screen layout and its attributes, and also prints a field attribute summary.

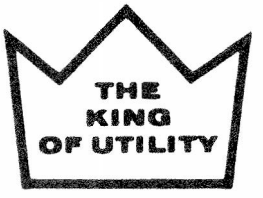
Making use of E/Z SCREEN is not extremely difficult, but as with any program offering so many features, it is necessary to thoroughly understand what you are setting out to accomplish, and how E/Z SCREEN will do it for you. This is best approached by reading the manual from front to back, at least once; then studying it while working with the sample exercise provided. As the manual succinctly states, "...Section 3 (the tutorial) is not something to read — it is something to do...You must be sitting in front of your machine, doing the examples, while following the text in the manual." Once familiar with E/Z SCREEN, you will feel confident in working with it with your own BASIC program.

Because the manual is such an essential part of working with E/Z SCREEN, I am going to digress from my usual review sequence and evaluate it now. This manual comprises 118 pages of loose-leaf 8 1/2 by 11 inch material in a three-ring hard cover binder. Printing is by offset from typewriter or impact printer, and is clear and legible in 10-point size. Spelling and typographical errors are minimal. An addendum sheet explains how to make a backup of the original disk, utilizing either TRSDOS 2.3 or NEWDOS 2.1.

"I believe Super Utility or Super Utility Plus should be present at every serious TRS-80 disk installation."

(We didn't say this; Paul Wiener did in 80 Microcomputing, Jan. '81. . .*but we sure agree with him!*)

Get Yourself a "Super" Utility from the most powerful software vendor.



NOW AT SPECIAL PRICES:

Super Utility*

- Mod I, single density only
- 24K machine language
- Zap
- Purge
- Format
- Special format
- Format without erase
- Disk copy
- Special disk copy
- Tape copy
- Disk repair
- Memory utility
- Protected disk
- MORE

- $29.95
  (Reduced from $49.95)

*While they last!

Quick Fix*

- Mod I, single density only
- Zap
- Purge
- Format
- Format without erase
- Copy
- Disk repair
- Recover killed files
- Memory utility
- TRS DOS, New DOS + (2.1) only
- MORE

- $19.95
  (Reduced from $34.95)

*While they last!

SuperUtility Plus

You heard about it! You read about it (80 Microcomputing). Now get the "cadillac" at a special price! Compatible with Mod I, and Mod III, and all the current operating systems! Copy files from any DOS to any OS, Mod I or III, without converting!

- Zap
- Purge
- Format
- Special Format
- Format without erase
- Disk copy
- Special disk copy
- Tape copy
- Disk repair
- Memory
- File Utility Configurable System
- MUCH MORE
- Mod I & Mod III on Same Disk

- $74.95
  (Now includes backup)

Dealers! Place your orders now at this special price!

POWERSOFT

11500 STEMMONS EXPRESSWAY, SUITE 125
DALLAS, TEXAS 75229
PHONE (214) 484-2976
PHONE (800) 527-7432

DEALER INQUIRIES INVITED

MasterCard VISA

Shipping is included in sales to Canada & U.S.
FOREIGN ORDERS ADD $10

POWERSOFT IS A DIVISION OF BREEZE/QSD, INC.

Model III users will use the CONVERT utility.

The largest part of the manual is a tutorial, already mentioned, of 35 pages and is very explicit. As stated, it should be read and worked with closely, and will probably appear tedious due to the length and detail of the instructions. I discussed this approach with the author, who stated that he felt that he wanted to provide every possible piece of information to the user, together with detailed guidance, so that there would be no questions in the mind of the person working with E/Z SCREEN. I certainly do not fault this approach — too many times have I struggled with a program while trying to determine exactly what the author of the manual really meant. It is far better to have too much instruction than too little. As most of the instruction is devoted to the preparation of a "Personal Telephone and Address Directory," it does have an ultimate usefulness too, and should permit the user to develop other data programs quickly.

The next largest portion of the manual relates to the "Screen Format Program" of E/Z SCREEN, which allows the programmer to actually change and create screen definitions that are used to generate screen-related BASIC code for programs. The remaining Chapters discuss the Cross Reference Utility, interfacing with BASIC programs, error messages and their cause. There are many error and report messages — errors and changes should be no problem. Information is also provided regarding the various screen formats of E/Z SCREEN.

Writing style is informal, but free of quips and sundry asides, which certainly permits it to be rated higher in my estimation than many publications. Overall, the manual rates an 8, on a scale of 1 to 10.

Once you feel familiar with the operating features of E/Z SCREEN, you will be anxious to design your own screens for some particular program. I know that I was, and I was not disappointed. The example Phone Directory is an easy one. How about a multi-field screen layout for a complete personnel medical history sheet? Although this will take more time than the former, it is just as easy. If you decide that you wish to change your fields, or have them in some specific sorted order after you have already completed a layout, the Commands available permit this. More about the Commands in a moment.

You will start your program development by designing your screens using the Screen Format program. You will probably have some idea of the proposed layout, or perhaps have it firmly done beforehand on a paper grid, just so that you will not forget anything. One advantage here is that E/Z SCREEN reports, as you move the cursor around the screen, the "print at" position, by showing a question mark and symbol followed by a number at the left side of line 16. The number, from 0 through 959, represents the screen position under the cursor at the moment. When you have completed the layout to your satisfaction, you "save" the layout using a file name plus /EZS.

When all screens required have been completed (it could be just one, of course), the Code Generation program is used for that purpose. This code will do the screen-oriented input/output for the program. When the generated code is complete, it too may be saved on disk, with the file name plus /EZC extension. You may then add your own code, either through the keyboard entry method or by a MERGE. The /EZS and /EZM files are used by the Cross Reference program to produce screen and program documentation (the /EZM having been a saved file of screen names produced by the Code Generation program).

The E/Z SCREEN generated code will occupy Lines 5 through 999. Your program should be written to begin at Line 1000. All communication between your program and E/Z SCREEN is accomplished by a simple GOSUB to Line 900, which takes it from there. The functions to be performed are contained in a string OB$. Control returns to the program when all functions in OB$ have been completed. The Screen Format Commands, which I said we would return to, are single-letter mnemonic ("cursor" is "K" as the only exception). There are many, which precludes discussing all of them here, but a few examples will suffice. For instance, <L>ENGTH is used to change the length of a field (remember, this could be a field you formatted many lines ago, or at a previous programming session). The cursor is placed on the desired field and the above letter plus ENTER is completed. The Field Name appears, followed by LEN=nn NEW LEN=nn SHIFT=Y. You merely enter the desired numerical character length (maximum of 64 alpha or 24 numeric), and accept or reject the shift default Y. SHIFT=Y will cause all data to the right of the field on that line to be shifted left or right by the same number of positions added or deleted. Making SHIFT=N will shorten or lengthen the field without repositioning data.

Another Command, <N>AME permits changing the name, type, autoskip attribute, or description of a field. The <K> command will define the cursor character for the screen. This Command too, is easy to perform, as are all of the Commands, for that matter. Upon entry, Line 16 will display: CURSOR (ASCII) ==> cursor value.

With the Model I, the ASCII cursor value (129-191) is entered; 129-255 is used for the Model III. The cursor will immediately change, and remain. The cursor selected will also be used on the generated screen.

E/Z SCREEN is a program with a potential that may be easily underestimated until you begin to use it. Programmers and TRS-80 users alike will find its unusual usefulness of real value, and its cost is low enough to make it a very worthwhile investment. I can recommend it without hesitation.

E/Z SCREEN — SOURCELiB Systems, 1670 Pershing Street, Valley Stream, NY 11580. For TRS-80 Models I and III, 48 K, with minimum of two disk drives — $149.95.

* * * *

Readers will be interested in a new catalog from Prosoft, producers of the eminent Newscript word processing program, and utility programs. This well prepared and printed catalog describes in great detail the upgraded Newscript — version 7.0, which has a number of enhancements not previously available. (I have updated my Newscript to this latest version, and consider the improvements as outstanding.) Newscript emphasizes printer compatibility in addition to being an excellent word processor, and now supports every printer available from many manufacturers. Present owners of Newscript 6.0, 6.1, and 6.2 may upgrade for a modest fee. The other programs from Prosoft are also described completely with all of the details one could ask. ■

ARRAY OF HOPE FOR BASIC PROGRAMMERS (PART 2)

Arne Rohde

5. Searching Techniques

5.1 Sequential Searching

Sequential searching of an array consists of starting a search at one end of the array and continuing with each element until the desired element is found. If all elements are equally likely to be required, the search routine will require a compare with half the elements in the array on average. This makes the search routine slow, especially for arrays with more than a small number of elements, perhaps 20 or so. For the following search algorithms we will assume that we are searching the array A, which contains AE elements. The required key is located in AR. A and AR can obviously be defined with any variable type, and AE should be defined as integer for fastest processing, as should all the subscripts which will be used. This can be done with the DEFINT statement, or by appending a &-sign onto the variable names.

5.1.1 Unlinked Arrays

The sequential search can be described with the following algorithm:

```
100 I = 0
110 IF I < AE THEN IF A(I) <> AR THEN I = I+1: GOTO 110
120 IF I = AE PRINT "Key not found"
    ELSE PRINT "Key found in position:"; I
```

There are several things to note about this code. The first is that the search starts with the first element in the array, and stops as soon as the required element is found, or the index I has reached the number of elements in the array. The first compare in line 110 checks for I less than AE since AE is the actual number of elements, and this value may be zero. The sequence for the two IF statements in line 110 is necessary to avoid referring to an element outside the limits of the array, with a resulting subscript out of range error. If the array A was defined with DIM A(20) then the maximum value of AE would be 21, since 21 elements can be contained in the array. Any reference to A(21) would, however, result in an error. The variable AE can also be thought of as a pointer to the next position in the array where a new element can be inserted.

A FOR .. NEXT statement could have been used to control the search loop, but this would either result in a GOTO out of the FOR .. NEXT loop, or destroying the index to terminate the loop prematurely if the element is found. If neither of these options is desired, then every element of the array must be compared. For example:

```
100 FOR I = 0 TO AE-1
110 IF A(I) = AR THEN PRINT "Key found at location:"; I
120 NEXT I
```

which will not print the message for key not found, and which will not execute correctly if there are zero elements present in the array (the array is empty). To cater for this special case, a test would have to be made, and the code shown above made conditional. Sometimes a sequential search can be simplified if the array is set up so that the desired element will always be found. This can be done by reserving an extra element in the array, for example element zero, where the search key can be stored. In this case AE would be the last subscript value which could be used. For this search it would be natural to start from the end of the array and search with a descending subscript.

```
100 A(0) = AR: I = AE
110 IF A(I) <> AR THEN I = I-1: GOTO 110
120 IF I = 0 THEN PRINT "Key not found"
    ELSE PRINT "Key found at location:"; I
```

This code has the advantage that the loop is simpler and thus faster to execute with only a single IF statement instead of two. This code will also be correct if the array is empty, the only element which will be examined in this case will be the extra one at location zero. The array definition for this example defines A with a DIM value equal to the actual number of elements which it can contain, for example DIM A(20) for an array with up to 20 elements.

There are other methods which can be used to speed up sequential array searches, especially if something is known about the distribution of key values in the array, or the distribution of the elements being searched for. If, for example, 20 per cent of the elements are required in 80 per cent of the searches, then these elements should be placed where they will be found first. This would be at the start of the array for the first method shown, and at the end of the array for the last method. Arranging the array data in this fashion can speed up the searches dramatically. The different key values may not be evenly distributed, but if the most likely keys are unknown or if they vary from time to time then the array cannot be organised to cater for this. If the distribution is very uneven then it may be faster to rearrange the array dynamically so that the keys referenced most often are moved to one end of the array. This could be done by exchanging the element just found with the one next to it, for example using the following code

```
100 A(0) = AR: I = AE
110 IF A(I) <> AR THEN I = I-1: GOTO 110
120 IF I < AE AND I > 0 THEN A(0) = A(I): A(I) = A(I+1):
        I = I+1: A(I) = A(0)
```

This would tend to move the keys referenced most often towards the upper end of the array, where they will be found with the fewest number of compares. For small arrays and for even distribution of key values searched for the code will be more inefficient, but for very uneven distributions it can save time for searches. Note that the index is updated when the

elements are exchanged so that it still points to the required key. The method can be refined by keeping a count of the number of times each key is referenced, and only exchanging the two contiguous elements if the reference count of the key found is larger than the reference count of the following key. This reduces considerably the number of exchanges, and can reduce the amount of oscillation which can occur if two contiguous keys are referenced alternately.

In the above examples, the array has been shown with only one dimension. The algorithms are, of course, also applicable to arrays with two or more dimensions, but for the algorithm which exchanges array elements more exchanges would be required if the array had more than one dimension. This would make the method less attractive, as would the case where the array consists of string keys since it would increase the number of string reorganizations required. To test the savings possible by moving the most used keys to one end of the array, a test was run with a two dimensional array, consisting of a sequential key value from 1 to 25 and a reference count to control the exchanges. The search key was unevenly distributed with almost half the key values unreferenced. It was calculated with the function RND(5)*RND(5) so all values would be found in the array.

The times recorded for the searches were very interesting. The times for the first 200 searches were almost identical using element exchanges and using a direct sequential search without exchanges. This was about 57 seconds, including time to fill the array and display the values searched for. The next 200 searches took almost the same time without exchanges (within a second or two), but using exchanges the time was reduced to about 39 seconds. Using exchanges the time is eventually reduced to 31 seconds for searching with 200 key values. The total time for 800 searches was 158 seconds using exchanges, and 223 seconds without exchanges. This represents a reduction of almost 30 per cent by using extra processing time to rearrange the array elements. It also indicates that the number of searches should be about 10 times the number of elements in the table before the break-even point is reached for a distribution similar to the one used here. For other distributions the break even point would be different, but the method could be worth trying for some applications. It has the added advantage that it is easy to implement in assembly language as well as in Basic.

Sometimes the array elements cannot be rearranged even if it could be advantageous to do so, but even in this situation the distribution of search keys can be exploited. Quite often the same key will be referenced several times in succession. If a customer number is entered for each item sold, and the number is to be found in an array, then the same customer number will be entered several times if the customer buys several items at the same time. Before searching through the array a quick check could be made to see if the search key was the same as the one used in the previous search. This would obviously mean having to store the subscript after each search, possibly only if the key is found in the array, or storing both the search key and the subscript. This could be done as follows:

```
10 AL = -999: REM Invalid key value
...
...
100 IF AR = AL THEN I = IL: GOTO 140
110 A(0) = AR: I = AE
120 IF A(I) <> AR THEN I = I-1: GOTO 120
130 IL = I: AL = AR :REM Store key and subscript
140 IF I = 0 THEN PRINT "Key not found"
    ELSE PRINT "Key found at location:"; I
```

where the stored key value is initially set to an invalid value to force a search on the first execution. The "search" here will be very fast if the same key value is used two or more times in succession, without any rearrangement of the array. A similar technique can be used if the array is known to be in sequence, but in this case a non-sequential search should probably be used. However, linear linked lists can only be searched sequentially, and therefore the technique could be used for these cases, although it will be shown here both for an ordinary array and for a linked list.

An example of a linked list which is searched using this technique is a program in Basic, where each line starts with a pointer to the start of the following line. Whenever a GOTO or GOSUB is used, the interpreter first checks whether the required line number is lower than or equal to the current line number. If so the search will start from the beginning of the program, otherwise it will start with the beginning of the current line. Thus a GOTO to the following line number is very fast while a GOTO to the same line number will be slower, especially towards the end of a long program. It is unfortunate that Microsoft did not use the opportunity to make the GOTO to the same line faster. It would have been simple to do, and would have made a lot of program loops very fast. Perhaps some versions of Basic do test for this, and for this reason I have attempted to keep the main loops of the algorithms as single lines. The technique for an unlinked array can be illustrated with the following code:

```
100 IF IL = 0 OR AR > A(IL) THEN I = AE ELSE I = IL
110 A(0) = AR
120 IF A(I) <> AR THEN I = I-1: GOTO 120
130 IL = I
```

In this case only the subscript needs to be saved, and it need not be initialised since Basic clears the value of all numeric data to zero, which is the value expected if the previous key was not found. The array is assumed to be in ascending sequence with the lowest key value having the lowest subscript value. Thus if the new key is larger than the previous one then the search has to start again at the end of the table, otherwise it can proceed from the point where it stopped on the previous iteration.

5.1.2 Linked Lists

Linked lists could be used in Basic to allow for a large number of insertions without having to move the remainder of the array to make room for the new elements. This would be useful for text information to avoid having to reorganise string memory too often. A linked list for text items would require two separate one dimension arrays, one for the links

which should be integer values, and one for the text items. The link array could be defined as DIM L(100) and the text array as DIM T(100). The link array could be initialised to contain the value -1 in each element, where -1 is used as an invalid pointer value to indicate the end of the list. A pointer is usually required to point to the first element in the list. The following algorithm could be used to find a particular text item TR in the text array.

```
10 LS = -1 :REM Empty list
...
...
100 I = LS :REM link start
110 IF I >= 0 THEN IF T(I) <> TR THEN I = L(I): GOTO 110
120 IF I < 0 PRINT "Text not found"
    ELSE PRINT "Text found at position:"; I
```

The values in the first 10 elements of the array L could be 1, 3, -1, 4, 7, -1, 8, 6, 5, -1 and the value of LS is zero. The text array would be referenced in the sequence 0, 1, 3, 4, 7, 6, 8, with termination at element 5. This may seem to be a complicated sequence, but similar sequences would only occur after a number of insertions or deletions from the linked list. No text or array elements need to be moved in order to insert a new text array element, so this function should be fast.

Several of the array search techniques presented earlier could also be used with the linked list, but the technique of inserting the required key into the array would be more difficult to implement. Some linked lists also use a variable which points to the last used element in the list (as in Basic which uses this pointer as the start address of the simple variables). A list can also be linked in both directions so that a search can be made in both descending and ascending sequences. This can be useful for sorted lists where the sequence of the items can be used to restrict the range of the searches.

5.1.3 Binary Trees

Binary trees are also a special form of linked lists, and since the word binary seems to relate them to computers, it has often been stated that they are well suited to computer implementation. They are suitable for implementation on computers for two reasons. The number of pointers in each element is limited to a left pointer and a right pointer, and the elements can therefore be of a fixed format, unlike the general tree where there may be any number of subordinate elements and therefore pointers in each record. The other reason is that pointers are easy to implement, and the link from one element to the next is simple to follow. All elements in a tree can be visited once in what is known as a tree traversal, for example if the values in all the elements are to be summed. Each element in a tree requires two pointers, one to the left node or leaf and one to the right node or leaf. By common convention the first or left pointer will point to a node with a lower key value, and the second or right pointer will point to a node with a higher or equal key value. Binary trees are usually drawn in the opposite form from real trees with the root element at the top and with the branches and leaves pointing downwards. Perhaps binary trees originated in the southern hemisphere!

A binary tree consisting of 7 elements with the key values from 1 to 7 could be drawn as follows:

```
            0  3 2 1
  2  1 -1 4                1  5 6 3
     4  2 -1 -1        6  4 -1 -1       3  6 -1 5
                                          5  7 -1 -1
```

The first value is the relative position in the array (from 0 to 6), the second is the key value (from 1 to 7), the third is the left pointer value and the fourth is the right pointer value. Note that the value -1 is used again to indicate the last pointer in a sequence. The tree is built so that the left pointer will point to an element with a key value lower than the value in the current element, and the right pointer will point to an element with a higher key value. Again a pointer to the root node is required to start the search. This tree is not easy to traverse in sequence since a pointer must be stored for each new level in the tree in order to find the way back when a leaf node is reached. The tree is obviously not stored in the manner shown in the diagram since memory is usually linear instead of two dimensional. The array element sequence would be

```
0  3  2  1
1  5  6  3
2  1 -1  4
3  6 -1  5
4  2 -1 -1
5  7 -1 -1
6  4 -1 -1
```

In this case all elements have been filled, but in practice empty spaces would be possible in the list if deletions could be made. In order to simplify traversal a number of techniques are available for manipulating the pointers. Since the range of negative numbers is not required they will often be used in the right pointer to point to the next element in the tree with a higher key value, after complementing or negating the pointer value. In this case zero could be used as a pointer terminator, leaving element zero in the array unusable since it cannot be directly complemented to yield a negative value. If space considerations made it imperative to use element zero, then the pointers could be stored as the actual value plus 1 (for the positive pointers), thus being 1 or -1 for a pointer to element zero, 2 or -2 for the pointer to element 1, etc. Using normal pointers our tree would get the following appearance:

```
            1 3 3 2
  3 1 0 5                2 5 7 4
     5 2 0 -1        7 4 0 -2     4 6 0 6
                                     6 7 0 0
```

To traverse the tree, the left branch is followed from the root node until the left pointer is zero. This is the element with the lowest key value and thus the actual starting point of the traversal. From here left branches are followed if a node

continued on page 47

FOUR BASIC PROGRAMS

Gordon Speer

KALEIDOSCOPE

I have always wanted to write a program like this, but one of our regular readers sent one that is super. If you aren't old enough to remember a kaleidoscope you just missed out, you probably can't buy them any more. The images drawn by this one resemble the floor plans for old European cathedrals, or perhaps a Rorschach test for engineers, or patterns for Indian rugs, or styrofoam packing material for sewing machine boxes. Try it yourself and see what you see in them. Continuous graphics programs always make an interesting demonstration when there are people passing by. Use them in store windows, or showcases, or on your desk.

Thanks to the Zeliff's for the program. Steve is a teacher at the high school in Minden, Nebraska.

```
100 'KALEIDOS
110 CLS :FOR DELAY=1 TO 1000:NEXT
120 A=RND(57)+2
130 B=RND(11)+17
140 L=INT(RND(59-A))+4
150 W=INT(RND(28-B))+2
160 FOR Y=B+W TO B STEP -1
170 X=A+L
180 GOSUB 380
190 X=X+1
200 GOSUB 380
210 NEXT Y
220 FOR X=A+L-1 TO A+1 STEP -1
230 GOSUB 380
240 NEXT X
250 FOR Y=B TO B+W
260 X=A
270 GOSUB 380
280 X=X-1
290 GOSUB 380
300 NEXT Y
310 FOR X=A+1 TO A+L-1
320 GOSUB 380
330 NEXT X
340 N=N+1
350 IF N=10 THEN S=1
360 IF N=20 THEN RUN
370 GOTO 120
380 '   SUBROUTINES
390 IF S=0 THEN 450
400 RESET(X,Y)
410 RESET(127-X,Y)
420 RESET(X,51-Y)
430 RESET(127-X,51-Y)
440 RETURN
450 SET(X,Y)
460 SET(127-X,Y)
470 SET(X,51-Y)
480 SET(127-X,51-Y)
490 RETURN
500 'STEVE AND SANDY ZELIFF
510 'MINDEN, NEBRASKA 68959
```

GASCOST and PERMILE

As the prices of motor fuel fluctuate and we all consider whether or not to trade in the old buggy, it is helpful to have some numbers at hand to help make the big decision. These two programs, excellent examples of charts, are for just that purpose. If you have a printer, change all the PRINTs to LPRINTS, or use your screen dump if you have one. Otherwise just read the video display. GASCOST gives the total cost of gasoline for a year of driving. Note that you save a lot more moving up from 15 MPG to 20 MPG than you do from 40 MPG to 50 MPG. In general the more fuel efficient the cars get, the less you stand to save by improving them. PERMILE gives the gasoline cost per mile driven, which is generally less than the depreciation or the insurance cost. (See "Car Expenses," *H & E Computronics*, February 1980, page 529.)

```
100 'GASCOST
110 CLS:CLEAR 500
120 INPUT"PRICE PER GALLON";P
130 CLS
140 PRINT,"COST OF GASOLINE FOR A YEAR OF DRIVING"
150 PRINT"       ==================== MILES DRIVEN
=====================
160 PRINT"        8000     10000     12000     14000
    16000     18000"
170 PRINT"MPG"
180 FOR MPG=10 TO 65 STEP 5
190 PRINT USING"## ";MPG;
200 FOR M=6000 TO 16000 STEP 2000
210 PRINT USING"$$#,###.##";M/MPG*P;
220 NEXT M
230 IF MPG=65 THEN 230
240 PRINT                        'LINE RETURN ONLY
250 NEXT MPG

100 'PERMILE
110 CLS:CLEAR 500
120 PRINT,"FUEL COST PER MILE OF DRIVING"
130 PRINT"       ============= FUEL COST PER GALLON
================"
140 PRINT"       1.20    1.40    1.60    1.80    2.00
   2.20    2.40"
150 PRINT"MPG"
160 FOR MPG=10 TO 65 STEP 5
170 PRINT USING"## ";MPG;
180 FOR C=1.20 TO 2.41 STEP .20  '*** DIDN'T WORK WITH 2.40
190 PRINT USING"    .###";C/MPG;
200 NEXT C
210 IF MPG=65 THEN 210
220 PRINT                        'LINE RETURN ONLY
230 NEXT MPG
```

BALLOON

One evening last June, just after sunset, we were driving around the Chicago suburbs and saw a hot air balloonist practicing landings. It must be quite an art adjusting the

temperature of the balloon with a giant gas torch to keep the buoyancy just neutral. Thinking about it brought back memories of gas law problems from chemistry class. Remember the alphabetic gas law inventors: Avogadro, Boyle, Charles, and Dalton? Actually Boyle's law was invented by Townley and plagiarized by Mariott, but to those of us in the midwest it's Boyle's law, and Mariott is an amusement park.

A hot air balloon rises when the weight of the air displaced is greater than the total weight of the balloon and the air inside. The weight of the balloon is rather constant, although as the propane is burned it would tend to get lighter. The weight of the displaced air will depend on its volume, temperature and pressure. At standard conditions of average sea-level pressure and freezing point of water temperature each liter of air weighs 1.29 grams. The weight of the contained air depends on the same factors. The pressure inside is always the same as outside because the balloon is open at the bottom, unless you get real technical and insist that it is the higher pressure of the warm air inside that makes the balloon rise. For purposes of calculating buoyancy, this pressure difference is negligible. To make the calculation straightforward we will assume the balloon is spherical, and evenly heated. The actual case is not so ideal.

```
100 '   BALLOON
110 CLS
120 PI=3.14159
130 'DENSITY OF AIR IS 1.29 GRAMS/LITER STP
140 PRINT:PRINT:PRINT
150 PRINT"THIS PROGRAM COMPUTES THE BUOYANCY, OR LIFTING
FORCE"
160 PRINT"OF A HOT AIR BALLOON AT VARIOUS TEMPERATURES"
170 PRINT
180 INPUT"EMPTY WEIGHT OF THE BALLOON (POUNDS)";FW
190 INPUT"DIAMETER OF THE BALLOON (FEET)";DIA
200 INPUT"BAROMETRIC PRESSURE (INCHES OF MERCURY)";P
210 INPUT"OUTSIDE AIR TEMPERATURE (FAHRENHEIT)";F
220 M=FW*453.6                  'MASS OF BALLOON IN GRAMS
230 DIA=DIA*12*2.54             'DIAMETER OF BALLOON IN CM
240 R=DIA/2                     'RADIUS OF BALLOON IN CM
250 V=4/3*PI*R↑3                'VOLUME IN CC
260 V=V/1000                    'VOLUME IN LITERS
270 ATM=P/29.92                 'PRESSURE IN ATMOSPHERES
280 C=5/9*(F-32)                'CELSIUS TEMPERATURE
290 K=C+273                     'KELVIN OR ABSOLUTE TEMPERATURE
300 D=1.29*ATM*273/K            'DENSITY OF DISPLACED AIR
310 M1=D*V                      'MASS OF DISPLACED AIR
320 CLS
330 PRINT"       INSIDE TEMPERATURE (F)   BUOYANCY (LB)"
340 FOR F2=95 TO 160 STEP 5     'INSIDE TEMPERATURE
350 C2=5/9*(F2-32)              'CELSIUS TEMPERATURE INSIDE
360 K2=C2+273                   'KELVIN TEMPERATURE INSIDE
370 D2=1.29*ATM*273/K2          'DENSITY OF INSIDE AIR
380 M2=D2*V                     'MASS OF INSIDE AIR IN GRAMS
390 B=M1-(M+M2)                 'BUOYANCY IN GRAMS
400 B=B/453.6                   'BUOYANCY IN POUNDS
```

continued on page 56

H&E COMPUTRONICS INC.

● ● ● EVERYTHING FOR YOUR TRS-80* ● ● ●
MODEL I, MODEL II & MODEL III

* **TRS-80™** is a trademark of Tandy Corp.

- All orders processed within 24 Hours
- 30-Day money back guarantee
- Add $3.00 for shipping in UPS Areas
- Add $4.00 for C.O.D. or NON-UPS Areas
- Add $5.00 to Canada or Mexico
- Add exact postage to all other countries

FROM

RACET COMPUTES LTD.

*** ESSENTIAL UTILITY PROGRAMS FOR EVERY TRS-80 OWNER ***

Facts About Racet Computes Utility Programs

*** ALL PROGRAMS ARE WRITTEN IN MACHINE LANGUAGE
*** ABSOLUTELY NO KNOWLEDGE OF MACHINE LANGUAGE IS NECESSARY TO USE ANY OF THE UTILITY PROGRAMS
*** EACH UTILITY PROGRAM IS CALLED UP FROM **BASIC** USING THE SIMPLE BASIC COMMANDS PROVIDED
*** EACH UTILITY PROGRAM COMES WITH A **RACET COMPUTES** INSTRUCTION MANUAL
*** EACH INSTRUCTION MANUAL INCLUDES SEVERAL EXAMPLES OF UTILITY USAGE
*** EACH UTILITY ALLOWS THE USER TO PERFORM CERTAIN BASIC OPERATIONS TEN, TWENTY OR MORE TIMES FASTER THAN THE EQUIVALENT BASIC ROUTINE (FOR EXAMPLE, **GSF** CAN SORT AN ARRAY OF 1000 RANDOM NAMES INTO ALPHABETICAL ORDER IN UNDER 9 SECONDS!!)

GSF (GENERALIZED SUBROUTINE FACILITY)

- SORTS 1000-ELEMENT ARRAYS IN 9 SECONDS
- SORTS UP TO 15 ARRAYS SIMULTANEOUSLY (MIXED STRING, FLOATING POINT AND INTEGER)
- SORTS SINGLE OR MULTIPLE SUBSTRINGS AS ASCENDING OR DESCENDING SORT KEYS
- READ AND WRITE ARRAYS TO CASSETTE
- COMPRESS AND UNCOMPRESS DATA IN MEMORY
- MOVE ARRAYS IN MEMORY
- DUPLICATE MEMORY
- FAST HORIZONTAL AND VERTICAL LINES
- SCREEN CONTROLS FOR SCROLLING THE SCREEN UP, DOWN, LEFT, RIGHT AND FOR GENERATING INVERSE GRAPHIC DISPLAYS
- ADDS PEEKS AND POKES (MOD-II VERSION ONLY)

MODEL-I VERSION $25.00
MODEL-II VERSION $50.00
MODEL-III VERSION $30.00

KFS-80 (KEYED FILE SYSTEM)

- CREATE ISAM FILES (INDEX SEQUENTIAL ACCESS METHOD)
- ALLOWS INSTANT ACCESS TO ANY RECORD ON YOUR DISKETTE
- INSTANTLY RETRIEVE RECORDS FROM MAILING LISTS, INVENTORY, ACCOUNTS RECEIVABLE OR VIRTUALLY ANY APPLICATION WHERE RAPID ACCESS IS REQUIRED TO NAMED RECORDS
- PROVIDES THE BASIC PROGRAMMER THE ABILITY TO RAPIDLY INSERT OR ACCESS KEYED RECORDS IN ONE OR MORE DATA FILES
- RECORDS ARE MAINTAINED IN SORTED ORDER BY A SPECIFIED KEY
- RECORDS MAY BE INSERTED OR RETRIEVED BY SUPPLYING THE KEY
- RECORDS MAY BE RETRIEVED SEQUENTIALLY IN SORTED ORDER
- RAPID ACCESS TO ANY FILE REGARDLESS OF THE NUMBER OF RECORDS
- MULTIPLE INDEX FILES CAN BE EASILY CREATED WHICH ALLOWS ACCESS OF A SINGLE DATABASE BY MULTIPLE KEYS (FOR EXAMPLE, BY BOTH NAME AND ZIP-CODE)

MODEL-I VERSION $100.00
MODEL-II VERSION $175.00
MODEL-III VERSION $100.00

DSM (DISK SORT MERGE)

- SORT AN 85K DISKETTE IN LESS THAN THREE MINUTES!
- SORTS LARGE MULTIPLE DISKETTE FILES ON A MINIMUM ONE DRIVE SYSTEM
- ALL RECORDS ARE PHYSICALLY REARRANGED-NO KEY FILES ARE REQUIRED
- SORTS RANDOM FILES CREATED BY **BASIC,** INCLUDING FILES CONTAINING SUB-RECORDS SPANNING SECTORS
- SORTS ON ONE OR MORE FIELDS IN ASCENDING OR DESCENDING ORDER
- FIELDS MAY BE STIRNGS, INTEGER, BINARY INTEGER OR FLOATING POINT
- THE SORTED OUTPUT FILE MAY OPTIONALLY HAVE FIELDS DELETED, REARRANGED OR PADDED
- SORT COMMANDS CAN BE SAVED FOR REUSE
- SINGLE SORT, MERGE, OR MIXED SORT/MERGE OPERATIONS MAY BE PERFORMED
- SORTED OUTPUT MAY BE WRITTEN TO A NEW FILE, OR REPLACE THE ORIGINAL INPUT FILE.

MODEL-I VERSION $75.00
MODEL-II VERSION $150.00
MODEL-III VERSION $90.00

MAILLIST (A MAILING LIST DATABASE SYSTEM)

- IDEALLY SUITED FOR ORGANIZATION MAILING LISTS, PERSONAL ADDRESSBOOK, OR MAILING LISTS BASED ON DATES SUCH AS REMINDERS FOR BIRTHDATES OR DUES PAYABLE
- USED ISAM (INDEX SEQUENTIAL ACCESS METHOD) FOR RAPID ACCESS TIMES
- YOUR **MAILLIST** CAN ALWAYS BE SORTED AND MAINTAINED BY UP TO FOUR INDEX FILES (FOR EXAMPLE, NAME, ZIPCODE, DATE AND NUMBER)
- **MAILLIST** ALLOWS UP TO 30 ATTRIBUTES TO BE SPECIFIED (TO BE USED IN SELECTION OF SPECIFIED RECORDS WHEN GENERATING REPORTS OR MAILING LABELS
- **MAILLIST** SUPPORTS BOTH 5 OR 9-DIGIT ZIPCODES
- PRINTING MAY BE STARTED OR ENDED AT ANY POINT IN THE LIST... THE USER CAN SPECIFY FIELDS OR CODES TO BE PRINTED
- CAPACITY IS 600 NAMES FOR MODEL-I, 3500 NAMES FOR MODEL II, 38,000 NAMES FOR MODEL II WITH HARD DISK DRIVE 1200 NAMES FOR MODEL III

MODEL-I VERSION $75.00
MODEL-II VERSION $150.00
MODEL-III VERSION $75.00

THE COMPUTRONICS INC.

HSDS HARD DISK DRIVE SOFTWARE

- MAKES TRSDOS COMPATIBLE WITH MOST HARD DISK DRIVES
- ADDS MANY EXTRA FEATURES TO TRSDOS $400.00

MODEL II FASTBACK — FULL DISK BACKUP IN 55 SECONDS

IN BUSINESS TIME IS MONEY, AND ONE BACKUP IS WORTH A THOUSAND TEARS.

- WORKS ON SYSTEMS WITH 2 OR MORE DRIVES
- CAN REPLACE YOUR EXISTING TRSDOS 1.2 or 2.0 BACKUP UTILITY

MODEL II ONLY $75.00

COMPROC (COMMAND PROCESSOR)

- AUTO YOUR DISK TO PERFORM ANY SEQUENCE OF INSTRUCTIONS THAT YOU NORMALLY GIVE FROM THE KEYBOARD (FOR EXAMPLE, INSERT THE DISKETTE, PRESS THE RESET BUTTON, YOUR COMMAND FILE COULD AUTOMATICALLY SHOW YOU THE DIRECTORY, SHOW THE FREE SPACE ON THE DIKSETTE, LOAD A MACHINE LANGUAGE SUBROUTINE, LOAD BASIC, LOAD AND RUN A BASIC PROGRAM, AND SELECT A GIVEN ITEM ON YOUR MENU...ALL WITHOUT TOUCHING THE KEYBOARD!)

MODEL-I VERSION $20.00
MODEL-III VERSION $30.00
NOT AVAILABLE FOR MODEL-II

DISCAT (DISKETTE CATALOG SYSTEM)

- THIS COMPREHENSIVE DISKETTE CATALOGUING/INDEXING UTILITY ALLOWS THE USER TO KEEP TRACK OF THOUSANDS OF PROGRAMS IN A CATEGORIZED LIBRARY...FILE INCLUDES PROGRAM NAMES AND EXTENSIONS, PROGRAM LENGTH, DISKETTE NUMBERS AND FREE SPACE ON EACH DISKETTE...KEEP A COMPLETE CATALOG OF THE DIRECTORIES ON ALL YOUR DISKETTES IN ALPHABETICAL ORDER (SORTED ON EACH DISKETTE...OR COMPLETE ALPHABETICAL LIST OF PROGRAMS ON ALL YOUR DISKETTES)

MODEL-I VERSION $50.00
MODEL-III VERSION $50.00
MODEL-II VERSION (SEE MODEL-II UTILITY PACKAGE)

BLINK (BASIC LINK FACILITY)

- LINK FROM **BASIC** PROGRAM TO ANOTHER SAVING ALL VARIABLES
- THE CHAINED PROGRAM MAY EITHER REPLACE THE ORIGINAL PROGRAM OR CAN BE MERGED BY STATEMENT NUMBER

MODEL-I VERSION $25.00
MODEL-III VERSION $30.00
MODEL-II VERSION (SEE MODEL-II UTILITY PACKAGE) $50.00

INFINITE BASIC

- ADDS OVER 80 COMMANDS TO BASIC
- SORTING...STRING CENTERING/ROTATION/TRUNCATION...JUSTIFICATION...DATA COMPRESSION...STRING TRANSLATION/COPYING...SCREEN DISPLAY...SCROLLING...MATRIX OPERATIONS...SIMULTANEOUS EQUATIONS (THROUGH MATRIX INVERSION)...DYNAMIC ARRAY RESHAPING

MODEL-I VERSION $50.00
MODEL-III VERSION $60.00
NOT AVAILABLE ON MODEL-II

INFINITE BUSINESS

- ADD ON PACKAGE TO **INFINITE BASIC** (REQUIRES INFINITE BASIC)
- ADDS PACKED DECIMAL ARITHMETIC WITH 127 DIGIT ACCURACY (+,0,*,/)
- COMPLETE PRINTER PAGINATION CONTROLS...AUTO HEADERS, FOOTERS AND PAGE NUMBERS
- BINARY SEARCH OF SORTED AND UNSORTED ARRAYS (INSTANT SEARCH OF AN ELEMENT WITHIN AN ARRAY)
- HASH CODES

MODEL-I VERSION $30.00
MODEL-III VERSION $30.00
NOT AVAILABLE ON MODEL-II

REMODEL-PROLOAD

- THE ULTIMATE RENUMBERING PROGRAM...RENUMBERS ALL OR PART OF A PROGRAM (ALLOWS PARTIAL RENUMBERING IN MIDDLE OF PROGRAMS)
- PARTIAL OR COMPLETE MERGE OF TWO CASSETTE PROGRAMS

MODEL-I VERSION $35.00
MODEL-III VERSION $35.00
NOT AVAILABLE ON MODEL-II

COPSYS

- COPY AND VERIFY ALL MACHINE LANGUAGE (SYSTEM) TAPES WRITTEN IN STANDARD FORMAT...IF YOU BUY A MACHINE LANGUAGE PROGRAM, **COPSYS** ALLOWS YOU TO EASILY COPY THE PROGRAM ONTO ANOTHER CASSETTE AS A BACKUP

MODEL-I VERSION $15.00
MODEL-III VERSION $20.00
NOT AVAILABLE ON MODEL-II

MODEL-II UTILITY PACKAGE

- ESSENTIAL FOR EVERY MOD-II OWNER
- RECOVER AND REPAIR FILES AND DIRECTORIES (BY JUST ENTERING A SINGLE COMMAND)
- XCOPY...SIMILAR TO COPY BUT CAN COPY ANY NUMBER OF FILES AT ONE TIME FASTER AND MORE ACCURATE THAN COPY SINCE RECORDS ARE COPIED IN GROUPS RATHER THAN ONE RECORDS AT A TIME...USING XCOPY YOU CAN COPY FILES THAT CAN NOT BE COPIED USING THE COPY COMMAND
- SZAP...PROVIDES THE CAPABILITY TO READ AND MODIFY ANY SECTOR ON A DISKETTE
- XHIT...CAN BE USED TO REPAIR A DISKETTE DIRECTORY
- DCS...DIRECTOR CATALOG SYSTEM IS A UTILITY FOR THE MANAGEMENT OF USER DISKETTES...SETS OF A MULTIPLE DISKETTE DIRECTORY FILE (WITH UP TO 1200 INDIVIDUAL FILE NAMES)...ALLOWS SELECTIVELY LISTED OR PRINTED LISTS OF DIRECTORY FILES IN COMBINED SORTED ORDER (FOR EXAMPLE, LISTED ALPHABETICALLY BY DISKETTE...OR A COMPOSITE ALPHABETICAL LIST OF ALL YOUR DISKETTES!)
- DEBUG-II...ADDS SEVERAL FEATURES TO THE PRESENT TRSDOS DEBUG UTILITY INCLUDING SINGLE INSTRUCTION CYCLE, AUTO (LOOP) BREAKPOINTS, SUBROUTINE CALLING, BREAK-KEY DETECTION AND MANY OTHERS

MODEL-II ONLY $150.00

MODEL-II DEVELOPMENT SYSTEM

- THIS PACKAGE IS A MUST FOR ASSEMBLY LANGUAGE PROGRAMMERS
- INCLUDES THE **MICROSOFT EDITOR ASSEMBLER PLUS** WITH ENHANCEMENTS FOR THE MODEL-II
- A COMPLETE DISASSEMBLER
- SUPERZAP FOR READING AND MODIFY ANY SELECTOR ON A DISKETTE

MODEL-II ONLY $125.00

MOD-II BASIC CROSS REFERENCE UTILITY

- LIST OR PRINT A SORTED CROSS REFERENCE TO ALL NUMBERS OR VARIABLES WITHIN A PROGRAM
- LIST OF PRINT ALL LINE NUMBERS CONTAINING A SPECIFIED STRING OF CHARACTERS

MODEL-II ONLY $50.00

***** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE *****

END USERS CALL:

THE COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SYSTEMS

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

VISA AMERICAN EXPRESS MasterCard

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

24 HOUR ORDER LINE
(914) 425-1535

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA AND MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO

FOR DEALER INFORMATION CALL:

RACET COMPUTES

1330 N. GLASSEL, SUITE M, ORANGE CA 92667
(714) 997-4950

ASSEMBLY LANGUAGE FOR BEGINNERS (PART 8)

Joseph Rosenman

This Month: Let's print the Alphabet!

```
Label   Command Argument
        ORG     7000H
DISP    EQU     0033H   ;ROM subroutine to display character.
CLR     EQU     01C9H   ;ROM subroutine to clear the screen.
DOS     EQU     402DH   ;Address to return to DOS.
;
PROG1   CALL    CLR     ;Clear the screen.
        LD      A,41H   ;Get an ASCII 'A'.
        LD      B,26    ;Number of letters in the alphabet.
;
LOOP    CALL    DISP    ;Display the character.
        INC     A       ;A now contains the next ASCII letter
        DEC     B       ;One less letter to go.
        JR      NZ,LOOP ;The Dec set the flags.
;
        JP      DOS     ;All done, so exit program.
        END     PROG1

7000  CD C9 01 3E 41 06 1B CD 33 00 3C 05 20 F9 C3 2D
7010  40 .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
```

The memory dump shows what this program looks like in RAM. Note that the program is only 11H bytes long (17 decimal). To emphasize that point, I entered double dots (..) to show that those values don't matter. Now, examining the program should reveal several unknown instructions. ORGs, EQUs, LDs, INCs, DECs, ENDs, that's all old hat (see how much you know already?). So what is a JR, a JP, and a CALL? Not to worry, here they are:

JP: Mnemonic for JUMP. This instruction causes the program to go to the address indicated in the argument field. There are two kinds of jumps: conditional and unconditional. An unconditional jump means "Do it, no matter what." A conditional jump means "do it only if the indicated conditions are met." Consider the following three instructions:

```
        JP      VECTOR1
        JP      Z,VECTOR2
        JP      NZ,VECTOR3
```

The first jump is an unconditional jump. There is no provision for any other action to be performed. The second jump (to VECTOR2) is a conditional jump. This means that if (and only if) the status is zero, will the jump take place. If not zero, the jump is ignored. The third jump (to VECTOR3) is also a conditional jump. In this case, the jump only takes place if the status is not zero. OK, I know there are at least two questions floating around. Question 1: how did the status get set? Refer to your trusty (and priceless) Z-80 instruction reference guide (published in the September 1982 issue of *Computronics*). As you scan through the tables, you will see that certain instructions cause the CPU status to be set or reset (remember, the status is contained in the F register). Now the DEC B instruction will set the status flag. Since B contains the maximum number of characters, and is clicking down every time a character is displayed, it will decrement to 0 when a Z is displayed. Every time that B is NOT 0, the jump on not zero will take place. When B finally does reach 0, the conditional jump is ignored. The next instruction happens to be an unconditional jump. This jump goes to a special address in DOS that, in effect, says "hello, I'm back!" By jumping to this location, you exit your program and return to the operating system. By the way, for those of you running BASIC only systems, you should replace "JP DOS" with "JP 0000H".

Question 2: What is a JUMP? Good question. Consider this: how does the program know where to execute your program? When it is loaded into memory (from disk or tape), it must be placed into a specific address (in this case, at 7000H). After the program is loaded (in fact, the very last thing that can be done), the PC register is loaded with the address of the program starting address (7000H). Remember, the PC (Program Counter) controls what the computer will do next. Every time an instruction is "fetched" from memory, the PC is incremented anywhere from 1 to 4 bytes, so that it points to the next instruction. When a jump instruction is executed, the PC contents are replaced with the address that is contained in the argument field (remember, labels are also addresses in symbolic form). In the case of a conditional branch, the PC is only replaced if the condition is met. If the condition is not met, nothing happens. Since the PC is all ready pointing to the next instruction, the program just goes on as it was.

I bet you are all snickering up your sleeves! "The silly fool has a JR instead of a JP in the first jump." Well, JR stands for Jump Relative (would you believe me if I told you that they named this instruction after me?). A jump relative acts just like a jump, with two special differences: (1) it is only two bytes long instead of three bytes, and (2) it can only jump 126 bytes back or 129 bytes forward (a regular jump can jump to any address in memory). Now that this question of jumping is settled, let's follow a program trace.

```
PC=7000  A=??  B=??   CALL CLR
PC=7003  A=??  B=??   LD   A,41H
PC=7004  A=41  B=??   LD   B,1BH
PC=7005  A=41  B=1B   CALL DISP      -> LOOP <-
PC=7008  A=41  B=1B   INC  A
PC=7009  A=42  B=1B   DEC  B   (flag NZ)
PC=700A  A=42  B=1A   JR   NZ,LOOP
PC=7005  A=42  B=1A   CALL DISP
PC=7008  A=42  B=1A   INC  A
PC=7009  A=43  B=1A   DEC  B   (flag NZ)
PC=700A  A=43  B=19   JR   NZ,LOOP
PC=7005  A=43  B=19   CALL DISP
PC=7008  A=43  B=19   INC  A
PC=7009  A=44  B=19   DEC  B   (flag NZ)
PC=700A  A=44  B=18   JR   NZ,LOOP
```

The three CALLs to DISP displayed the letters A, B, and C. The program would continue on as above until it reached the letter Z. Let's pick up our trace during the display of the letter Y.

```
PC=7005   A=59  B=02    CALL DISP
PC=7008   A=59  B=02    INC  A
PC=7009   A=5A  B=02    DEC  B  (flag NZ)
PC=700A   A=5A  B=01    JR   NZ,LOOP
PC=7005   A=5A  B=01    CALL DISP
PC=7008   A=5A  B=01    INC  A
PC=7009   A=5B  B=01    DEC  B  (flag Z)
PC=700A   A=5B  B=00    JR   NZ,LOOP
PC=700C   A=5B  B=00    JP   DOS
PC=402D
```

As you can see in the second trace segment, the B register decrements to 0, the jump relative fails, then the unconditional jump to DOS is executed. Remember that the action of a jump instruction is to place the target address in the PC? Take note of the fact that after the JP DOS is executed the PC contains the number 402D. Looking at the Equates at the start of the program, you will notice that the symbol DOS and the number/address 402D were "Equated." In the trace, you will see that the PC address following the JR (except for the last one) all contain the address of the instruction associated with "LOOP," the CALL DISP.

Now that I've mentioned it, what IS a CALL? Time for a new instruction, and a new register. For those of you who know a higher level language, CALL is the machine language equivalent of a Subroutine. In fact, Call calls a subroutine. Consider the action of call in this way: you are performing some task (task A). Suddenly, you need to stop performing task A, and start performing task B. You note the current state of task A, then perform task B. After task B is completed, you will recall where you left off in task A, and pick it up from that point. Here is an example of a useless program that will illustrate the CALL instruction:

```
Label   Command Argument        Comment
        ORG     5200H
START   LD      A,0             ;Put a zero in A.
LOOP    INC     A               ;A=1
        INC     A               ;A=2
        INC     A               ;A=3
        CALL    UNDO            ;Make the subroutine call.
        JP      LOOP            ;Do it again for no  reason.
;
UNDO    DEC     A               ;A=2
        DEC     A               ;A=1
        DEC     A               ;A=0
        RET                     ;Go back to the calling prog. ;
        END     START
```

This is an example of an "endless loop," because the program never stops. For the sake of clarity, let's trace it through two passes.

```
5200  3E 00 3C 3C 3C CD 0B 52 C3 02 52 3D 3D 3D C9 ..
5210  .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..

PC=5200   SP=8000  A=??    LD   A,00
PC=5202   SP=8000  A=00    INC  A        -> Loop <-
```

continued on page 38

PRACTICAL BUSINESS PROGRAMS

CASH FLOW ANALYSIS FOR REAL ESTATE AND OTHER INVESTMENTS

S. M. Zimmerman, Ph.D. and L. M. Conrad

This program is designed to calculate the cash flows resulting from a real estate or other type investment and then to calculate a measure of the present worth of the cash flows based on the classic internal rate of return concept. The specific approach used is called the present worth analysis.

The model is as accurate as we could make it, but it is still a model. All models have approximations and assumptions, ours is no exception. Establish procedures for using the model, then examine the results when using the model on completed projects. Review the decisions you made with the decision the model recommends. If necessary adjust your procedures for using the model until it becomes a useful predictor, then use it on a trial basis. The output can only be as good as the data you input.

One local investor has been using this model for about 10 months. The approach has saved him money by helping reject poor projects, although errors do occur. One of 13 business deals failed to behave as predicted. An "after the fact" examination of the investment project indicated the poor results came from inputting faulty information into the model.

A track record of 12 out of 13 or 92.3% + is good for any decision making procedure. Compare this record to yours. Do you have any use for a program? Even if your track record is better than ours you still may want to look at this program.

Investment analysis is not a simple activity. You cannot expect a simple answer to a complex problem. If you need a first approximation to an investment situation then the break-even program we published in Feburary 1980 in this magazine may be the best approach for you. If you need an accurate and detailed approach to decision making then this program may be better.

SAMPLE APPLICATION

The sample application is based on a real estate investment I involving a 44 unit apartment complex. It has the potential ability to earn $96,000 per year. The vacancy rate is estimated to be approximately 5%. In order to purchase this complex $100,000 cash is needed. A 25 year mortgage for $600,000 to cover the remaining cost will be issued by the seller at 10% for the first five years, then escalates to 12% thereafter. The estimated selling price of the investment in 10 years is $700,000.

When the complex is taken over, initial repair costs are estimated to be $15,000 to bring the units up to the acceptable quality. Annual maintenance costs are estimated to be $25,000. Based on the Economic Recovery Act of 1981 a table look up method of depreciation may be used for this type of investment. This method is built into the program along with other methods of depreciation. A land value of 10% of the investment was used.

Time should be spent in choosing the study period used in your analysis. It can make a difference. A 10 year period was selected because it balances the value of annual cash flows with the value of the resale cash of a project of the type under consideration.

An internal rate of return of 15% will be used for this analysis. This number has nothing to do with the cost of capital above. It is a number which represents the expectation of local investors.

Inflation is an emotional subject to all investors. This program forces you to consider inflation as just one more item to be forecast. It will show you the consequences of different inflation estimates. We usually make at least two evaluations, one at zero inflation and one at 6% inflation.

For this example we have assumed that the evaluation is being made for an individual in the 50% tax bracket.

RUNNING THE PROGRAM

Let's run our example through the computer model. Since we have reviewed each step, we will show the input and output as produced by the computer with a minimum amount of comment.

```
CASH FLOW PROGRAM
DEVELOPED BY:
     STEVEN M. ZIMMERMAN,PH.D.
 &   LEO M. CONRAD       1982

PERCENTS AS WHOLE NUMBERS

TITLE :?  SAMPLE
DATE : (XX/XX/XX) ? Ø3/Ø5/82
MAXIMUM YEARS? 1Ø
```

The screen will clear and the input of data continues:

```
INPUT REGULAR INCOME
INPUT MENU REGULAR INCOME
   C  CONSTANT
   S  SPECIFIED YEAR BY YEAR
   R  REDO ALL REGULAR INCOME
   E  END CYCLE
```

The earnings were specified as a constant in our sample write up. We select C for CONSTANT earnings. The next question is:

```
(M) MONTHLY INPUT OR (A) ANNUAL INPUT?
```

Since the information was given as an annual amount we select A. The next question is:

```
          YEARLY INCOME?
```

The yearly income for our sample problem was $96,000. We answered the above question with this number. The ($) and (,) were left out. After this question is inputted we are returned to the INPUT REGULAR INCOME menu.

The program will now return you to the INPUT MENU.

```
INPUT REGULAR INCOME
INPUT MENU REGULAR INCOME
   C  CONSTANT
   S  SPECIFIED YEAR BY YEAR
   R  REDO ALL REGULAR INCOME
   E  END CYCLE
```

This time we answer E since there is no additional income in our problem. The next question is:

```
VACANCY FACTOR (NOTE: APPLIED TO ALL VALUES INPUTED)?
```

We did not specify whether the $96,000 income was net or gross. We will assume that it is gross and enter a five for an assumed 5% vacancy factor.

The next question is:

```
INPUT CAPITAL INCOME YEAR BY YEAR
YEAR AND AMOUNT.. Ø,Ø WHEN COMPLETE?
```

Capital income must be put in year by year. In our example we assume the selling price in 10 years is to be equal to the purchase price of $700,000. Type 10,700000 as the answer to the above question. The question now repeats itself:

```
YEAR AND AMOUNT.. Ø,Ø WHEN COMPLETE?
```

This time the question should be answered with a 0,0 so as to terminate this series of questions.

The next thing you will see is:

```
INPUT YEARLY COSTS MENU
   C    CONSTANT
   S    SPECIFIED YEAR BY YEAR
   R    REDO ALL YEARLY COSTS
   E    END CYCLE?
```

This menu is similar to the regular income menu. In this case we must use option (C) and option (S). Selecting option (C) first we are asked:

```
INPUT YEARLY COST?
```

The estimate for yearly cost was $25,000. After answering (25000) to the above question we will be returned to the INPUT YEARLY COSTS MENU. This time we must answer (S) so we can input the initial cost of $15,000 which must be spent for clean up and start up. After inputing (S) we will see on the CRT:

```
YEAR AND AMOUNT.. Ø,Ø WHEN COMPLETE?
```

The answer we must give to get the $15,000 into its proper year is (1,15000). Since this is the only unusual cost we now may enter (0,0) and return to the INPUT YEARLY COSTS MENU.

Since this is the last of the yearly cost information we enter (E) to move to the next menu.

```
DEPRECIATION MENU
   L     STRAIGHT LINE
   D     DECLINING BALANCE
   S     SUM OF DIGITS
   R     REDO ALL DEPRECIATION CALCULATIONS
   T     TABLE LOOKUP 1981
   E     END CYCLE?-
```

The Economic Recovery Act of 1981 has a very specific depreciation method to be used for all real estate. This method has been built into our program. All the traditional methods are also built in. Type T and hit ENTER to obtain the new table lookup depreciation as per the 1981 Economic Recovery Act.

The depreciation question and anawers are shown:

```
YEAR OF EXPENSE (COUNT Ø AS START YEAR)? Ø
CAPITAL EXPENSE ? 7ØØØØØ
SALVAGE VALUE (LAND VALUE)? 7ØØØØ
LOW INCOME HOUSING Y/N? N
```

The assumption was land value was equal to 10" of the price of the investment and the unit under study was not a low income housing investment.

After you have completed the above you will return to the depreciation menu. This time input E to end the cycle.

The next menu is the mortgage menu:

```
MORTGAGE MENU
   P   EQUAL PAYMENTS
   D   EQUAL REDUCTION
   I   INTEREST ONLY
   R   REDO ALL MORTGAGES
   E   END CYCLE?
```

An equal payment loan may be input by typing P and hitting ENTER. The next question and answer are:

```
BALLOON YEAR (1ØØØ IF NONE)? 1ØØØ
```

The next question is a series of questions. It is shown with our answers:

```
INPUT Ø FOR PAYMENT IF YOU WISH PROGRAM TO CALCULATE
START YEAR, LOAN, APR, PAYMENTS PER YEAR, PAYMENT, YEARS
Ø,6ØØØØØ,1Ø,12,Ø,25
```

The program now returns to the mortgage menu. Continue by typing E and hit the ENTER key.

The next question is:

```
1       INDIVIDUAL
2       CORPORATE
SELECTION?
```

The tax treatment of individuals is very different from corporations relative to capital gains. We will assume the analysis is for an individual and type 1 followed by ENTER. The next question is:

```
TAX    RATE?
```

A tax rate of 50% will be assumed. Input 50 and hit ENTER. The next question is:

```
OUTPUT MENU
    C   CRT
    P40  PRINTER 40 COLUMNS
    P80  PRINTER 80 COLUMNS
    P132 PRINTER 132 + COLUMNS?
```

We needed the output formatted in the above manner for different situations. All have been included so you have a choice. Select P40 if you wish to match the output we will show.

The next question is:

```
LINE FEED (Y/N)?
```

If you have an operating system such as DOS PLUS which has a built in line counter, this statement allows you to line up your paper. If you do not have such a system you will not be able to make use of this feature. Type N and hit ENTER assuming you do not have such a system.

The next question is:

```
NUMBER OF YEARS OF STUDY <= 10 ?
```

It is sometimes desirable to have study periods less than the maximum. This question allows you to do this. We answered 10 and hit ENTER.

The next question is:

```
INTERNAL RATE?
```

The discount rate was called the internal rate of return in the program. You may search for an internal rate of return if you wish or you may enter one discount rate for all your studies. We like the discount approach which yields a net present value. For our sample run the required internal rate of return discount rate was 15%. Type 15 and hit ENTER.

The next question and answer is:

```
INFLATION MENU
   S   SINGLE RATE ALL FACTORS
   I   INDIVIDUAL RATES-REG. INCOME/CAPITAL INCOME
       REGULAR COSTS/CAPITAL COSTS? S
```

The final question and answer is:

```
INFLATION?  0
```

The program now produces the following results:

```
TITLE  :SAMPLE INFLATION 0
DATA : 03/05/82  INTERNAL RATE .15
INDIVIDUAL

YEAR  REGULAR  CAPITAL  OTHER  CAPITAL
      INCOME   INCOME   COSTS    COSTS
  0        0        0      0   700000
  1    91200        0  40000        0
  2    91200        0  25000        0
  3    91200        0  25000        0
  4    91200        0  25000        0
  5    91200        0  25000        0
  6    91200        0  25000        0
  7    91200        0  25000        0
  8    91200        0  25000        0
  9    91200        0  25000        0
 10    91200   700000  25000        0

YEAR  DEPRECIATION  MORTGAGE   INTEREST  REDUCTION
      EXPENSE       CASH       EXPENSE   LOAN
  0         0   600000             0      0
  1     75600        0         59744   5683
  2     63000        0         59149   6278
  3     56700        0         58492   6935
  4     50400        0         57766   7661
  5     44100        0         56963   8463
  6     37800        0         56077   9350
  7     37800        0         55098  10329
  8     37800        0         54017  11410
  9     37800        0         52822  12605
 10    259000        0         51502  521286

YEAR  TAXABLE   TAX        TAX        CASH
      INCOME    RATE       PAID       FLOW
  0         0    0.00          0    -100000
  1    -84144    0.50          0     -14227
  2    -55949    0.50          0        773
  3    -48992    0.50          0        773
  4    -41966    0.50          0        773
  5    -34863    0.50          0        773
  6    -27677    0.50          0        773
  7    -26698    0.50          0        773
  8    -25617    0.50          0        773
  9    -24422    0.50          0        773
 10    -16802    0.50      94500      98912
DISCOUNT CASH-84905
DISCOUNT TAX SAVINGS-115243
DISCOUNT CASH AND TAX SAVINGS   30338.3
```

The theory used is the traditional cash flow theory, or cookie jar theory. All that is important is what is left in the cookie jar at the end of the year. After all expenses have been paid including the tax bureau you simply count what is left. The program is designed with this concept in mind. All positive and negative cash flows are identified. The sum of the positive cash flows less the sum of all the negative cash flows yields the end of year cash flow.

The program uses four types of inputs for cash flow: Regular incomes & expenses and capital incomes and expenses. From this basic input plus information about the investor is calculated the tax consequences, the taxable income, the cash flow for each year. The present worth discounting procedure is then used to determine an overall measure of the yearly cash flows.

Calculation of cash flows require detailed knowledge of tax laws and constant update of the internal programming procedure when tax laws change. We are not accountants. We have had the aid of several accountants in designing various aspects of the program. If you have unusual tax

problems, use the results of the analysis only in close coordination with your accountant. Congress can and will change our tax laws. The model may be outdated as these changes occur.

```
1Ø CLEAR 3ØØ:REM  "CASH"
2Ø CG=.4Ø:GC=1-CG:REM  1981 CAPITAL GAINS RATE
3Ø GG$="GO":GX$="GO":GA$="GO":GB$="GO":GC$="GO":
DIM TAX(17,3),ZD(15)
4Ø CLS:PRINT CHR$(23):PRINT "CASH FLOW PROGRAM"
5Ø PRINT "Copyright"
6Ø PRINT "    STEVEN M. ZIMMERMAN,PH.D. & LEO M. CONRAD   1982"
7Ø C$="## ######## ######### ######### ######## ##########
######### ######## ######## ########### ###.## ########
#########   ###########"
8Ø CD$="## ######### ######### ######### ####.## #########
########"
9Ø CC$="## ######### ######### ####.## ######### #########
 #########"
95 CX$="## ######### ####.## ######### #########  #########"
1ØØ DIM A$(26): A$(2)="REGULAR  ":A$(3)="CAPITAL   "
A$(4)="OTHER   "
1Ø3 A$(5)=A$(3):A$(6)="DEPRECIATION ":A$(7)="MORTGAGE  ":
A$(8)="INTEREST  "
1Ø5 A$(9)="REDUCTION  ":A$(1Ø)="TAXABLE ":A$(11)="TAX       "
1Ø6 A$(13)="CASH      ":A$(15)="INCOME   ":A$(16)=A$(15)
11Ø A$(17)="COSTS     ":A$(18)=A$(17):A$(19)="EXPENSE      ":
A$(2Ø)=A$(13):A$(21)="EXPENSE   "
115 A$(22)="LOAN      ":A$(23)=A$(16):A$(24)="RATE      "
116 A$(26)="FLOW     ":A$(1)="YEAR ":A$(14)="     ":
A$(12)="TAX      ":A$(25)="PAID    "
12Ø PRINT:PRINT"PERCENTS AS WHOLE NUMBERS": PRINT:
B1$="TITLE :"
125 B2$="DATE :": PRINT B1$;: INPUT T$: PRINT B2$;
"(XX/XX/XX)";
127 INPUT D$:INPUT "MAXIMUM YEARS";G:DIM Z(G,13),X(5Ø):CLS
13Ø PRINT "INPUT REGULAR INCOME"
14Ø PRINT "INPUT MENU REGULAR INCOME"
15Ø IF GG$="GO" THEN FOR I=1 TO G:Z(I,1)=Ø:NEXT I:GG$="STOP"
16Ø PRINT "   C  CONSTANT"
17Ø PRINT "   S  SPECIFIED YEAR BY YEAR"
18Ø PRINT "   R  REDO ALL REGULAR INCOME"
19Ø INPUT "   E  END CYCLE";IX$: IF IX$="E" THEN 31Ø
2ØØ IF IX$="R" THEN GG$="GO": GOTO 14Ø
21Ø IF IX$<>"C" THEN 28Ø
22Ø REM CONSTANT INCOME
23Ø INPUT "(M) MONTHLY INPUT OR (A) ANNUAL INPUT";L$
24Ø IF L$="M" THEN INPUT "MONTHLY INCOME";Y:Y=Y*12
25Ø IF L$="A" THEN INPUT"YEARLY INCOME";Y
26Ø FOR I=1 TO G:Z(I,1)=Y+Z(I,1):NEXT
27Ø GOTO 14Ø
28Ø INPUT "YEAR AND AMOUNT.. Ø,Ø WHEN COMPLETE";I,Y
29Ø IF I=Ø AND Y=Ø THEN 14Ø
3ØØ Z(I,1)=Y+Z(I,1): GOTO 28Ø
31Ø INPUT "VACANCY FACTOR (NOTE APPLIED TO ALL VALUES
INPUTTED)";VF:VF=.Ø1*VF
32Ø FOR I=1 TO G:Z(I,1)=(1-VF)*Z(I,1): NEXT
33Ø REM CAP INCOME
34Ø IF GC$<>"GO" THEN 149Ø ELSE PRINT "INPUT CAPITAL INCOME
YEAR BY YEAR"
35Ø FOR I=1 TO G:Z(I,2)=Ø:NEXT
```

```
360 INPUT "YEAR AND AMOUNT.. Ø,Ø WHEN COMPLETE";I,Y:Z(I,2)=Y
370 IF I=Ø AND Y=Ø THEN 390
380 GOTO 360
390 IF GC$="STOP" THEN 1490
400 PRINT "INPUT YEARLY COSTS MENU"
410 IF GX$="GO" THEN FOR I=1 TO G:Z(I,3)=Ø:NEXT:GX$="STOP"
420 PRINT"   C   CONSTANT"
430 PRINT"   S   SPECIFIED YEAR BY YEAR"
440 PRINT"   R   REDO ALL YEARLY COSTS"
450 INPUT"   E   END CYCLE";IX$:IF IX$="E" THEN 550
460 IF IX$="R" THEN GX$="GO":GOTO 400
470 IF IX$="S" THEN 510
480 INPUT"INPUT YEARLY COST";C
490 FOR I=1 TO G:Z(I,3)=Z(I,3)+C:NEXT
500 GOTO 400
510 REM YR SPEC
520 INPUT"YEAR AND AMOUNT.. Ø,Ø WHEN COMPLETE";I,Y
530 IF I=Ø AND Y=Ø THEN 400
540 Z(I,3)=Z(I,3)+Y:GOTO 520
550 IF GC$<>"GO" THEN 1490
560 PRINT"DEPRECIATION MENU"
570 IF GA$="GO" THEN FOR I=Ø TO G:Z(I,4)=Ø: Z(I,5)=Ø: NEXT:
FOR I=1 TO 50:X(I)=Ø:NEXT:GA$="STOP":D5=Ø
580 FOR I=1 TO 50:X(I)=Ø:NEXT I
590 PRINT"   L   STRAIGHT LINE"
600 PRINT"   D   DECLINING BALANCE"
610 PRINT"   S   SUM OF DIGITS"
620 PRINT"   R   REDO ALL DEPRECIATION CALCULATIONS"
630 PRINT"   T   TABLE LOOKUP 1981"
640 INPUT"   E   END CYCLE";IX$: IF IX$="E" THEN 1090
650 IF IX$="R" THEN GA$="GO": GOTO 560
660 W$(1)="YEAR OF EXPENSE (COUNT Ø AS START YEAR)"
670 W$(2)="CAPITAL EXPENSE"
680 W$(3)="SALVAGE VALUE (LAND VALUE)"
690 W$(4)="LIFE"
700 D(5)=1
710 IF IX$="D" THEN D(5)=2
720 IF IX$="T" THEN D(5)=5
730 IF IX$="S"THEN D(5)=3
740 W$(7)="RATE"
750 W$(8)="DEPRECIATION"
760 D(8)=100
770 FOR I=1 TO 50:X(I)=Ø:NEXTI
780 D(4)=15: PRINT "INPUT FOR NEXT ASSET ": FOR I=1 TO 3:
PRINT W$(I);: INPUT D(I): NEXT I: IF IX$<> "T" THEN PRINT
W$(4);: INPUT D(4)
790 Z(G,5)=Z(G,5)+D(3):Z(D(1),4)=Z(D(1),4)+D(2):D(1)=D(1)+1
800 IF D(5)=2 THEN PRINT W$(7);: INPUT D(7):D(7)=D(7)*.01
810 IF D(5)=1 THEN I=1:GOTO 940
820 IF IX$="T" THEN 980
830 IF D(5)=2 THEN 880
840 REM SYD
850 SD=(D(4)*(D(4)+1))/2:BV=D(2)
860 FOR I=1 TO D(4) : DX=((D(4)-(I-1))/SD)*(BV-D(3)) :
SU=SD+DX : D(2)=D(2)-DX : K%= I : X(I)=DX
870 D(1)=D(1)+1 : NEXT I : GOTO 1050
880 REM DECL
890 DR=D(7)/D(4)
900 FOR I=1 TO D(4) : DX=DR*D(2) : SX=(D(2)-D(3))/(D(4)-I+1):
IF D(3)>(D(2)-DX) THEN DX=D(2)-D(3): GOTO 920
910 IF SX>DX THEN 940
920 X(D(1))=X(D(1))+DX: SU=SU+DX: D(2)=D(2)-DX
930 D(1)=D(1)+1: NEXT I
940 REM STR
950 IF (D(4)-I+1)>0 THEN DX=(D(2)-D(3))/(D(4)-I+1) ELSE 960
960 FOR J=I TO D(4): SU=SU+DX: D(2)=D(2)-DX:
X(D(1))=DX+X(D(1))
970 D(1)=D(1)+1: NEXT J: GOTO 1050
980 REM 1981 LAW
990 INPUT "LOW INCOME HOUSING (Y/N)";D7$: IF D7$="Y" THEN FOR
I=1 TO 15: READ JU:NEXT I
1000 FOR I=1 TO 15: READ ZD(I): NEXT I
1010 DATA 12,10,9,8,7,6,6,6,6,5,5,5,5,5,5:REM REG
1020 DATA 13,12,10,9,8,7,6,5,5,5,4,4,4,4,4:REM LOW
1030 RESTORE
1040 FOR I=1 TO 15: X(I)=ZD(I)*(D(2)-D(3))/100: NEXT I
1050 D5=D5+X(G): FOR I=1 TO G:Z(I,5)=Z(I,5)+X(I): NEXT I
1060 KK=G: IF KK>D(4) THEN KK=D(4)
1070 IF D(4)>G THEN S=Ø: FOR I=G+1 TO D(4): S=S+X(I): NEXT I:
Z(G,5)=Z(G,5)+S
1080 GOTO 560
1090 IF GB$<>"GO" THEN 1490
1100 PRINT"MORTGAGE MENU"
1110 IF GB$="GO" THEN GB$="STOP" : FOR I=Ø TO G :
Z(I,8)=Ø:Z(I,6)=Ø:Z(I,7)=Ø:NEXTI
1120 PRINT"   P   EQUAL PAYMENTS"
1130 PRINT"   D   EQUAL REDUCTION"
1140 PRINT"   I   INTEREST ONLY"
1150 PRINT"   R   REDO ALL MORTGAGES"
1160 INPUT"   E   END CYCLE";IX$: IF IX$="E" THEN 1490
1170 X2$="START YEAR, LOAN, APR, PAYMENTS PER YEAR, PAYMENT,
YEARS"
1180 IF IX$="R" THEN GB$="GO": GOTO 1100
1190 IF IX$="P" THEN INPUT "BALLOON YEAR (1000 IF NONE)";BA
1200 IF IX$<>"P" THEN 1340
1210 PRINT "INPUT Ø FOR PAYMENT IF YOU WISH PROGRAM TO
CALCULATE"
1220 PRINT X2$: INPUT L,LO,APR,PP,PAY,YEAR :Z(L,6)=Z(L,6)+LO:
APR=.01*APR
1230 R=APR/PP : IF PAY=Ø THEN II=(1+R)↑(PP*YEAR) :
FF=(R*II)/(II-1) : PAY=FF*LO
1240 N=YEAR*PP : KK=G : IF YEAR<G THEN KK=YEAR
1250 J=Ø:JL=Ø
1260 J=J+1:S=Ø: SS=Ø: IF BA+1=J THEN YY=BA+L: GOTO 1320
1270 IF JL>=G THEN YY=KK: GOTO 1320
1280 FOR I=1 TO PP
1290 II=R*LO:RED=PAY-II:LO=LO-RED
1300 S=S+II:SS=SS+RED
1310 NEXTI : YY=KK : JL=J+L : Z(JL,7)=Z(JL,7)+S :
Z(JL,8)=Z(JL,8)+SS : IF J<KK THEN 1260
1320 IF LO>0 THEN Z(YY,8)=Z(YY,8)+LO
1330 GOTO 1100
1340 IF IX$<>"D" THEN 1420
1350 PRINT "START YEAR, LOAN, APR, REDUCTION PER YEAR, YEARS"
: INPUT L,LO,APR,RED,YEAR : Z(L,6)=Z(L,6)+LO:APR=.01*APR
1360 PP=1:KK=G:IF YEAR<KK THEN KK=YEAR
1370 FOR I=L+1 TO KK : II=LO*A : LO=LO-RED : Z(I,7)=Z(I,7)+II
: Z(I,8)=Z(I,8)+RED: IF LO<Ø THEN RED=RED-LO :
Z(I,8)=Z(I,8)-LO : GOTO 1390
1380 NEXT I
1390 IF KK=YEAR THEN KK=KK-1
1400 IF LO>0 THEN Z(KK+1,8)=Z(KK+1,8)+LO
```

```
1410 GOTO 1100
1420 REM I/O
1430 IF IX$<>"I" THEN 1100
1440 PRINT"START YEAR, LOAN, APR, YEARS" : PP=1 : INPUT
L,LO,APR,YEAR : Z(L,6)=Z(L,6)+LO:APR=APR*.01
1450 KK=G: IF KK>YEAR THEN KK=YEAR
1460 FOR I=L+1 TO KK:Z(I,7)=Z(I,7)+APR*LO: NEXT I
1470 Z(KK,8)=Z(KK,8)+LO
1480 GOTO 1100
1490 GC$="STOP"
1500 CLS:C$(1)="INDIVIDUAL":C$(2)="CORPORATE"
1505 FOR I=1 TO 2 : PRINT I,C$(I) : NEXT I :B3$="SELECTION" :
PRINT B3$; : INPUT SS
1820 PRINT A$(11),A$(24);: INPUT TR:TR=TR*.01: FOR I=1 TO G:
Z(I,10)=TR: NEXT I
1830 IF SS<1 OR SS>3 THEN 1500
1840 PRINT"OUTPUT MENU"
1850 PRINT"    C      CRT"
1855 PRINT"    P40     PRINTER 40 COLUMNS"
1860 PRINT"    P80    PRINTER 80 COLUMNS"
1870 INPUT"    P130   PRINTER 130+ COLUMNS";IX$
1880 IF IX$="C" THEN OP=1
1890 IF IX$="P80" THEN OP=2
1900 IF IX$="P130" THEN OP=3
1905 IF IX$="P40" THEN OP=4
1906 IF OP>1 THEN INPUT "LINE FEED (Y/N)";QQ$: IF QQ$="Y"
THEN CMD"FORMS(T)"
1910 PRINT "NUMBER OF YEARS OF STUDY <=";G; : INPUT YS :
H$="INTERNAL RATE" : PRINT H$; : INPUT R : H1$="INFLATION" :
R=.01*R
1920 IF YS>G THEN 1910
1930 PRINT "INFLATION MENU"
1940 PRINT "   S   SINGLE RATE ALL FACTORS"
1950 PRINT "   I   INDIVIDUAL RATES-REG. INCOME/CAPITAL
INCOME"
1960 INPUT "       REG. COSTS/CAPITAL COSTS";IX$
1970 IF IX$="I" THEN 2000
1980 PRINT H1$;:INPUT M:M=.01*M
1990 FOR I=1 TO 4:IM(I)=M:NEXTI:GOTO 2050
2000 REM INFLATE
2010 INPUT"INPUT INFLATION RATE FOR REGULAR INCOME";
IM(1):M=-9999
2020 INPUT"INPUT INFLATION RATE FOR CAPITAL INCOME";IM(2)
2030 INPUT"INPUT INFLATION RATE FOR REGULAR COSTS";IM(3)
2040 INPUT"INPUT INFLATION RATE FOR CAPITAL COSTS";IM(4):
FOR I=1 TO 4:IM(I)=IM(I)*.01: NEXT I
2050 IF OP>1 THEN LPRINT" ": IF M>=0 THEN LPRINT B1$;T$;" ";
H1$;M ELSE LPRINT B1$;T$;" ";H1$;IM(1);IM(2);IM(3);IM(4)
2060 IF OP>1 THEN LPRINT B2$;D$,H$;R:LPRINT C$(SS):LPRINT" "
2080 IF OP>1 AND SS=3 THEN LPRINT VV$(ST)
2085 IF OP=4 THEN FOR I=1 TO 5: LPRINT A$(I);: NEXT: LPRINT :
FOR I=14 TO 18 : LPRINT A$(I);: NEXT : LPRINT
2090 IF OP=3 THEN FOR I=1 TO 13: LPRINT A$(I); : NEXTI :
LPRINT "ADJUSTED" : FOR I=14 TO 26 : LPRINT A$(I); : NEXTI :
LPRINT "$ FLOW"
2100 IF OP=2 THEN FOR I=1 TO 8: LPRINT A$(I); : NEXTI :
LPRINT : FORI=14 TO 21: LPRINT A$(I);: NEXT I: LPRINT
2110 IF OP=1 THEN FOR I=1 TO 6: PRINT A$(I);: NEXT I: PRINT
A$(7);: FOR I=14 TO 19: PRINT A$(I);: NEXT I: PRINT A$(20)
2120 RR=Z(0,6)-Z(0,4):IF OP=3 THEN LPRINT USING C$;
0,0,0,0,Z(0,4),0,Z(0,6),0,0,0,0,0,RR,RR
```

continued on page 46

BEGINNER'S CORNER

Spencer Koenig

Lend an Ear to the Big Boys (Part II of a series)

Hidy ho and say hey! This is to be the last episode on a subject I didn't realize would be so immense. BBS's are all the rage, and well they should be. The doors for information exchange and processing at relatively inexpensive rates are just getting opened. If you are into playing with the latest in technology for pleasure or business (like the newest stereos, the latest software, etc.), then I'm sure you can understand the excitement I feel in getting this chance to tinker with these systems (and for free too!). I feel like the kid in the toy store and someone says "have a good time, kid"!

This month ends the series with a small discussion on THE SOURCE. I say "small discussion" because, if you saw all the materials that you get and the amount of information they contain, you could see how an in depth look at THE SOURCE could become a magazine by itself. In fact, let's start with who, how much, and what you get.

THE SOURCE is a subsidiary of Reader Digest, as I'm sure most of you know. The actual company that THE SOURCE belongs to is Source Telecomputing Corporation.

To get an account up on this service requires a $100 registration fee and an hourly charge that varies from $18 during peak business hours to as little as $4.25 after midnight. There is a monthly minimum of $10.00. The system itself is basically six Prime 750 mainframes located somewhere in Mclean, Virginia (1616 Anderson Road, to be exact).

At the beginning of this article I said that there is so much information about this service that you could dedicate a whole magazine to the subject. The fact is, SOURCEWORLD is just such an item. It is published for those on the system and deals with the ever changing facilities and system up-dates. This is included as part of the package.

In the issue I received, the topics covered were very interesting. Volume 2 issue 3's first subject dealt with getting degrees in areas such as biomedical sciences, solar engineering, and electronics, by taking classes on the BBS. These are two-year Associate degrees, accredited by the North Central Association of Colleges and Schools and the Accreditation Board for Engineering and Technology. I'm involved at the present time with education (I teach in an intermediate school in Oceanhill Brownsvlle in Brooklyn New York), and this kind of news is very important for everyone. It shows that there's a big change comming in the field of education.

The subject matter in SOURCEWORLD also covers bargain shopping by computer (see EXAMPLE 2), new menu options, new adventures on the system (my favorite), where to get explanations on how the system functions, and also a subject that will prove to become more important in this technological age: an on-line book ordering center that specializes in hard to find texts. It seems that they have some 40,000 plus titles held in inventory. That's quite a feat for any book store.

Included with the magazine is a user's manual that, from the advertisements, seems very extensive. What I like best of all in the documentation is a little pamphlet called the SOURCE DIGEST. It is essentially a directory of services available. This is a big difference from Compuserve, where searching for topics of interest is done while on-line (searching around costs you).

This way you can go through the guide at your leisure. One of the ideas suggested in some of the documentaion I received is to have your system boot up onto THE SOURCE and input the required information automatically (if your terminal package is capable of it). This could include what subject area you want.

The table of contents contains a multitude of subjects an extracted list can be found in EXAMPLE 1.

```
**********************************************************
EXAMPLE 1.

Chat-on-line conversations with other subscribers

Electronic mail- sending, receiving, scanning etc

Bulletin board/Classifieds-

----------------------------------------------------------
INFORMATION SERVICES
   BUSINESS NEWS, SERVICES AND FINANCIAL MARKETS

Business services- airline schedules, travel tips, restaurant
reviews, business ventures for source subscribers

Communicatins Industry-

Current Business Trends- 36 catagories of news and  commentaries.

Taxes- Tax tips (this could pay for itself) and computer based
calculations.

----------------------------------------------------------
Catalogue Shopping

Electronic shopping- lots of discounts

Trading and Bartering- Nationwide bartering and brokerage firm;
classified ads etc

----------------------------------------------------------
Education and Careers

This area contains information about education (language arts,
math, foreign languages, programming languages etc) financial
aid; instructional programs and employment opportunities.
----------------------------------------------------------
Government and Politics
```

```
------------------------------------------------------------
Home and Leisure

Consumer Affairs

Games- 75 electronic games

Health and medicine
------------------------------------------------------------
News, sports, features
------------------------------------------------------------
Science and Technology
------------------------------------------------------------
```

The listing continues in detail, but you get the general idea.

```
************************************************************
EXAMPLE2
      >>>WELCOME TO THE TRADENET<<<

    YOUR LINK TO BARTER---WORLDWIDE!!!

   COPYRIGHT 198Ø, BARTER WORLDWIDE INC.

>>TO TRADE APPLE HWRE FOR ART WRITE US DIRECT!  MAIL EX BARTER!<<

1-TO PLACE A LISTING.....TYPE P899

2-TO SEE LISTING FORMAT.....TYPE P223

3-TO SEE LATEST LISTINGS.....TYPE P1Ø3

4-TO SCAN BY KEYWORD.....TYPE PØØ1

5-READ THE WORLD OF BARTER...TYPE PØØ2

6-IF YOU ARE NEW TO TRADENET TYPE PØØ3

IF YOU HAVE A N Y QUESTIONS WRITE US SOURCE MAIL EX BARTER!!

     WELCOME TO THE TRADENET!  TO GAIN ACCESS TO THE VARIOUS
FILES OF THE NET, SIMPLY ENTER THE "P-CODE" FOR THAT FILE WHEN
PROMPTED.  FOR EXAMPLE, TO VIEW THE INDEX OF FILES YOU MAY READ
TO LEARN ALL OF THE AVAILABLE OPTIONS, SIMPLY TYPE IN 'PØØ3'
WHEN  PROMPTED.  THE "PROMPT" LOOKS LIKE THIS:

ENTER A SUBJECT, PARAGRAPH NUMBER OR "HELP" FOR INSTRUCTIONS.
     AFTER THE COMPUTER DISPLAYS THIS PROMPT IT WILL WAIT FOR
YOU TO TYPE IN SOMETHING. . .YOU MAY ENTER A KEYWORD THAT NAMES
WHAT IT IS YOU ARE LOOKING FOR.

     FOR EXAMPLE, IF YOU ARE INTERESTED IN CONDOMINIUM
TIMESHARING,
>>USING ALL CAPITAL LETTERS<< YOU COULD SIMPLY TYPE IN:  CONDO
OR EVEN A CHARACTER STRING SUCH AS  'TIMESH.'  THE SYSTEM WILL
RESPOND WITH A LIST OF THE APPROPRIATE CATEGORIES AND THE P-CODE
YOU MUST ENTER TO READ THE DETAILS OF THE LISTINGS FOR (IN THIS
CASE) CONDOS.
```

continued on page 36

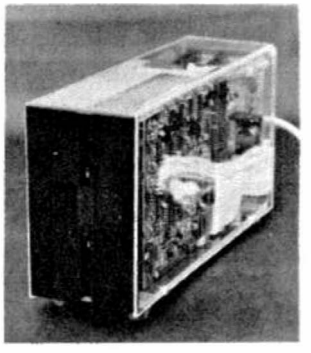
Disk Drives and Cases

The B.T. Enterprises Mini-Floppy 5-1/4 inch Disk Drive Case is unique in that it is constructed of clear plexiglass. This innovative design is the first on the market which enables the user to actually see the Disk Drives working while using the computer. The case is constructed of a 1/4 inch durable plexiglass base with a 1/16 inch clear plexiglass cover. The case and power supply are available for single or dual drives. One advantage of the dual design is that both floppy drives are built into one case for saving space and easier handling.

The unit is equipped with a linear filtered power supply that is regulated to match the power consumption of the mini drives. The power is AC fuse protected with a chasis mounting fuse on the back of the case for easy maintenance. An on/off switch is provided on the bacck of the case. The AC line cord is a standard 3 prong plug designed to meet UL listing.

FLOPPY DISK DRIVES

201100*	5-1/4 Disk Drive Case & Power Supply (Sgl)	$ 79.95
201200*	5-1/4 Disk Drive Case & Power Supply (Dbl)	$ 109.95
201101*	Tandon 40tk SH w/case & Supply	$ 299.95
201102*	Tandon 40tk DH w/case & Supply	$ 399.95
201103*	Tandon 80tk SH w/case & Supply	$ 399.95
201104*	Tandon 80tk DH w/case & Supply	$ 499.95
201201*	Dual Tandon 40tk SH w/case & Supply	$ 579.95
201202*	Dual Tandon 40tk DH w/case & Supply	$ 779.95
201203*	Dual Tandon 80tk SH w/case & Supply	$ 779.95
201204*	Dual Tandon 80tk DH w/case & Supply	$ 979.95
200300	Disk Drive Extender Cable	$ 9.95



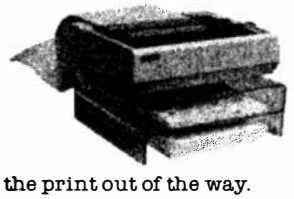
*Complete with extender cable

Bare Drives

200101	Tandon 40tk SH Bare	$ 259.95
200102	Tandon 40tk DH Bare	$ 359.95
200103	Tandon 80tk SH Bare	$ 359.95
200104	Tandon 80tk DH Bare	$ 459.95

Helpful Products for Your Computer

Printer Stands



End the paper mess on your computer desk. Our printer stand allows your paper to be fed from under the printer, making room for the used paper to stack behind the print out of the way. Available with an optional removable shelf (Shown) for easy computer forms change. Available in Large size also, for MX-100 and other large printer users also. Also available with center slot for bottom feed printers. (Large only).

Regular Stand	(300010)	$29.95
Regular w/shelf	(300011)	$44.95
Large Stand	(300020)	$34.95
Large w/shelf	(300021)	$49.95
Large w/slot	(300050)	$49.95

Diskettes

Elephant Memories Systems Diskettes at a special LOW price!!!

Single Density/Single Sided/Soft Sector	500351	$22.50
Double Density/Single Sided/Soft Sector	500352	$26.50
Double Density/Double Sided/Soft Sector	500353	$32.50

Portable Disk Bulk Eraser

Clean all old data off of your diskettes before reusing them! This is a MUST for your computer room. Easy to use, non-electric, portable!!!

500102 ..... Only ...$4.95

Ribbon Reloads for MX 70/80/100

Reload your old ribbon cartridge for much less than replacing the cartridge! Quick and easy to do, takes only about 2 minutes each to reload. Includes complete instructions. Available in Black and 4 popular colors!

Black Reload	(500000)	$3.95 ea.	(500001)	$39.95 dz.
Red Reload	(500010)	$4.95 ea.	(500011)	$49.50 dz.
Blue Reload	(500020)	$4.95 ea.	(500021)	$49.50 dz.
Green Reload	(500030)	$4.95 ea.	(500031)	$49.50 dz.
Brown Reload	(500040)	$4.95 ea.	(500041)	$49.50 dz.

Ribbon Cartridges for MX 70/80

Replacement cartridges for your printer. Complete cartridge for your printer at great savings! Available in Black and 4 popular colors! Buy in 3 packs and save!

Black Cartridge	(500050)	$10.95 ea.	(500051)	$29.95/3
Red Cartridge	(500060)	$11.95 ea.	(500061)	$31.95/3
Blue Cartridge	(500070)	$11.95 ea.	(500071)	$31.95/3
Green Cartridge	(500080)	$11.95 ea.	(500081)	$31.95/3
Brown Cartridge	(500090)	$11.95 ea.	(500091)	$31.95/3

BT ENTERPRISES

171 Hawkins Rd.
Centereach, N.Y. 11720
(816) 981-8868 (voice)
(816) 888-8836 (modem)
Dealer Inquires Welcome
Prices subject to change

master charge VISA

Welcome

N.Y.S. Residents Add Tax Add $2 shipping – handling

DON'T JUDGE A COMPUTER BY IT'S COVER!!!

...If you were to have a Microcomputer operating at 4MHZ
...If you were to have a 5 Megabyte Hard Drive built in
...If you were to have a floppy disk drive built in
...If you were to have an Amber Phospher Monitor built in
...If you had the finest Hard Disk Operating System available

You would have the New B.T. 4x5 Model III microcomputer, the only one in its class.

Starting with a basic TRS-80 Model III, add one Tandon 40 Track Dual Sided Disk Drive, 48K of High Grade NEC Memory, the famous Micro Mainframes Disk Controller System, and the security of B.T. Enterprises engineering assuring long life and dependability!

Then we really get started include a 4 MHZ Module installed to make the Model III Microprocessor operate at twice the speed of a standard Radio Shack Model III. Also include the use of a Superdense Disk Drive for extra Floppy disk storage (360K). Then have installed a 5 Megabyte Hard Disk Drive (Winchester Technology). When the system is configured, the Hard Disk Drive contains the operating system and becomes the master (primary) drive. The Hard Disk has 40 times the speed of data transfer as compared to a normal floppy disk as well as containing the storage capacity of more than 30 standard floppy disk drives.

Each unit is built and tested in our Long Island New York assembly plant and delivered to your dealer complete with a Dos Plus 4.0 Disk Operating System which we consider have the finest features of any Hard Disk Operating System available.

Price:

$3995.00

Add $20.00 shipping and handling

B.T. Hard Disks

We want you to have all the power that your TRS-80 is capable, at prices that have other manufacturers cross-eyed! These Five and Ten Megabyte units are easy to use, just plug them into the expansion buss of your Model I or Model III. Comes complete with the fantastic DOS PLUS 4.0 Operating System! Available in two configurations: 5 Megabyte Fixed, 10 Megabytes Fixed. Look at the incredible low prices of our systems!

SYSTEM PRICING

5 Megabyte Winchester Fixed Disk		
201505	Model I Version	$2795.00
203505	Model III Version	$2795.00
10 Megabyte Winchester Fixed Disk		
201510	Model I Version	$3195.00
203510	Model III Version	$3195.00

Add $10.00 shipping and handling

Upgrade Kit

Putting Disk drives into your TRS-80 Model III will turn it into the powerful computer it was designed to be. The B.T. Enterprises DISK DRIVE INSTALLATION KIT is easy to do and represents a substantial savings over the cost if done by Radio Shack.

The kit was designed to be installed by a non-technical person, and takes an average of 45 to 75 minutes to complete. The only tools necessary are a phillips and flat head screw driver and a pair of diagonal cutters. No cutting of traces or soldering is necessary. The full instruction procedure was the topic of a feature in 80 US Magazine in the May, 1982 issue, and was given high marks for ease of installation.

KIT CONSISTS OF:

- FDC III B* Micromainframe Controller Board assembled & tested
- Switching power supply
- Disk drive mounting brackets
- All cables and hardware
- Instructions with diagrams
- Drives are optional

*FDC III C Board optional at additional cost supports 8 inch external drives and extended density 5-1/4 inch drives (dual headed and 80 track)

203103	Disk Upgrade Kit B Version w/hardware	$279.95
203104	Disk Upgrade Kit C Version w/hardware	$329.95
203113	Disk Upgrade B Version w/2 40tk & 32K	$ 799.95
203114	Disk Upgrade C Version w/2 40tk & 32K	$ 849.95
203115	Disk Upgrade C Version w/2 40tk DH & 32K	$1049.95
203116	Disk Upgrade C Version w/2 80tk & 32K	$1049.95
203117	Disk Upgrade C Version w/2 80tk DH & 32K	$1249.95
203003	FDC III B Controller Board only	$ 139.95
203004	FDC III C Controller Board only	$ 224.95
200500	Switching Power Supply (3 voltages)	$ 99.95
203010	Controller Assembly Kit	$ 75.00
203510	Controller Assy kit w/power Supply	$ 174.95

DH — Dual Headed

BT ENTERPRISES

171 Hawkins Rd.
Centereach, N.Y. 11720
(516) 981-8568 (voice)
(516) 588-5836 (modem)
Dealer Inquiries Welcome
Prices subject to change
N.Y.S. Residents Add Tax

master charge
VISA
Welcome

Radio Shack is a trademark of Tandy Corp.

continued from page 33

```
     IF YOU ENTER P998 YOU WILL BE ABLE TO READ A DETAILED LIST
OF >>>ALL<<< OF THE VARIOUS COMMANDS WHICH ARE AVAILABLE ON THE
SYSTEM.

     IF YOU HAVE ANY DIFFICULTY WHATSOEVER IN UNDERSTANDING
THESE (Press return for more output) INSTRUCTIONS OR IN USING
THE TRADENET,  PLEASE LET US KNOW BY EXITING THE BARTER DATA BASE
(BY ENTERING 'STOP' IN CAPITAL LETTERS WHEN YOU SEE THE PROMPT)
AND THE IN COMMAND MODE TYPE THIS:

>MAIL EX BARTER

IF YOU DO NOT KNOW HOW TO USE THE SOURCE ELECTRONIC MAIL SERVICE,
CALL CUSTOMER SUPPORT AT 1-800-336-3330.
OR, AFTER YOU'VE EXITED THE BARTER DATA BASE BY TYPING 'STOP'
WHEN PROMPTED, YOU WILL SEE A "SYSTEM PROMPT" THAT LOOKS LIKE
THIS: >

WHEN YOU SEE THAT, YOU MAY TYPE IN
INFO MAIL INFO
AND PRESS RETURN...YOU WILL THEN RECEIVE INFORMATION ABOUT THE
MAIL SERVICE.

HAPPY TRADING!!!
************************************************************
```

One of any time sharing system's most appealing and used option is the ability to talk to other users who are on-line. When I was using the Queens College or Brooklyn college system, I often talked to fellow students, some of whom might have been located hundreds of miles away. On the SOURCE I had the pleasure of talking to two users. Some excerpts are in Examples 3 and 4.

```
************************************************************
EXAMPLE 3 (MY FIRST CONVERSATION)
>CHAT STC303 HI CARE TO CHAT I'M WRITING AN ARTICLE FOR A MAG

* STC303 user 3 SYS12 *

HI!

CHAT STC303 HI MY NAME SI SPENCER KOENIG I WRITE FOR COMPUTRONICS.
WHO ARE YOU AND WHERE ARE YOU?

Good morning! I'm Pat Trenner, and I work at the source.
What is Computronics?
************************************************************
I think he was just kidding (ha ha)
************************************************************
CHAT STC303 COMPUTRONICS IS A MAGAZINE FOR THE TRS-80.  WHAT DO
YOU DO FOR THE SOURCE?

I work in the editorial department...on documentation, stuff on
the system, and for cable products. Where are you located?

CHAT STC303 IM IN NYC. FLUSHING TO BE EXACT AND YOU?

Mclean, VA. By the way, you needn't preface each transmission
with "!!CHAT CHAT STC303", now that we are "CONNECTED"...

THAT'S GOOD TO KNOW IT ISN'T CLEAR IN THE DOCUMENTS I RECEIVED
OR IN THE INSTRUCTIONS IN THE HELP FILE.

Are you a brand-new SOURCE user?

IM USING THE SOURCE FOR A SERIES ON BBS AND I WAS GIVEN SOME
TIME ON THE SYSTEM .

AH...WELL, ENJOY! Is there any info I can give you that will
help your research?

NONE THAT I CAN THINK OF. I HAVE MOST OF WHAT I NEED . WHAT I
WANTED TO TO WAS INCLUDE A SHORT DISCUSSION WHICH  IS JUST WHAT
I HAVE NOW.

Good! Well, I'd better get back to printing out some UPI
stuff...call again whenever you have time!

OK  THANKS
>
Bye
************************************************************

EXAMPLE 4

>CHAT TCT112 HI CARE TO CHAT I'M WRITING FOR A MAGAZINE.
>
* TCT112 user 23 SYS11 *
Yeeeeeessss?##      (it's nice to see enthusiasm-S.K.)

CHAT TCT112 HI WHO ARE YOU? AND WHERE ARE YOU?

Allo allo?##

HI WHO ARE YOU?

whoops...Hi, who are you, too?  I am Maria.  To
save you the effort of typing in the next logical
question, I'm in Long Beach, CA.  And to save myself
the effort, where are you?  Are you a demo account?##

YES I'M A DEMO ACCOUNT, ALTHOUGH I DON'T LIKE REFERRING TO
MYSELF AS SUCH. I'M IN FLUSHING N.Y. AND I WRITE FOR
COMPUTRONICS MAGAZINE, WHICH IS FOR THE TRS-80 COMPUTER.
MY NAME IS SPENCER KOENIG.

Hi Spencer.  Sorry to have referred to you as a "demo account,"
but I couldn't imagine what else "DMO" might stand for, other
than Department of Motorized Olives, which didn't seem likely.
What do you write, for the magazine "Computronics?"##
************************************************************
```

Well, as you can see, the conversation could only get more interesting and it did. I intend to contact Maria on Compuserve, where we both maintain accounts. As it turns out she and has quite a number of interests such a computer controlled synthesizers, a husband who has a Buick dealership (I assume he's at least an interest) and a few other things that sounded time consuming and

profitable. My point here is that computers are no longer the loner-sits-at-home type of machine. They're the latest in how to get and maintain pen pals, business contacts, or any other type of human relationship you can conceive of.

To make this point, there was a section that dealt with dating that I happen to stumble across (see EXAMPLE 5).

```
**********************************************************
EXAMPLE 5

Whether you're buying or selling, looking for a new love
or just chattering, the POST database puts your message
online in a few easy steps!

 At the first prompt, "Read, Post, Purge, Scan, or HELP:"

Enter the coreading all the instructions by typing HELP and the
command name (Ex:

>POST3   DATING
DATING?
Do you need help? Not really

<R>ead,<PO>st,<PU>rge,<S>can, or <H>elp:S

<C>ategory,<U>ser ID,<D>ate,<K>eyword:C

Categories, or <H>elp:DATING
Searching...
38 notices valid.

<N>arrow, <E>xpand or Return for all :
Wait..
1     12 JUL ST2449 LOVE DOING IT?(DATING)
2     11 JUL CLØ48Ø FUN FLORIDA GUY(DATING)
3     11 JUL STØ331 MATCHMAKER #14(DATING)
4     11 JUL STØ331 MATCHMAKER #14(DATING)
5     11 JUL TI-Ø757 SWINGERS UNITE L.A.(DATING)
6     11 JUL TCV73Ø HELP, I JUST MOVED TO N.J.(DATING)
7     11 JUL ST32Ø2 POST SCAN DATING(DATING)
8     11 JUL ST32Ø2 SINGLES(DATING)
9     11 JUL TCLØ17 HEY HEY %%%%%(DATING)
1Ø    1Ø JUL TCK836 (DATING)
11    1Ø JUL ST3Ø58 NEVER TRIED THIS BEFORE!!!(DATING)
12    1Ø JUL STØ897 FLA.GUY WANTS TO MEET FLA.LADY(DATING)
13    1Ø JUL TCP528 BICOASTAL WOMAN AVAILABLE(DATING)
14    1Ø JUL TCP528 VERMONT VOYEUR(DATING)
15    1Ø JUL TCKØ85 DIAL-YOUR-MATCH NUMBERS(DATING)
16    1Ø JUL CL2372 IN SEARCH OF.....????(DATING)
17    1Ø JUL TCV34Ø SOUTHEAST LADIES(DATING)
18    1Ø JUL TI-Ø8Ø4 RAVISHING BLONDE(DATING)
19    1Ø JUL TI-Ø8Ø4 OK, SHAWNA IS BACK IN TOWN!!!(DATING)
2Ø    9 JUL STØØ48 GUYS WANTED!(DATING)
21    9 JUL STØØ48 ANY GIRLS OUT THERE ?(DATING)
22    9 JUL ST239Ø PITTSBURGH LOVER RETURNS !!(DATING)
23    9 JUL ST17Ø3 WANTED: HOUSTON GIRLS(DATING)
24    9 JUL TCZ762 ANY DENVER/BOULDER WOMEN??(DATING)
25    8 JUL ST239Ø MALE:SEEKING WILD WOMEN !!(DATING)
26    8 JUL ST33Ø3 SYSTEMS ANALYST PILOT MIDWEST(DATING)
27    8 JUL TCZ985 LOOKING FOR TALL AND TAN!!(DATING)
28    8 JUL ST13Ø6 MORE OF THE DWARVES(DATING)
```

```
29    8 JUL TCM090 LOOKING FOR CHAT FRIENDS(DATING)
30    7 JUL TCT593 WELL-RIPENED NY FEMALE . . .(DATING)
31    7 JUL IX0011 FEMALES WANTED(DATING)
32    7 JUL TCW890 LURLINE(DATING)
33    7 JUL ST1306 DWARVES!!!!(HEH HEH)(DATING)
34    6 JUL TI-0757 L.A. AREA AFTERNOONS(DATING)
35    6 JUL ST2449 NAUGHTY ? LADIES DARE TO BE NAUG(DATING)
36    5 JUL TCZ507 N.Y. STUDENT WANTS TO MEET SAME(DATING)
37    5 JUL ST1616 MALE 22 WANTS FEMALE OVER 18(DATING)
38    5 JUL ST1964 COMPLIMENTARY DRINKS AND DINNER(DATING)
************************************************************

EXAMPLE 6
Category:DATING
Subject:MORE OF THE DWARVES
From:ST1306
Posted:8 JUL 12:28 am


<N>ext,<PO>st, or Return for text-

HIHOHIHO!!!
          WE ARE SOME VARIOUS NUMBER
OF SHORT, EXTREMELY UNWASHED  DWARVES,
AND WE WANT TO MEET YOU!! WE ARE LOC
ATEDIN ST. LOUIS, MO., AND SHORTSVILLE N.J.
(OUR FEMALE MEMBER IS IN N.J.)  IF YOU
WANT TO JOIN US, OR JUST CHAT, WE ARE
AT ST1306(ST. LOUIS) OR TCX706(N.J.)

                    SMELLY
                ACTING SUPREME DWARF
                (THE REAL SUPREME DWARF IS AT CAMP,
                I'VE GOTTEN A LOT OF LETTERS)
                     ST1306


Category:DATING
Subject:DWARVES!!!!(HEH HEH)
From:ST1306
Posted:7 JUL 12:33 am


<N>ext,<PO>st, or Return for text-
          HIHO!!!!!!!!!!

THE DWARVES ARE HERE!  WE ARE SHORT, FOUL SMELLING, AND WE WANT
TO MEET YOU!!!
(ESPECIALLY IF YOU ARE A PRETTY GIRL!)
WRITE US AT ST1306

                    SMELLY
                ACTING SUPREME DWARF
************************************************************
```

As you can see, everything is pretty much business on The Source. Most of the users have no sense of humor. By the way, just for fun and in the interest of good journalism I tried calling the "Bi-coastal Woman" (sure, for journalism, ok). As it turns out her name is Lisa XXXX, (I didn't ask her last name), and it seems her sister and a friend put her number on the listing.

She was rather shocked when I called (I have that kind of affect on women), because she had her answering device on for the last few days and was hoping the calls would have stopped (I just keep on trying). She wants you, the reader, to know that the SOURCE does work, and that she was getting calls from across the country (i.e. Chicago, etc. — She wanted to know which coast it was on). All in all that kind of thing can be funny at first, but soon turns into the nuisance you would expect.

After all I've said so far, I've only just scratched the surface of what is available. The educational information is tremendous: how to fund a college education and get work experience at the same time in government sponsored programs; what the menus are like and how to use them efficiently; how to chain menus and get to where you want quickly; how to use the various databases for business; how you can use the system to write your own programs; about data storage. As you can see, the list goes on and on. If you want to see more on this kind of information, let me know (I didn't even talk about Startrek or their new adventure "BLACKDRAGON"). Until next time at Beginner's Corner.

Spencer Koenig
153-27 73 Avenue
Flushing NY 11367 ■

ASSEMBLY LANGUAGE FOR BEGINNERS

continued from page 25

```
PC=5203   SP=8000  A=01    INC  A
PC=5204   SP=8000  A=02    INC  A
PC=5205   SP=8000  A=03    CALL UNDO
PC=520B   SP=7FFE  A=03    DEC  A
PC=520C   SP=7FFE  A=02    DEC  A
PC=520D   SP=7FFE  A=01    DEC  A
PC=520E   SP=7FFE  A=00    RET
PC=5208   SP=8000  A=00    JP   LOOP
PC=5202   SP=8000  A=00    INC  A       -> Loop <-
PC=5203   SP=8000  A=01    INC  A
PC=5204   SP=8000  A=02    INC  A
PC=5205   SP=8000  A=03    CALL UNDO
PC=520B   SP=7FFE  A=03    DEC  A
PC=520C   SP=7FFE  A=02    DEC  A
PC=520D   SP=7FFE  A=01    DEC  A
PC=520E   SP=7FFE  A=00    RET
PC=5208   SP=8000  A=00    JP   LOOP
PC=5202   SP=8000  A=00    INC  A
etc.
```

Note: this little program will go on for ever and ever (until you power down or hit reset). If you were to enter this program (with EDTASM, a very good idea), assemble it, then run it, you would see and hear nothing. Remember, this program doesn't do anything except increment and decrement the A register.

The CALL instruction will save the NEXT address to be executed (all ready present in the PC register) in the stack, and then Jump to the target address (in the argument field). Whenever a subroutine is CALLed, it must end with a RETurn. The rule is: every CALL must have a matching RET. Now, the only question remaining is, "What in the world is a Stack?"

A Stack is a special kind of "list." Values are stored in the stack in sequential order. In a sense, a stack is just like an array with one dimension. There are a couple of very special differences, however. The structure of the stack is such that values are always added to the top of the stack, and taken off of the top of the stack. This means that the last item in the stack is always the first item out (this structure is known as a LIFO structure). Every item on the stack must be a sixteen bit value. And finally, the stack is supported by the CPU hardware. There are several instructions that are specifically intended for the stack. The CALL and RET are two of them. As we explore other programs and programming techniques, we will learn about additional stack instructions. There is one other strange feature about the stack: it grows upside down. If we want a stack that is 100H is size, and is located at the top of memory (in a 16K system), we would place it at location 7F00. Whenever a value is added to or removed from the stack (as in a CALL or RET), the SP (or Stack Pointer register) is DECremented (for a CALL) and INCremented (for a RET). That is why (in the second trace), every time a call was made, the SP was DECremented by 2, and the next address to be executed is stored in the memory location pointed to by the SP. Whenever the RET is executed, the value pointed to by the SP is loaded into the PC, and the SP is INCremented by 2. Since all stack accesses are sixteen bits long (requiring 2 bytes in memory), all Stack INCs and DECs are double INCs and double DECs.

Now what about the two routines that were CALLed? CLR is a special routine that clears the screen and positions the cursor to the HOME position (upper left corner). In one of the upcoming issues, I will show you how to write your own clear routine. Ths DISP routine is a little more complicated. This routine will display whatever character is in the A register, at the current cursor position. Then, the cursor position is INCremented by one. If the cursor is at the end of the line, it jumps down to the next line. If the cursor was at the last position on the last line, the screen is scrolled up 1, and the cursor is set to the first position of the last line.

It's hard to believe, but that just about wraps up this months issue. I suggest that you try entering and running the alphabet display program. Remember, first you enter the program with the I command, save it to cassette or tape, Assemble it with the A command (saving the machine code). Then, run the machine code. By the way, if you don't want the program to re-boot after execution (a desirable feature with BASIC only systems), replace the "JP DOS" or "JP 0000H" with a "JP $". This will cause another endless loop (it will Jump to itself until the computer is reset). Next month, I will begin by re-structuring this program in two different ways.

Joseph Rosenman
35-91 161st Street
Flushing, NY 11358 ■

COLOR COMPUTER CORNER

Joseph Rosenman

This month, I will begin with a review of a book for those Color Computer users who have very little background in computers.

TRS-80 Color Basic

by Bob Albrecht, published by John Wiley & Sons, New York, 1982. List price: $9.95

TRS-80 Color Basic is a well-written and accessible book which effectively introduces the novice computer user to the TRS-80 Color Computer and Color Computer Basic. This book doesn't intend to replace the Radio Shack manuals, but provides and easy-going and "fun" guide to assist in the learning process. The book only covers the "Standard" Color Basic (that is, it doesn't cover "Extended" Color Basic). The book doesn't get profound; it carefully maintains its focus on the needs and questions of beginners.

How does he do it? He starts by writing a 378 page book that includes 13 chapters and 11 appendices. After assuring the readers that there is nothing to fear (and after acquiring a "griddleyesque" artist to illustrate the book), the author jumped right into a description of the Color Computer. Throughout the book, there is a liberal sprinkling of "question and answer" sections included. I suppose one way to make silicon chips more "appealing" is to compare them to multi-legged centipedes. As soon as the equipment has been introduced (and given a chance to take a bow), understanding the USE of the Color Computer is stressed. Indeed, one of the strongest points of this book is its emphasis on understanding Basic through example. (Needless to say, you MUST have a Color Computer available if you want to make effective use of this book.) Each chapter opens with an introductory description and a list of goals.

I mentioned above that this book has 11 appendices. It's true! Now (as we all know), an appendix can be one of three things: a useless appendage that costs several thousand dollars to get rid of, a place in a book where you can stick unnecessary and incomprehensible details, or (as is the case with *TRS-80 Color Basic)* a place to include miscellaneous details, charts, and diagrams that can REALLY help. While it may seem silly to all of "us" experienced users, there really are people who need to see a big diagram of the cassette recorder showing the correct buttons to push. Since this book is targeted for beginners, the author didn't fail to include such details. Note: the appendices also include more complex sections such as a list of reserved words, screen location charts, and error message descriptions.

I would recommend this to any person who is a beginner on the Color Computer and has no experience on other microcomputers. If you have worked through Standard Basic, and are already using Extended Basic, this book won't have very much to offer. If YOU are a beginner, don't even think twice. Ten dollars for a well written book in this field is a real buy. (I have spent as much as fifty dollars for Computer Science texts!)

Ready for some reviews? Good! We start off with another winner.

Color Berserk

from Mark Data Products, 23802 Barquilla, Mission Viejo, CA 92691, 714-768-1551

System requirements: 16K non-extended or extended BASIC.
Supplied on: Cassette. Joysticks required.
Disk compatible: Probably.
Price: $24.95

This machine language game is another example of what Color Computer users should expect out of their machines. Model 1/3 users might know of this game as Robot Attack from Big Five software (I believe there is also an Atari version around). In this game, you suddenly find yourself in a (simple) maze with electric walls. If you touch one of the walls, ZAP!!! To make matters worse, the maze is filled with nasty Laser-wielding robots. Your goal: destroy the robots with YOUR Laser before they destroy you. Then, escape from the maze. In this game, your man actually LOOKS like a man! I was impressed. Anyway, after you escape, you will be offered a new maze and more robots. You get to be destroyed only three times before you lose (an extra man awarded every 5000 points). Ah, but there is a catch. "Evil Orville" lurks nearby. You can't destroy Evil Orville. Evil Orville appears as a big smiling face, and moves quite a bit faster then the robots. One touch from Evil Orville and ... it's curtains. This use of color isn't spectacular, but it is interesting and appropriate. The same holds true for the sound, which is low key but effective.

Evaluation: An excellent and challenging machine language arcade game. I would place this game in the same category as Ghost Gobblers from Spectral Associates, and would recommend them both hardily. The only complaint I have is with the documentation. Instead of including a booklet (or even a sheet), the instructions are typed onto the back of the cassette packaging. Of course, this packaging must be torn in order to use the cassette!

Sir Eggbert Jumper

from David Dawson, DDF Soft, 10317 N St., Omaha, NE 68217

System requirements: 16K Extended BASIC.
Supplied on: Cassette. Joysticks required.
Disk compatible: Probably.
Price: $15.95 (I've seen it advertised for $12.95)

This is an interesting little game that (especially for BASIC) makes impressive use of the color computer. The object of the game is to grab a special key (visible on the screen), an get to the exit door. Of course, to do this, the adventurer must jump from level to level, and avoid an occasional unpleasant monster. Jumping and moving is

accomplished with the joystick.

Evaluation: This program, while being fun, is not extremely substantial. The entire "dungeon" is visible on the screen (and there for small not complicated). The graphics and sound are handled well, and the game does provide some challenge. In addition, the price is very reasonable compared to other Color Computer software. The instructions were complete, but came in several small sheets (photocopied from a "modest" resolution matrix printer. I would have preferred a somewhat bolder print on 8.5 x 11 paper any day. If computer "coordination/skill" games are your thing, you will certainly enjoy Sir Eggbert Jumper.

More about Busses

Several issues back, I had talked about microcomputers (in general), and system "busses." In that article, I mentioned that the TRS-80 Model 1, 3 and Color Computer didn't use buses (all true). I also said that I believed the same was true for the Model 2. Recently, I needed to open up a Model 2. Inside, I discovered that Radio Shack DID use a bus structure. How could I tell? There were several computer cards (special Printed circuit cards that contain specific circuitry) that plugged into slots. As far as I could tell, any card could plug into any slot. That means that the slots conformed to a bus structure. Some of you might have heard about the new TRS-80 model 16 (based on the 16 bit 68000 microprocessor). This new computer is built upon the Model 2. A card containing the 68000 MPU is added to the Model 2 (no doubt, accompanied by other changes) to upgrade from a Model 2 to a Model 16. The reason Radio Shack was able to design and implement this upgrade is due to the versatile bus structure.

Joseph Rosenman
35-91 161st Street
Flushing, NY 11367 ■

PROGRAM CONVERSION (PART IX)

Richard Kaplan

HOW DO I CONVERT MY 123-XY OPERATING SYSTEM TO READ QUAD-DENSITY, DOUBLE-SIDED QWX 334/4 DISKS?

WILL DIGICALC WORK WITH MY 5-SIDED, 17 MEGABYTE HARD DISK DRIVE?

WHY DOES COMPUSORT SORT BACKWARDS WHEN RUN ON MY Z29 DISK DRIVES WITH A MAXISOFT INTERFACE?

As some Computronics readers may be aware, a substantial portion of my time at Computronics is spent providing technical support for the many products we write. I would venture to guess that at least 20% of all of these technical inquiries relate to an operating system incompatibility, such as those listed above (admittedly exaggerated). I do not mean to parody these callers; my point is simply that the abundance of alternative MODEL I/III operating systems has placed many TRS-80 owners in a very confusing situation.

This article is the first of several I will write (not necessarily in consecutive months) dealing with operating system difficulties. This month I will explain how to convert to/from several popular MODEL I/III alternative operating systems. First, however, I would like to share a letter I received recently from Vernon B. Hester, author of MULTIDOS, which expands upon my July article:

Dear Mr. Kaplan:

I really enjoy your series Program Conversion. *I have used it to assist several people in converting programs beween the MODEL I and the MODEL II.*

In the July issue of Computronics, *I noticed several omissions. Here is a list of omissions and other available features found in DOSs not mentioned in your series:*

HEADING	OMISSION	AVAILABLE
SWAP	CMD"F=SWAP",vn1,vn2	NEWDOS/80
ERASE	CMD"F=ERASE",vn1	NEWDOS/80
SYSTEM"I"	CMD"DIR"	LDOS,MULTIDOS
CMD"I"	CMD"XXX"	LDOS,MULTIDOS

We realize that you do not have a copy of MULTIDOS. Therefore, please accept these complimentary copies from us.

Sincerely,
Vernon B. Hester
Cosmopolitan Electronics Corporation

Most of Mr. Hester's letter is self-explanatory. However, what I feel may be most helpful to MODEL I/III users is the availability of an ERASE function in NEWDOS/80. In July, I stated (incorrectly) that it is not possible (easily) to ERASE an array on the MODEL I/III, as is possible with the MODEL II. This NEWDOS/80 function can be a real time saver for programmers who use strings often, particularly users of large string arrays.

WHAT IS AN ALTERNATIVE OPERATING SYSTEM?

An alternative operating system is a substitute for TRSDOS, the operating system which virtually every TRS-80 owner posesses. Alternative operating systems often offer significant advantages to users over TRSDOS. These advantages may include faster disk access, utility programs (such as utilities to allow one program to be printed while another is being entered—a technique known as SPOOLING), additional BASIC functions (such as a machine-language sort), and MODEL I/III read/write compatibility, to name just a few.

Operating systems are usually purchased as a single disk, similar to a TRSDOS master disk. When this disk is placed into the owner's computer, many commands take on a different format. Each operating system generally has its own BACKUP and FORMAT procedures, its own version of BASIC, its own syntax for commands such as DIR, etc. This leads us to what is likely the most common question among all MODEL-I or MODEL-III owners:

WILL PROGRAM X WORK WITH OPERATING SYSTEM Y?

THE ANSWER: Probably yes, provided that the disks are first converted.

Although commands such as BACKUP and FORMAT (operating system commands) are generally "operating system-dependent", machine language programs will almost always execute on a multitude of operating systems without program alteration. Versions of BASIC used by various operating systems are also usually standard.

Most operating systems use the standard MODEL I/III BASIC commands as a base and expand upon this "BASIC" (excuse the pun) set of commands by adding more functions. As long as these special BASIC commands added by each operating system are not used (and frequently they are not in commercial software), BASIC programs will be portable between operating systems. The only requirement is conversion between differing disk formats.

WHY MUST DISK FORMATS BE CONVERTED?

To TRS-80 owners, it must seem as if there are as many disk formats around as there are operating systems. To some extent this is because operating system companies want their users only to use their DOS (Disk Operating System). To a larger extent, however, many operating system developers have been quite innovative in developing more efficient disk formats which store more data on a disk than does TRSDOS.

A disk written in a very efficient format on a MODEL-I ("DOUBLE DENSITY", for example, which stores twice as much data on one disk as does a standard Tandy disk drive

with TRSDOS) cannot be read by TRSDOS. There are exceptions (One notable exception—MULTIDOS—to be discussed later on), but generally a disk created with any one operating system may be read by another operating system only after performing some type of conversion procedure.

THE ACTUAL CONVERSION TECHNIQUES

What follows are procedures which may be used to convert a program between any two of five common operating systems:

NEWDOS/80	Apparat, Inc.
(version 2)	4401 South Tamarac Parkway
MOD I/III	Denver, CO 80237
DOSPLUS	Micro Systems Software Inc.
(version 3.4)	5846 Funston Street
MOD I/III	Hollywood, FL 33023
MULTIDOS	Cosmopolitan Electronics Corporation
MOD I/III	P.O. Box 89
	Plymouth, MI 48170
TRSDOS 2.3	Standard with purchase of any Tandy
MOD I	MODEL-I disk drive
TRSDOS 1.3	Standard with purchase of any Tandy
	MODEL-III disk drive.

Users or manufacturers of other operating systems are encouraged to submit information pertaining to other operating systems for the MODELS I, II, or III. I will be glad to include this information in the future.

DOSPLUS TO NEWDOS/80

MODEL-I single density users of DOSPLUS need not convert a disk at all for use with NEWDOS/80. MODEL-I single density DOSPLUS disks are directly readable by MODEL-I NEWDOS/80. (Place NEWDOS/80 into drive 0 and a DOSPLUS MODEL-I single density disk into drive 1.)

MODEL-I or MODEL-III double-density users of DOSPLUS must convert a disk before it is readable by NEWDOS/80. First, the DOSPLUS diskette must be converted to single density. This is done by FORMATting a disk in single density (by responding "S" when asked for disk density and <ENTER> to all other prompts) and COPYing or TRANSFERring programs onto the single-density disk. Note that it may be necessary to break one DOSPLUS double density disk into two DOSPLUS single density disks.

MODEL-I users of DOSPLUS may now insert their single-density DOSPLUS disk(s) into drive 1 and a NEWDOS/80 operating system into drive 0. It will then be possible to obtain a DIRectory of drive 1 or to LOAD and SAVE programs.

MODEL-III users of DOSPLUS should insert a NEWDOS/80 operating system into drive 0 and their single density DOSPLUS disk into drive 1. Now enter the following:

```
PDRIVE,Ø,9=1
PDRIVE,Ø,1,TI=A,TD=A,TC=35,SPT=1Ø,TSR=3,GPL=2,DDSL=17,DDGA=2
WRDIRP,1
```

The PDRIVE commands will literally "fool" NEWDOS/80 into "thinking" drive 1 is a MODEL-I disk drive. IT WILL NOT BE POSSIBLE TO FORMAT OR COPY DOUBLE DENSITY (MODEL-III) disks in this drive until the completion of this procedure or until you reboot with a different NEWDOS/80 operating system. PDRIVE makes a change on your NEWDOS/80 operating system disk which is restored to normal upon completion of this procedure.

Incidentally, the PDRIVE command on NEWDOS/80 is perhaps the most powerful command on any operating system in existence. This command allows NEWDOS/80 to read/write to disk drives of virtually any combination of size (5 1/4" or 8"), density, sectors, etc. For anyone who intends to do a lot of conversion between operating systems (such as commercial software vendors), both NEWDOS/80 and MULTIDOS (to be discussed later on in this article) can be indispensible tools.

The WRDIRP command will convert the disk in drive 1 to MODEL III-readable single density format. This disk will then be directly readable by MODEL-III NEWDOS/80.

You should now COPY your files from drive 1 to drive 0. When this is done, the following command should be executed in order to restore your drive # 1 to normal:

```
PDRIVE,Ø,1=9
```

NEWDOS/80 TO DOSPLUS

MODEL-I single density NEWDOS/80 disks are directly readable by MODEL-I DOSPLUS. Similarly, MODEL-III single density NEWDOS/80 disks are directly readable by MODEL-III DOSPLUS. Simply insert DOSPLUS into drive 0 and a NEWDOS/80 disk into drive 1.

MODEL-I and MODEL-III double density users of NEWDOS/80 must first convert their disk(s) to MODEL I/III single density before they are readable by DOSPLUS. To do so, it is necessary to FORMAT a new disk in drive 1 by executing the following commands:

```
PDRIVE,Ø,9=1
PDRIVE,Ø,1,TI=A,TD=A,TC=35,SPT=1Ø,TSR=3,GPL=2,DDSL=17,DDGA=2
FORMAT,1
```

As with the DOSPLUS to NEWDOS/80 technique described above, the PDRIVE commands temporarily transform drive 1 into a MODEL-I disk drive.

After FORMATting the disk in drive 1, you should COPY your double density files onto that disk. The disk in drive 1 will then be readable by DOSPLUS.

The following command should then be executed in order to restore your drive # 1 to normal:

```
PDRIVE,Ø,1=9
```

TRSDOS 1.1, 1.2, & 1.3 TO DOSPLUS

MODEL-III users of DOSPLUS, unlike MODEL-I users, will find that it is not possible to read TRSDOS disks with DOSPLUS. Since most purchasers of DOSPLUS wish to convert their old files, the designers of DOSPLUS incorporated a conversion utility to perform this task.

First, it is necessary to define TRSDOS 1.1 and TRSDOS 1.2, since these operating systems are no longer distributed by Radio Shack. TRSDOS 1.1 and TRSDOS 1.2 were the first two versions of MODEL-III TRSDOS. Each had severe errors, and they have now been replaced with TRSDOS 1.3.

Why would anyone still own TRSDOS 1.1 or 1.2? Most people would have no need for these versions of TRSDOS. However, the ROUTE command (which routes output between the RS-232 port, the printer, and the keyboard) does not exist in TRSDOS 1.3. Therefore, anyone who has a need for this command might still be using the older version of TRSDOS. Fortunately, DOSPLUS, which does not have severe errors, has a direct equivalent of ROUTE - RS232.

(As an aside, I might point out that the deletion of ROUTE from TRSDOS 1.3 is an undocumented change on Tandy's part. Rumor has it that certain combinations of this command would result in ludicrous results, such as the printer's wildly feeding paper. As a result, the command was deleted. It still *does* appear, however, when the LIB or HELP commands are executed; this may cause some confusion among MODEL-III users.)

After digressing so much, I will now explain how to convert a TRSDOS 1.1 or 1.2 disk to DOSPLUS 3.4. Simply insert a DOSPLUS operating system into drive 0 and a TRSDOS disk into drive 1. Then enter the following:

CONVERT :1 :0 <ENTER>

(Note the spaces before the colons.)

This command will move all of your TRSDOS files in drive 1 onto your DOSPLUS disk in drive 0. The equivalent command for TRSDOS 1.3 would be the following:

CONVERT :1 :0 (V13) <ENTER>

DOSPLUS TO TRSDOS 1.1, 1.2 OR 1.3

In order to transfer a MODEL-III DOSPLUS disk to MODEL-III TRSDOS, it is necessary to first create a MODEL-I format disk. This MODEL-I format disk may then be run through Tandy's CONVERT routine, which is included and documented in every MODEL-III release of TRSDOS.

To create a MODEL-I format disk, insert DOSPLUS into drive 0 and a blank disk into drive 1. Then FORMAT the disk in drive 1, answering "S" to the prompt for disk density and "35" to the prompt for "Number of Cylinders". Now you should COPY or TRANSFER your MODEL-III DOSPLUS double density files to your newly-made MODEL-III DOSPLUS single density disk. It may be necessary to use two disks for this.

The disk in drive 1 (onto which your files have been TRANSFERred or copied may now be run through the CONVERT routine of TRSDOS. To do this, your DOSPLUS single density disk should be inserted into drive 1 and TRSDOS should be inserted into drive 0. Then type the following:

CONVERT :1 :0 <ENTER>

(Note the spaces before the colons.)

Your single density DOSPLUS files will now be transferred to drive 0.

TRSDOS 1.1, 1.2 OR 1.3 TO NEWDOS/80

In order to convert a MODEL-III TRSDOS disk to NEWDOS/80, it is necessary to first execute the following while your NEWDOS/80 operating system is in drive 0:

```
PDRIVE,Ø,9=1
PDRIVE,Ø,1,TI=AM,TD=E,TC=4Ø,SPT=18,TSR=3,GPL=6,DDSL=17,DDGA=2
```

These commands "fool" your computer into reading disks in drive 1 in TRSDOS format. This will be reversed when the conversion is complete.

Next, you should COPY all your files from drive 1 to drive 0. When this is done, the following command should be executed:

```
PDRIVE,0,1=9
```

NEWDOS/80 TO TRSDOS 1.1, 1.2 OR 1.3

In order to convert a disk from NEWDOS/80 to MODEL-III TRSDOS, it is first necessary to execute PDRIVE commands, as follows:

```
PDRIVE,Ø,9=1
PDRIVE,Ø,1,TI=AM,TD=E,TC=4Ø,SPT=18,TSR=3,GPL=6,DDSL=17,DDGA=2
```

At this point, you should COPY your files on drive 0 to a MODEL-III TRSDOS disk in drive 1. Finally, the following command should be executed to restore your drive # 1 to normal:

```
PDRIVE,0,1=9
```

CONVERSION BETWEEN TRSDOS, DOSPLUS OR NEWDOS/80 AND MULTIDOS

Probably the most versatile of all operating systems regarding differing disk formats is MULTIDOS. MULTIDOS can read/write to virtually all popular operating sytems *without* any conversion whatsoever. In other words, MULTIDOS CAN AUTOMATICALLY RECOGNIZE VARYING DISK FORMATS WITH NO OPERATOR INTERVENTION REQUIRED.

With MODEL-III MULTIDOS, disks formatted by NEWDOS/80, DOSPLUS, VTOS, LDOS, or DBLDOS may be inserted into drive 1. After inserting MULTIDOS into drive 0, any of the disks mentioned in the preceding paragraph may be read from/written to without any conversion procedure.

(NOTE: There are some restrictions upon compatibility with MODEL-III NEWDOS/80 and the MODEL-III and upon several operating systems and the MODEL I; these restrictions are explained in the manual—there is no need to reiterate that material here.)

The only MODEL-III operating system mentioned which is not directly readable by MULTIDOS is TRSDOS. A conversion utility is provided to read TRSDOS disks. It is not possible to write or convert directly to TRSDOS disks, which is one drawback to MULTIDOS. (Of course, it is possible to save a progam on DOSPLUS or NEWDOS/80 and then transfer to TRSDOS.)

MULTIDOS is an excellent transfer medium for operating system conversions. If there is a combination of operating systems between which is is not possible to convert files, it will most likely be possible to first convert to MULTIDOS and then to the desired operating system.

"LAST RESORT" TECHNIQUES

In the event that a user of a MODEL I/III encounters a group of operating systems between which it is absolutely imposible to convert disks, there are two little-known techniques which may be used between many operating systems: BASIC * and cassette transfer.

Many operating systems enable users to enter the operating system and return to the previous BASIC program through the command "BASIC *". This command may be

used in a slightly different manner.

If you are using a BASIC program in DOSPLUS and wish to transfer the program onto TRSDOS, the first thing to do is boot up with DOSPLUS and load the program. Then, with the program in memory, reboot with TRSDOS, and when "TRSDOS READY" returns, enter the following:

BASIC * <ENTER>
(Type "*" but not the <>'s.)

Now, SAVE your program on your TRSDOS disk.

The preceding technique definitely does work between DOSPLUS and TRSDOS. It is possible that it also works between other operating systems.

When dealing with BASIC programs, the absolute last resort, which will work with virtually any combination of operating systems, is to save a program on cassette with one operating system and load it in with the other system. This is a fairly long technique, but it sure beats keying in a program!!

This concludes this month's tips on program conversion. As always, I would appreciate any reader feedback, particularly regarding operating systems. If you have a shorter method to convert between operating systems, or if you know of an operating system I have not covered, please send me the information. Of course, any inquiries about MODEL I, MODEL II, MODEL III, APPLE, CP/M incompatibilities are always welcome.

NOTE TO MODEL-II USERS

Although no material pertaining to the MODEL-II was presented this month, I do intend to continue writing about the MOD-II. Particularly, I am interested in the MODEL-II version of DOSPLUS when it becomes available. Any information readers can provide would be most welcome.

In addition, if any MODEL-II user has any information regarding MOD II/16 compatibility, it would be greatly appreciated. At present, it seems that these computers really are 100% compatible, but when the MODEL-III was first released Tandy said it was 100% compatible with MODEL-I programs. Need I say more?

Richard Kaplan
H & E Computronics
50 N. Pascack Road
Spring Valley, NY 10977 ■

CASH FLOW ANALYSIS

continued from page 31

```
213Ø IF OP=1 THEN PRINT USING C$;Ø,Ø,Ø,Ø,Z(Ø,4),Ø,Z(Ø,6)
214Ø IF OP=2 THEN LPRINT USING C$;Ø,Ø,Ø,Ø,Z(Ø,4),Ø,Z(Ø,6),Ø
2145 IF OP=4 THEN LPRINT USING C$;Ø,Ø,Ø,Ø,Z(Ø,4)
215Ø ST=Ø
216Ø REM INFLAT
217Ø SR=RR : ST=Ø : FOR I=1 TO YS-1: FOR J=1 TO 4:
IN(J)=(1+IM(J))↑I : NEXT J :
Z(I,9)=IN(1)*Z(I,1)-IN(3)*Z(I,3)-Z( I,5)-Z(I,7) : IF I=1 THEN
Z(1,9)=Z(1,9)+Z(Ø,9)
219Ø IF SS=1 THEN Z(I,11)=CG*Z(I,1Ø)*(IN(2)*Z(I,2) -
IN(4)*Z(I,4)) ELSE Z(I,11)=Ø
22ØØ D6=Z(I,1Ø)*Z(I,9) : K%=I/13 : K%=K%*13 : KK%=I:IF SS <>1
THEN Z(I,9)=Z(I,9)+IN(2)*Z(I,2)
22Ø5 IF S=1 AND  Z(I,11)<Ø THEN PRINT "CAPITAL LOSS - TAXABLE
INCOME ADJUSTED":Z(I,9)=Z(I,9)+Z(I,11)/Z(I,1Ø)
221Ø IF D6>0 THEN Z(I,11)=Z(I,11)+D6
222Ø Z(I,12)=Z(I,9)-Z(I,11)-Z(I,4)*IN(4)+Z(I,5)-Z(I,8)+
Z(I,6): IF SS=1 THEN Z(I,12)=Z(I,12)+Z(I,2)*IN(2)
223Ø X=(1+R)↑I: SR=SR+Z(I,12)/X: IF Z(I,9)<Ø THEN ST=ST+
Z(I,9)*Z(I,1Ø)/X
224Ø IF OP=1 THEN PRINT USING C$;
I,Z(I,1)*IN(1),Z(I,2)*IN(2),Z(I,3)*IN(3),Z(I,4)*IN(4),Z(I,5),
Z(I,6): IF K%=KK% THEN INPUT "ENTER TO PAGE";R$
225Ø IF OP=2 THEN LPRINT USING C$;I, Z(I,1)*IN(1),
Z(I,2)*IN(2), Z(I,3)*IN(3), Z(I,4)*IN(4), Z(I,5), Z(I,6),
Z(I,7)
226Ø Z(I,13)=Z(I,12): IF Z(I,9)<Ø THEN Z(I,13)=
Z(I,13)-Z(I,1Ø)*Z(I,9)
227Ø IF OP=3 THEN LPRINT USING C$;I, Z(I,1)*IN(1),
Z(I,2)*IN(2), Z(I,3)*IN(3), Z(I,4)*IN(4), Z(I,5), Z(I,6),
Z(I,7), Z(I,8), Z(I,9), Z(I,1Ø), Z(I,11), Z(I,12), Z(I,13 )
2275 IF OP=4 THEN LPRINT USING C$;I, Z(I,1)*IN(1),
Z(I,2)*IN(2), Z(I,3)*IN(3), Z(I,4)*IN(4)
228Ø NEXT I
229Ø FOR I=1 TO 3:E(I)=Ø: NEXT I
23ØØ FOR I=YS TO G:E(1)=E(1)+Z(I,2): E(2)=E(2)+Z(I,5):
E(3)=E(3)+Z(I,8): NEXT I
231Ø FOR J=1 TO 4:IN(J)=(1+IM(J))↑(I-1): NEXT J : I=YS :
Z(I,9)=IN(1)*Z(I,1)-IN(3)* Z(I,3)-D5-Z(I,7)
233Ø IF SS=1 THEN Z(I,11)=CG*Z(I,1Ø)*(IN(2)*E(1)-
E(2)+D5-IN(4)*Z(I,4)) ELSE Z(I,11)=Ø
234Ø D6=Z(I,1Ø)*Z(I,9):IF SS <>1 THEN Z(I,9)=Z(I,9)+
IN(2)*Z(I,2)-Z(I,5): Z(I,11)=Z(I,9)*Z(I,1Ø): IF Z(I,11)<Ø
THEN Z(I,11)=Ø
2345 IF SS=1 AND Z(I,11)<Ø THEN PRINT "CAPITAL LOSS - TAXABLE
INCOME ADJUSTED":Z(I,9)=Z(I,9)+Z(I,11)/Z(I,1Ø)
235Ø IF D6>0 THEN Z(I,11)=Z(I,11)+D6
236Ø Z(I,12)=Z(I,9)+E(2)-E(3)-Z(I,11)+Z(I,6):IF SS=1 THEN
Z(I,12)=Z(I,9)+D5-E(3)-Z(I,11)+E(1)*IN(2)+Z(I,6)
237Ø X=(1+R)↑I:SR=SR+Z(I,12)/X:IF Z(I,9)<Ø THEN
ST=ST+Z(I,9)*Z(I,1Ø)/X
238Ø IF OP=1 THEN PRINT USING C$;I,Z(I,1)*IN(1),
E(1)*IN(2),Z(I,3)*IN(3),Z(I,4)*IN(4),E(2),Z (I,6)
239Ø IF OP=2 THEN LPRINT USING C$;I,Z(I,1)*IN(1),
E(1)*IN(2),Z(I,3)*IN(3),Z(I,4)*IN(4),E(2),Z (I,6),Z(I,7)
24ØØ Z(I,13)=Z(I,12):IF Z(I,9)<Ø THEN
Z(I,13)=Z(I,13)-Z(I,9)*Z(I,1Ø)
241Ø IF OP=3 THEN LPRINT USING C$;I,Z(I,1)*IN(1),
E(1)*IN(2), Z(I,3)*IN(3), Z(I,4)*IN(4), E(2),Z(I,6), Z(I,7),
E(3), Z(I,9), Z(I,1Ø), Z(I,11), Z(I,12),Z(I,13)
2415 IF OP=4 THEN LPRINT USING C$;I, Z(I,1)*IN(1),
E(1)*IN(2), Z(I,3)*IN(3), Z(I,4)*IN(4)
242Ø IF OP=1 THEN INPUT"ENTER TO PAGE";DU$
243Ø IF OP=2 THEN LPRINT" "
244Ø IF OP=1 THEN PRINT A$(1);: FOR I=8 TO 12: PRINT A$(I);:
NEXT I : PRINT A$(13);: PRINT A$(14);: FOR I=21 TO 25: PRINT
A$(I);: NEXT I: PRINT A$(26)
245Ø IF OP=2 THEN LPRINT A$(1);: FOR I=9 TO 13: LPRINT
A$(I);: NEXT I: LPRINT"  ADJUSTED ": LPRINT A$(14);: FOR I=22
TO 26: LPRINT A$(I);: NEXT I: LPRINT "    $ FLOW"
2456 IF OP=4 THEN LPRINT " ": LPRINT A$(1);: FOR I=6 TO 9 :
```

```
LPRINT A$(I);: NEXT I: LPRINT : LPRINT A$(14);: FOR I=19 TO
22: LPRINT A$(I);: NEXT I: LPRINT
246Ø IF OP=1 THEN PRINT USING CD$;Ø,Ø,Ø,Ø,Ø,Ø,RR
247Ø IF OP=2 THEN LPRINT USING CC$;Ø,Ø,Ø,Ø,Ø,RR,RR
2475 IF OP=4 THEN LPRINT USING C$;Ø,Ø,Z(Ø,6),Ø,Ø
248Ø K=Ø:FOR I=1 TO YS:K=K+1:EE=Z(I,8):IF I=YS THEN EE=E(3)
249Ø IF OP=1 THEN PRINT USING CD$; I, Z(I,7), EE, Z(I,9),
Z(I,1Ø), Z(I,11), Z(I,12)
25ØØ IF OP=2 THEN LPRINT USING CC$; I, EE, Z(I,9), Z(I,1Ø),
Z(I,11), Z(I,12), Z(I,13)
2505 IF OP=4 THEN LPRINT USING C$; I, Z(I,5), Z(I,6), Z(I,7),
Z(I,8)
2510 IF OP=1 THEN IF K=13 THEN K=Ø:INPUT"ENTER TO PAGE";DU$
2520 NEXTI
2521 IF OP<>4 THEN 253Ø
2522 LPRINT " ": LPRINT A$(1);: FOR I=1Ø TO 13:LPRINT A$(I);:
NEXT I: LPRINT: LPRINT A$(14);: FOR I=23 TO 26: LPRINT
A$(I);: NEXT I: LPRINT
2523 LPRINT USING CX$;Ø,Ø,Ø,Ø,RR
2528 FOR I=1 TO YS:K=K+1:EE=Z(I,8):IF I=YS THEN EE=E(3)
2529  LPRINT USING CX$;I,Z(I,9),Z(I,1Ø),Z(I,11),Z(I,12):NEXT
253Ø H$="DISCOUNT CASH" : H1$="DISCOUNT TAX SAVINGS" :
H2$="DISCOUNT CASH AND TAX SAVINGS " : T=SR-ST: IF OP=1 THEN
PRINT H$;SR,H1$;ST,H2$;T;
254Ø IF OP>1 THEN LPRINT H$;SR:LPRINT H1$;ST:LPRINT H2$,T
255Ø IF OP=1 THEN INPUT"HIT RETURN";R$
256Ø SR=Ø : ST=Ø : IF OP>1 THEN FOR I=1 TO 4: PRINT"FOR PRINT
OUT" : INPUT A$ : LPRINT A$ : NEXT I
257Ø PRINT"RECYCLE MENU"
258Ø PRINT"    1   REGULAR INCOME"
259Ø PRINT"    2   CAPITAL INCOME"
26ØØ PRINT"    3   YEARLY EXPENSES"
261Ø PRINT"    4   DEPRECIATION"
262Ø PRINT"    5   MORTGAGES"
263Ø INPUT"    6   INFLATION ETC.";IX%: IF IX%<1 OR IX%>6
THEN 257Ø
264Ø ON IX% GOTO 13Ø  ,35Ø  ,4ØØ  ,56Ø  ,11ØØ  ,15ØØ
265Ø REM TAX SUB
266Ø INC=OI+Z(I,9)
267Ø IF INC<Ø THEN Z(I,1Ø)=Ø:RETURN
268Ø IF RB$="R" THEN 272Ø
269Ø FOR K=1 TO ZZ-1:Z(I,1Ø)=TAX(K-1,3)
27ØØ IF INC>TAX(K-1,1) THEN IF INC<TAX(K,1) THEN RETURN
271Ø NEXT K: RETURN
272Ø SX=Ø: SUM=Ø : FOR K=1 TO ZZ-1 : Z(I,1Ø)=TAX(K-1,3) :
SUM=SUM+TAX(K,2) : SX=SX+TAX(K,1)
273Ø IF INC>TAX(K-1,1) THEN IF INC<TAX(K,1) THEN 275Ø
274Ø NEXTK : K=ZZ
275Ø DT=INC-SX:IF DT<Ø THEN 279Ø
276Ø SUM=SUM+DT*Z(I,1Ø)
277Ø Z(I,1Ø)=SUM/INC:IF Z(I,1Ø)<Ø THEN Z(I,1Ø)=Ø
278Ø RETURN
279Ø Z(I,1Ø)=SUM/SX:IF Z(I,1Ø)<Ø THEN Z(I,1Ø)=Ø
28ØØ RETURN
```

SUMMARY

This program calculates the cash flows resulting from a real estate or other type investment and calculates a measure of the present worth of the cash flows based on the classic internal rate of return concept. The program can be used to model many different investment situations. The user can control all aspects of the investment.

The model is a good approximation of real life. It is not perfect. Care should be used when you are not sure of any particular application. When in doubt, check yourself manually until you understand the abilities and limitations of this model.

Steven M. Zimmerman,Ph.D.
College of Business and Management Studies
University of South Alabama
Mobile, Alabama 36688

Leo M. Conrad
Imagineering Concepts
P.O. Box 9843
Mobile, Alabama 36691-0843 ■

ARRAY OF HOPE FOR BASIC PROGRAMMERS
continued from page 19

is reached by a positive pointer, and right branches if it is reached with a negative pointer. Whenever a right pointer is to be used the element is summed. Starting at element 1 the elements will be traversed in the sequence (1), 3, 5, 1, (2), 7, 2, 4, and 6 where the elements in parentheses are only used to find the next element in sequence and will not be summed at that point. Later we shall insert elements in the tree and perform a non sequential search, but here we shall look at the code required to traverse the tree in key sequence.

The tree is assumed to be stored in an array defined with dimension A(Max,2), the key value is in A(I,KEY), the left pointer in A(I,LEFT), the right pointer in A(I,RIGHT), and ROOT points to the root node of the tree. If ROOT is zero, then there are no elements present in the array. The code to traverse the tree could be as follows:

```
  1Ø KEY = Ø: LEFT = 1: RIGHT = 2
  ...
  ...
  1ØØ IF ROOT = Ø THEN PRINT "Tree empty": GOTO 17Ø
  11Ø NXT = ROOT
  12Ø IF A(NXT,LEFT) > Ø THEN NXT = A(NXT,LEFT): GOTO 12Ø
     :REM Continue until left pointer zero
  13Ø PRINT A(NXT,KEY) :REM To be printed or summed
  14Ø NXT = A(NXT,RIGHT) :REM Follow right branch
  15Ø IF NXT < Ø THEN NXT = -NXT: GOTO 13Ø
      :REM Element reached with negative pointer
  16Ø IF NXT > Ø THEN 12Ø
      :REM Element reached with positive, try left path
 17Ø PRINT "Traversal end"
```

Perhaps it would be an idea to try following through the code for the small tree shown in the diagram. It should be relatively easy to follow, and the nodes should be visited in the sequence mentioned in the text.

This seems to be an appropriate point to finish with sequential searching techniques, and to start looking at some non-sequential search techniques.

(To be continued)

Arne Rohde
Pilevej 31
7600 Struer, Denmark ■

COTTAGE COMPUTERY COMES OF AGE

Michael Herbert Shadick

The very first UNIVAC computer was, of course, a vacuum tube affair (remember them?) which occupied two full floors and a large portion of another in one of Manhattan's most stellar skyscrapers. Today, a single sophisticated microchip—small enough to fit on the tip of your finger—contains more RAM power, any way you utilize it, than *three* of those archetypal UNIVACs.

So how far has the computer age *really* come? For the past several years, the stuff of which personal computery is composed (hardware, software, and the operator's own "brainware") has rapidly been approaching the stage where large numbers of homeloving workers (perhaps including you and me?) are enjoying full-fledged membership and participation in the business world—*from the comfort of our dens,* or anywhere else in the world, for that matter! Truly, we are no longer on a threshold; we are in the very midst of personally witnessing the coming of an unprecedentedly dynamic era—the era of *cottage computing.*

Webster has defined "cottage industry" as *any means of livelihood, the main income thereof which is chiefly derived from work conducted in or around the h*ome. America before the industrial revolution of the mid and late 1800s was a coast-to-coast beehive of cottage industrial activity. Farmers, blacksmiths, shoemakers, potters, and other tradesmen of every then-known variety earned their living in their homes. That is where they made their wares, and that is where the people who wanted them came to purchase them. In other words, the cottage worker was, in the truest possible sense of the term, an I/O conduit for goods and services. He or she was a *processor,* if you will!

With the fiery advent of the steam locomotive (and later the automobile), millions of our young nation's human processors began to see some very attractive alternatives to cottage industry, and new industrial centers, cleverly disguised as cities, began popping up all over the place. Lo and behold, the smithys began foregoing their forges, in favor of earning an hourly wage and, perhaps, getting one of those newfangled contraptions called a telephone.

Little did Mr. Bell and his inventive peers realize that they were laying the groundwork (quite literally) for what would one day make *their* industrial revolution look, in comparision, like a tempest in a teapot. For it is the telephone, or more precisely the communications network it represents, that is making possible the new industrial revolution of which we ourselves are a pioneering and, I dare say, pivotal part.

The computer. The telephone. And we mustn't forget that vital link between them, our mighty little friend, the Modem. It is to these three indispensible components of cottage computery—and to the people whose very lives are increasingly being liberated by them—that I would like to address myself in this article.

Shelly Patterson is a typist. Not a particularly fast one, but very accurate. She is also eight months pregnant. "She was making very good money as a legal secretary," her husband Hank told me, "but we both felt that, after the blessed event, something would have to suffer—either the job or the child. We opted for our family-to-be.

At first this posed a rather serious problem for the Pattersons—how to keep their household income up at a time when they need it more than ever, and still give their child the nurturing which they feel is its birthright.

"There was only one way to go," as Shelly explains it. "I had to find some viable at-home employment." The Pattersons considered a number of job possibilities, ranging from telephone sales work to stuffing envelopes for a direct-mail firm. But none of those alternatives took full advantage of Shelly's typing skills and experience.

"Y'know," Hank recalls with a grin, "sometimes you learn about things in the most unexpected ways. I was sitting in the barber shop a couple of months ago, and this guy waiting next to me says, right out of the blue, 'My God!'" I looked over at him, and saw he had been reading an article in a very shopworn magazine, about computers. 'Look at this,' he says to me, pointing to something in the magazine. 'Now you can write somebody a letter electronically—and they'll receive it almost before you send it! Isn't that amazing?' Well sir, I didn't quite find it cause for cussing, but it was interesting. And it got me to thinking—Shelly types many letters every day in the law office where she works. If she had some way to receive and send information electronically, she could do much the same work at home—maybe even while nursing the baby!"

Well, to make a 64-Megabyte story short, Hank mentioned the idea to his wife, who in turn talked it over with the law firm partners—and they informed her that they *already* had half the hardware and most of the software necessary! It so happened that one of the partners had recently purchased a personal computer along with a modem, so it was relatively easy and equally economical for the firm to adapt his system for office use. The partners decided to lease a small but powerful word-processing system for Shelly's use in her home, for as long as she wanted to work for them from there!

Once set up, it took little more than a week or two for Shelly and her employers to iron out the few bugs in the system. As things stand right now, Shelly is doing almost the identical work that she did at the office—with slightly *improved* efficiency of output—all from the nursery!

"It's very nice for me and her," Hank remarked to me recently, "and it'll be even nicer for the *four* of us!" (Shelly, her doctor confirms, is now expecting *twins.*)

In the 1980s (already being hailed as "the decade of software") we are already beholding what can only be termed—and even then with a certain understatement of reality—a programming *explosion.* From three-dimensional computer generated graphics to digital typesetting, from light pen CRT manipulation magic to any number of even more exotic computer-linked applications, the software revolution—completely without precedent though it may be—seems to have *only begun!* In investigating the myriad forms of cottage computery available today, and becoming available tomorrow, I have—because of my background in music—decided to devote part of this article to the subject

of microprocessing sound (read "music") generators and their software associations.

Those of you who are familiar with the profound influence which electronics in general (and the sound synthesizer in particular) have had on the music world, even before the development of the personal computer, have probably heard the rather spacy-sounding brand names of "Arp" and "Moog," the two most popular synthesizers on the market. They've been around, though not in their present forms, for decades. What *hasn't* been around very long, however, is the capacity to *program* them digitally. This means that for the first time in musical history, a composer can hear the finished product of his noteable labors in all (or at least most) of its fully-orchestrated glory—*before a single musical note has been written.* Beethoven, Tchaikovsky, *et al.,* would have had veritable fits of ecstacy, had they enjoyed the capabilities of today's "smart" compositional microprocessors. In addition—and this by no means incidentally—those long-deceased symphonic hitmakers and their contemporaries would have been able to do more *composing*—much more, most likely. For this is what today's musically-programmed home computers free the composer to do!

"The great classical composers did all their own orchestration," explained a songwriter friend of mine, "but since that romantic era, there has been—or I should say *was* a movement towards the composer enlisting someone else, an arranger, to orchestrate his or her own works. The sheer labor involved, you see, in writing different musical parts for every single instrument and voice you want to use, note by note, boggles the minds of most modern composers! But recently, there has been a most historically-unexpected *countermovement*, if you will: an as-yet mild but nonetheless noticeable tilt *back* to the composer doing his or her own orchestral thing from start to finish. This is happening, of course, in large measure because of the musical talents, so to speak, of today's personal computers with appropriate programs. If you yourself haven't as yet seen or *heard* what certain software applications can do, I dare say you won't be able to believe your eyes and ears, the first time you experience it. *I* sure couldn't!"

One such software system, offered by the California-based Symtec, Inc., is based on a sound-synthesizer card for the Apple II (among others). The card's technical functions aside, what it *does* is liberate the composer from the often tediously time-consuming chore of "hand-notating" his arrangements. Now the entire scores of his compositions can be printed out, measure by measure and even instrument by instrument, through the use of a graphics printer integrated into the Symtec system. And that operation—which is one of the *last* steps in the music orchestration process—is only the *beginning* of what an appropriately-programmed Symtec card can do:

- Its music-*composing* (you read right) software provides for entry and editing of the entire score, using *keyboard* commands.
- *Sound* orchestrations can be produced (in stereo or quad matrices) using multiple cards.
- Up to *twenty-one* voices (instruments) can be accessed by the user with a complete compliment of SSG (Super Sound Generator) cards.
- Perhaps most useful (and amazing) of all, at least from an itinerant composer's standpoint (most of whom are neither conspicuously well-off nor well-known) is that the basic price of your SSG software is *under* $200, and is compatible with a wide variety of personal computers. Which means that a home-based composer—or a creative musical hobbyist, for that matter—needn't go elsewhere, and needn't shell out the necessarily steep sums being charged for today's recording studio time, in order to orchestrate his or her own works and then *hear how they sound.*

"It's very much," my musical friend claims, "like having a fully-instrumented symphony orchestra on floppy disks, *under your own baton.*"

Brahms, eat your heart out!

Those of you readers who recall the old-time hot lead print shops of a now (nearly) bygone day, probably know what a Merganthaler Linotype machine looks like (they can still be found in use today, I understand). The Linotype machine resembles a gigantic Rube Goldberg contraption at its craziest, with a seemingly endless array of various and sundry levers, wheels, pulleys and racks, all moving at what appears to be cross-purposes to each other, but in fact operating in a magnificently well-coordinated manner, for the purpose of casting molten metal (SSSSST!) into finished, press-ready galleys of solid type. This is exactly what emerged from the Linotypes—letter by letter, word by word, publication by publication, for almost a century—at the skilled hands of thousands of sweaty Linotype operators.

Now imagine, if you can, having one of Mergenthaler's ungainly contraptions—complete with its attendant kettle of hot lead—in your *home.* Ludicrous, is it not? Yet its modern-day equivalent is exactly what you'll find in the homes, apartments and condominiums of a rapidly growing number of *cottage compositors.* Compositor, by the way, is a fancy word for typographer—and these profitably homely folks are, quite literally, setting type by telephone.

The homebound compositor can use a modem to interface a home computer (with word processing software) through phone lines wyth a companion computer or typesetter at the printing house which employs them. The home compositor can place special codes into the text that specify the type size, font (typestyle) and spacing that is to be used for setting the finished text. The typesetter's own software will automatically recognize, interpret, and act on these codes, producing finished, camera-ready galleys of type in virtually any font and size desired.

"Frankly," confesses Dave Simpson, manager of Cold Type Setters of Minneapolis, "we didn't think it would work. It *couldn't.* What do typists know about *typesetting?* Zero, most likely! But that, we discovered, is part of the magic of our Superbrain system (trademark of Intertec, Inc.). It really is very close to magic. For the Superbrain takes a home operator's keyboard output, along with a few basic commands which take the operator only a few minutes to learn, and outputs the highest-quality galleys you've ever seen. Our customers have never been happier! And, on our part, we don't have to go through the lengthy and expensive process of training new typesetters. Give us

continued on page 56

BUILD WITH THE ELECTRIC PENCIL

Jerome I. Weintraub

The BUILD utility program is very useful for creating an automatic command input file. The procedure for creating this file is clearly described in the Model III Disk Manual (page 31). Programmers writing short BUILD programs should probably use the procedure as it is described in the manual. Longer ones, however, presented me with an editing problem, as there does not seem to be any way to edit them. This means that if you make an error and do not detect it until after you have pressed ENTER for a given entry, you have to start the BUILD file all over again.

I examined the file using SUPERZAP and realized that the BUILD file is a text file saved in the ASCII format, exactly the same as an ELECTRIC PENCIL file with the exception of the filespec extension (/PCL) which is automatically added to every filespec by the ELECTRIC PENCIL program. This led me to suspect that I could write a BUILD file with PENCIL and execute it by the command DO *filespec*/PCL. A simple experiment proved that this indeed works perfectly.

A good example of a long BUILD file came to my attention when I was preparing to enter the patches from the September and October (1981) issues of the *TRS-80 Microcomputer News*. I have over thirty TRSDOS 1.3 disks, and the prospect of entering these patches individually on all my disks indicated that it would take a great deal of time. My friends at the Radio Shack Computer Center suggested that I use the XFERSYS program for this, but I found that XFERSYS can be used only to change from one version of TRSDOS to another. If you already have TRSDOS 1.3, XFERSYS will abort.

Earl R. Savage, in his *Education 80* column (February 1982, page 355) described a procedure for setting up a BUILD file to enter a large number of patches. I have already hinted at the difference between his method and mine, in that a BUILD file cannot be edited after it has been saved. You can, if you wish, append additional patches to a BUILD file with APPEND. To set up a collection of patches in a BUILD file with the ELECTRIC PENCIL, follow these specific steps:

1. Starting with the very first line of your PENCIL file, enter each patch on a single line, exactly as presented by Radio Shack. An entry in a BUILD file may not be longer than 63 characters—exactly the length of one PENCIL line on the screen. It is possible for a patch to be longer than this, depending on how many bytes are being changed. Patches longer than 63 characters could be entered by simply using the original PATCH command, or, if you are careful, you can divide a long patch into two shorter ones.

You may notice that Radio Shack has already done this at least once: two of the three patches to #7 (*Microcomputer News,* Oct. 1981, page 12):

```
PATCH *7 (ADD=5135,FIND=207468652064,CHG=3F20033A7D4E)
PATCH *7 (ADD=513B,FIND=6973B657474,CHG=FE81CA0D55C9)
```

(Note: There must be no spaces inside the parentheses.)

There are six bytes in the first patch, starting at address 5135, and ending at 513A, and the second patch picks up at 513B. Therefore it seems that Radio Shack intends to keep individual patches relatively short. However, you can shorten them by carefully calculating the address of each type by counting from the first address:

5135 to be changed from 20 to 3F
5136 to be changed from 74 to 20
5137 to be changed from 68 to 03 (etc.)

2. Save your PENCIL file with any filespec you choose, but remember that PENCIL will add the /PCL extension to your filespec.

3. If you have two disk drives, place the disk with the patches into Drive 1, and the disk to be updated into Drive 0. With one drive, you will have to get the PENCIL-PATCH program onto each disk to be patched.

4. Get to "TRSDOS READY" and type "DO *filespec*/PCL". All patches listed in your program will be executed. In the event that one or more of the patches in your file have already been made, no harm is done. You will get a "STRING NOT FOUND" message (because the FIND bytes are no longer there) and the DO program will continue without stopping until all patches have been entered.

5. Keep a file of all patches. Since Radio Shack hasn't yet numbered or identified their patches, it will become difficult to know whether any given patch has been made as the number of patches increases. As a matter of fact, two of the patches in the October *Microcomputer News* were repeated from the September issue. Patches,may arise from other sources as well.

Using the PENCIL file, place a number *after* each patch. This entry will not affect the patches during their entry onto a new disk. In this way you will have a permanent record of all patches you have collected.

6. If your collection begins to get larger, you may not want to DO all of the patches that have already been made to your older disks. You can delete those patches from your master file and save the remainder under a new filespec and enter only the new ones. But keep the master file intact for future reference.

This is what my PENCIL PATCH file looks like:

```
PATCH *0 (ADD=5044,FIND=32,CHG=31)                                  1
PATCH *0 (ADD=503A,FIND=20536174204D,CHG=57656420204A)              2
PATCH *0 (ADD=5040,FIND=6179,CHG=756C)                              3
PATCH *6 (ADD=5850,FIND=3A62,CHG=BF5F)                              4
PATCH *6 (ADD=5FBE,FIND=20697320616374,CHG=0D116544C31C44)          5
PATCH *7 (ADD=579C,FIND=0955,CHG=3851)                              6
PATCH *7 (ADD=5135,FIND=207468652064,CHG=3F20033A7D4E)              7
PATCH *7 (ADD=513B,FIND=69736B657474,CHG=FE81CA0D55C9)              8
PATCH *10 (ADD=4E2E,FIND=CD3E4B,CHG=CD8A50)                         9
PATCH *10 (ADD=508A,FIND=4469736B,CHG=4FC33E4B)                    10
```

7. Finally, make a note on each disk of the number of the last patch you have entered. If you are as busy as I am,

continued on page 63

Attention
BARGAIN HUNTERS
Receive Hundreds of Classified Ads Like These Every Month

HARD DISK DRIVE Diablo Mod 31 1.2 MByte std. density. Includes power supp. and cable, rack mount slides, amd manual. Excellent condition. $450. Ca... 1601 B...

IMPACT PRINTER 165 CPS Serial and parallel interfaces-Eight Selectable character sizes-Single and double width characters-uses standard plain paper - same mechanism as the integral data syst... 1 year old $589, Sta... St., 250...

HEATHKIT H-11/DEC LSI-11 system, 32K Byte storage, reader 1 punch, video terminal, complete software. Cost $4500 assembled, $3500 kit. Like new. Sell for $2250. 305-962-6677. 2058 Griffin Rd., Ft Lauderdale, FL 33312.

FOR SALE: Interdata (Perkin-Elmer) 7/16 Mini with 32KB core, front panel, 50A PWR supply. Includes HS tape reader, interfaces for LP, 2 (TTY), and RS-232 (Full duplex, programmable). Includes manuals and much SW (Basic, Fortran, OS, etc.,). $800 - After 6 P... 2035...

COMPUTER AUTOMATION ALPHA 16; 16 k-word core memory, RTC, PF-R. Modified Mod. ASR-33 TTY. Manuals, utilities, assemblers and many option boards - 16 bit I/O Driver, 16 bit I/O, Asynch modem contr. 64 bit output, 10 bit A/D - D/A. Fairly complete documentation. Up and running in Fortran. Not much more than TTY at $1000. **Herb Sauer, 303-494-8724.**

FOR SALE: Heath H9 video terminal, excellent condition, $175 or best offer. You ship. **[214] 962-4484**

WANTED: DIGITAL Group 32K memory board without memory chips and Phi deck controller board (kit, assembled or not working). W... 1510 NW 35th. T... 32...

PET COMPUTERS moving up to LSI-11. Pet business system priced to sell. PET 2001-16N Computer $800, 2040 Dual Floppy 340K (holds more data than 6 TRS-80 disks) $1,100. Digital cassettes (2) $60 each. System complete with Text Editor, disk sort, database software, real estate software and more $2,100. Call **PAUL** (313)971-8447.

COMPUTER SHOPPER, the new buy, sell, and trade publication, is ready to help you with the latest information on personal, small business and large-system computers, accessories and software.

Each ad-packed issue is full of bargains you are looking for. Included are ads from individuals throughout the United States who are selling their good, pre-owned equipment just so they can trade-up to new equipment coming on the market.

But, COMPUTER SHOPPER'S bargains won't be yours unless you subscribe. This useful, money-saving publication can become your way to communicate with other buyers, sellers, and traders all over the nation.

Whether you are a hobbyist or a part-time user, COMPUTER SHOPPER will put you in touch with the nationwide computer marketplace in time for you to take advantage of bargain opportunities.

Have something to sell? A COMPUTER SHOPPER subscriber probably wants to buy it.

Looking for a part, component or even a complete system? A COMPUTER SHOPPER subscriber probably wants to sell it.

COMPUTER SHOPPER is THE marketplace for anything in computers and is read by thousands of people who are ready to buy.

COMPUTER SHOPPER offers a unique format in which classified ads are categorized for fast location of specific items. Combining this with low individual ad rates — 12 cents a word - makes it the ideal place for buyers and sellers to communicate. And, its mix of individual, dealer, and manufacturer ads enable subscribers to find what they want at the best price possible.

COMPUTER SHOPPER will work for you in other ways, too. If you are just thinking about getting into computers, it can help you learn product availability and prices before you make a decision. And, through the timely ads, COMPUTER SHOPPER will keep you abreast of changes in the market which could create bargain opportunities for you.

BUT COMPUTER SHOPPER cannot work for you unless you subscribe.

Want to look us over first? We'll give you your first issue FREE and then bill you for the next 12. If you are not convinced COMPUTER SHOPPER suits your needs, just write "cancel" on the invoice and return it.

And, to let COMPUTER SHOPPER start working for you right now, with a paid subscription we'll also give you a FREE classified ad to sell your pre-owned equipment or to find equipment you want.

If you don't need to use the free classified ad now, use it anytime during your subscription.

Subscription $10/year, 12 issues plus your first free one. Bank cards accepted. Money back guarantee.

The Nationwide Marketplace for Computer Equipment

COMPUTER SHOPPER

P.O. Box 23 ● Titusville, Florida 32780

Telephone 305-269-3211

MasterCharge or VISA orders only, call **TOLL FREE 800 327 9920**

HOW TO GET RICH AND FAMOUS!

C. Brian Honess

Now, I didn't say it was going to be easy, but it is certainly possible that you could become both rich and famous with your microcomputer! The vehicle for this is going to be some fairly simple little problems—problems that have absolutely defied solution up to the time that *you* solve them.

We start with one that is easily stated, easy to program to test, but it has simply defied solution. It is known as "Fermat's Last Theorem." Pierre de Fermat was a famous number theorist who made a habit of scribbling notes for all the ideas that came into his head as he read. He usually scribbled these notes in the margins of his copy of *Diophantus*. His most famous note said that it is impossible to find whole numbers, *x*, *y*, and *z* which satisfy the equation:

$$x^n + y^n = z^n$$

where **n** is an integer greater than 2. He wrote: "I have found for this a truly wonderful proof, but the margin is too small to hold it." Trying to solve this problem has led to countless other discoveries but still no solution. Money prizes have been offered, but to no avail. As far back as 1816, the Paris Academy proposed the proof of the theorem as its prize problem for a 3-year period. No takers! Several people tried to get Carl Freidrich Gauss to tackle the problem, but he said it was an "isolated proposition" having very little interest for him because he "could easily lay down a multitude of such propositions, which one could neither prove nor dispose of." Incidently, Fermat was a jurist and parliamentarian, and like you, dabbled in mathematics as a hobby! He lived from 1601 to 1665, and certainly didn't have a computer to work with. Euler presented an incomplete proof for **n** = 3, which has been completed now, and there are some other numbers we oan rule out, since:

$$x^{mn} = (x^m)^n$$

(and likewise for *y* and *z*) it would seem that all you need try is odd prime exponents *n,* that are greater than 3.

Incidentally, I can think of a couple of examples where **n** = 2:

$$3^2 + 4^2 = 5^2 \quad \text{and} \quad 5^2 + 12^2 = 13^2$$

and you might practice by finding other (if there are any other) examples where *n* = 2. Let's not count multiples of the above, such as:

$$6^2 + 8^2 = 10^2$$

So get going with your FOR—NEXT loops, and crunch out those values of *x*, *y*, *z*, and *n*. (Note: I don't want to stifle your creativity, so I'll not bother coding a program for testing Fermat's Last Theorem. A few FOR—NEXT loops will get you started though!)

Another fairly simple program to code involves the sum of the cubes of the digits in a number. For all the integer numbers greater than 1, there are conjectured to be only four numbers that are equal to the sum of their digits cubed. In fact, all of the four lie in the interval between 100 and 500. I'll give you the first one:

$$153 = 1^3 + 5^3 + 3^3$$

and now let's begin by finding the other three. First we'll set up a loop and let it run through the numbers 100 to 500, and we'll have to have some way to divide up the number into its individual digits, so we'll use a trick employing the INT function:

```
10 FOR I = 100 TO 500
20 J = INT(I/100)
30 K = INT(I/10) - (J*10)
40 L = I - (K*10) - (J*100)
50 N = J↑3 + K↑3 + L↑3
60 IF N = I GOTO 80
70 GOTO 90
80 PRINT I
90 NEXT I
99 END
```

Of course lines 20, 30 and 40 will produce the individual digits, and line 50 sees if the sum of the cubes equals I, and if it does, directs control to the PRINT statement in line 70. If you're going to check numbers larger than 999 later, you'll have to add another term, and "strip off" this fourth digit in the same fashion, and add it to the sum in line 50.

Now, if you've already keyed this in and tried it, you're way ahead of me, and you're maybe saying terrible things, like "It doesn't work!" and "It didn't even find 153, and it should have!" Well, I admit that on some computers this isn't likely to work as is.

The problem is that we've got some "rounding-off" error problems in line 50. Most computers have this trouble. They usually raise numbers to powers using logs, and this doesn't result in the greatest accuracy. If you change line 50 to read:

```
50 N = J*J*J + K*K*K + L*L*L
```

you'll start getting some results. With the latest version of line 50, of course we're finding the cube in each case, by multiplying the digit by itself three times.

A second way to solve the round-off problem is to make all of the variables integer variables. This can be done by inserting:

```
5 DEFINT A-Z
```

on many BASIC systems, which simply says to make every variable starting with any letter from A through Z an integer variable. We could have made each and every variable an

integer variable throughout the program by adding the correct symbol to the variable name, but this seems lots easier. When I tried each of these versions, one ran much faster than the other, and it wasn't the one I thought it would be. How about when you ran it? (By the way, I hope you found that the four numbers are: 153, 370, 371, and 407).

In a somewhat similar vein as the previous two problems, Fermat also offered a proof that every rational integer is a sum of four rational integer squares (note that zero can be included as one or more of the four). For example:

$$0^2 + 0^2 + 1^2 + 3^2 = 10$$

and,

$$2^2 + 8^2 + 9^2 + 12^2 = 293$$

This one should be both easy and fun to program, and you'll be famous if you can find one that doesn't "fit" the rule!

Here are a couple of "quick" programming problems, not unlike the previous problems. Can you write a program which will find all of the three digit numbers that are composed of three different digits, such that the number is divisible without a remainder by the square of the sum of the digits? I've found 10 of them, which I think is all that there are. One of them is 162, because 1 + 6 + 2 = 9, and 9 squared is 81 and 81 divides 162 exactly twice with a zero remainder, and the digits 1, 6, and 2 are all different. What are the other nine?

Can you arrange all 10 of the digits, 0 through 9, in such a way that you'll make three different formulae, each with a different arithmetic operator (addition, subtraction, multiplication, and division) that will use each digit only once? For example:

$$4 + 3 = 7 \quad 9 - 1 = 8 \quad 30/6 = 5$$

but this example is no good, because I've used the 3 twice, and not used the 2 at all.

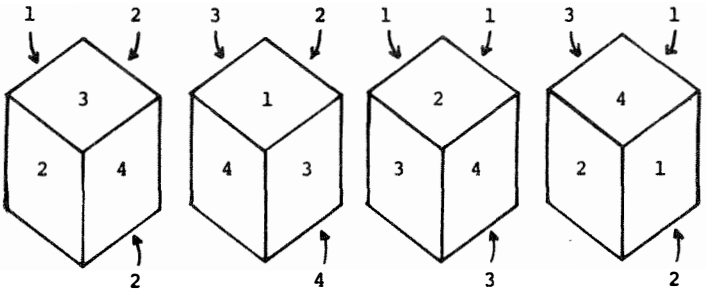
There is, according to Fermat, only one integer solution to this equation:

$$X^2 + 2 = Y^3$$

(that solution being: $x = 5$, $y = 3$). Can you code a simple program which will test all other integer values of x and y, up to the limits of the numbers your machine can handle?

Instant Insanity

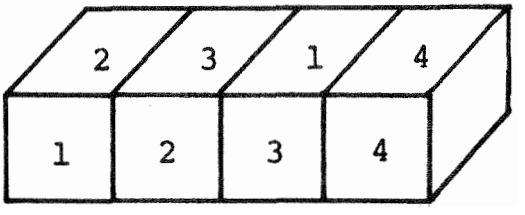
I picked up a little puzzle in England called "Tantalizer" and I understand that there is an American version called "Instant Insanity." All it is, is four cubes of wood, about an inch or so on a side, with a colored dot in the middle of each side. The dots are 4 different colors (red, green, blue, and yellow) and there is no pattern between their placement on each block—the blocks are all different. Here is what they look like:



Notice that I've started getting the problem ready for computer solution by substituting numbers for the 4 different colors. In case you want to make a set of blocks to get a visual perspective on how to go about solving the problem, I used the following code:

Yellow = 1 Green = 2 Red = 3 Blue = 4

but, of course, you could make the cubes with the numbers on them too. OK, now what we want to do is to arrange the four blocks in such a way as to have four different numbers on each side. In other words, we want it to look like this:



As you rotate the figure, you must have one of each of the four numbers visible on each of the four sides. The numbers on each end don't matter, nor do the "inside" numbers. The four numbers can be in any order—just as long as there is one of each. The makers of this puzzle claim that there is only one solution—I'm not so sure. A properly set up BASIC program with a few nested FOR—NEXT loops should tell us if the claim of only one solution is true. (It will also give you the one solution that they claim is the only solution—*I* can't give it to you, because I lost the answer sheet! Sorry 'bout that!)

More Short Number Theory Problems

If I count the lines of a page of a book by threes I have a remainder of 2. If I count the lines by fives, I have a remainder of 2, and if I count by sevens, the remainder is 5. How many lines on the page?

Let's say a printer wants to number the pages of a 288-page book, and he doesn't want to use any of the type pieces more than once. How many different pieces of type will he need? (Answer = 756). Same problem, except he uses 15,321 pieces of type. This time, how many pages in the book? Now, how many pieces of type will he need, if he can use the pieces over and over, but not for the same

page number? In other words, to print page 999, he'll need three different 9's. Solve this problem for the 288 page book and also a 1000 page book.

How close can your computer come to the square root of 2 by using this infinite continued fraction:

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \ldots.}} \quad \text{etc.}$$

How many terms does it take until you cannot increase the accuracy within the bounds of your machine and your BASIC by adding additional terms?

There have been many different estimates of the value of π through the ages: Archimedes proved it was less than 3 1/7 and greater than 3 10/71 and Hero of Alexandria said it was 22/7. Ptolemy said $\pi = 3 + 8/60 + 30/3600$ and Baudhayana took 49/16 as the value. Arya-Bhata gave 62832/20000 as the value, in about the year 530, and in about 650 Brahmagupta said it was equal to the square root of 10. In about 1150, Bhaskara claimed it was 3917/1250 or 754/240, and he just couldn't be sure which (Neither!) (Honess, 1979). The Chinese astronomer Tse Ch'ungchih said it was 355/113 in about the year 450. Leonardo of Pisa thought it was equal to 1440/458 1/3. How close are these values to the accuracy attainable by your machine and BASIC?

How many terms of the series:

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \ldots \qquad \text{(Euler, 1736)}$$

will it take to get to the limit of your machine?

How many terms of the series:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots \qquad \text{(Gregory, 1671)}$$

will it take to reach the limit of your machine? This is a fairly slowly converging series, but is something that can be easily programmed. A series that converges more rapidly is one discovered by Machin in the early 18th century:

$$\frac{\pi}{4} = 4(1/5 - 1/(3.5^3) + 1/(5.5^5) - 1/(7.5^7) + \ldots.) - (1/239 - 1/(3.239^3) + 1/(5.239^5) - 1/(7.239^7) + \ldots$$

How many terms for this series to get to the limit of your machine and BASIC?

In the late 17th century, John Wallis found that:

$$\frac{\pi}{2} = \frac{2*2*4*4*6*6*8*8\ldots.}{1*3*3*5*5*7*7*9*9*\ldots.}$$

How long does it take this series to converge to the most accurate estimate of π your machine can hold?

In the 3rd century, Diophantus of Alexandria claimed there are no integers x, y, and u, that satisfy the equation:

$$x^4 + y^4 = u^2$$

Was he correct? On the other hand, he claimed to have found four integer numbers x, y, z, and u, such that:

$$x^4 + y^4 + z^4 = u^2$$

Can you and your computer find them?

There is one more interesting series for calculating the value of π, which is easy to program and easy to to determine how fast it converges:

$$\frac{\pi - 3}{4} = \frac{1}{2*3*4} - \frac{1}{4*5*6} + \frac{1}{6*7*8} - \ldots.$$

Let's pause and code a BASIC program to solve for the value of π using this series, in case you've had any trouble coding the various series to this point. We'll present one possible coding first, then an explanation:

```
15 T = 0
20 S = 1
25 PRINT" TERMS          VALUE"
30 N = 1
35 FOR I = 2 TO 100 STEP 2
40 X = 1 / (I * (I+1) * (I+2) )
45 Y = X * S
50 T = T + Y
55 PI = 4 * T + 3
60 PRINT N, PI
65 N = N + 1
70 S = -1 * S
75 NEXT I
80 END
```

Since we'll be having a number of terms on the right side of the equals sign, line 15 sets up a location in which to Total these values. Since the sign in front of the term changes from "+" to "−" with each alternating term, I've set up a location called Sign in line 20 and set it equal to 1. More on this trick later. The counter in line 30 is just going to be used on the printout, to let us know how many terms we've accumulated on the right of the equals sign. In line 35 I set up my loop. Initially, I'll look at 50 terls (I go to 100, but I STEP 2, so it produces 50 terms). In line 40, I calculate a particular term and then in line 45 I multiply it by the Sign (which I set to 1 back in line 20). Then in line 50 I add this term to the total of the terms, and print *N* and my current value of π in line 60, after having adjusted for the fact that the formula says what we've really found is (π-3)/4 in line 55. Line 65 adds one to our "times-through-the-loop" counter, and in line 70 we multiply S by −1, which in effect changes the sign for the next time through.

This little program actually does fairly well—as I got a value of 3.14159 when I had reached just 32 terms. This 6-digit accuracy is the best that most versions of BASIC can do, without going to double-precision. Of course you could investigate the convergence of the series by increasing the 100 in line 35, possibly to 1000 or so, and using double-

precision variables. Many versions of BASIC allow you to declare blocks of variable names as being double precision variables by using:

```
10 DEFDBL P-Z
```

This defines all the variables in the program except the counter *N*, to be double precision. Incidently, I guess I'd better give you the value of π so you can check your results:

$\pi =$
3.14159265358979323846264338327950288419716939937519

to 51 places! Incidently, the value has been calculated to a far larger number of places than I've given. In fact, way back in 1959, using a first generation computer, the IBM 704, Mr. F. Genuys calculated it to 10,000 places in an hour and 40 minutes at the Institut de Calcul Scientifique in Paris, France. In the same year, a Mr. Felton computed it to 10,007 plabes using a Ferranti PEGASUS computer in England.

Another number that has been computed to a large number of places is e, the base of the natural logs. The series for computing e is much simpler than any of those for π, and it was calculated to 60,000 places in 1952 by D. J. Wheeler on the ILLIAC computer at the University of Illinois. It took 40 hours to do the job.

$e = 2.718281828459\ldots.$

This is easily found by working with the series:

$$e = 1 + \frac{1}{1} + \frac{1}{1*2} + \frac{1}{1*2*3} + \frac{1}{1*2*3*4} + \ldots.$$

You should have no difficulty coding a program to investigate the convergence of this formula, especially since you don't have to fool with the changing signs in front of the terms, etc. The denominators are called "factorials" and we can use a special notation for them:

$1*2*3*$ can be written $3!$

$1*2*3*4*5*6$ can be written $6!$ etc.

As you code your program, don't evaluate each denominator all over again each time. Just save the previous value and multiply it by the next number in your loop!

Trying to prove that e and π are not transcendental numbers, and will eventually start repeating, has been a job taken on by a large number of mathematicians over the years—and it continues today. There are many other problems of the same type. For example, there is a number called Euler's Constant, which is designated by the Greek letter gamma (γ) which has been computed to over 350 digits and hasn't shown any sign of any sort of cyclic behavior yet. Mathematicians have been trying to show that it can be expressed as a fraction, in much the same way they'd like to show that the other two constants we've looked at are fractions. Euler's Constant is 0.577215664, and can be expressed as:

$$\left(1 + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right) = \log n$$

These days, the calculation of these constants out to thousands of decimal places in efforts to prove that they will start repeating if you go far enough is really not carried out seriously to any degree. The work is now usually done in an effort to study the distribution of the digits, and to see how the constants behave statistically. The digits are tested in any number of ways, using a Chi-square test, for example.

Another interesting test that can be applied is the so-called "Poker" test. Here, we group the digits in groups of five digits, and, for example, if a group was 82882, you'd have a "Full House." The percentages for each of the possible Poker hands with the digits 0 to 9 can be easily calculated. For example, we'd have exactly 100,000 possible Poker hands made up of these 10 digits. Therefore you could expect the following "hands" for the percentage of the times as follows:

1 pair	50.40%
2 pair	10.80
3 of a kind	7.20
full house	.90
straight	.72
4 of a kind	.45
5 of a kind	.01
"loser"	29.52
	100.00%

By the way, you should realize that these percentages are very different from the percentages associated with each Poker hand when you use playing cards instead of the 10 decimal digits!

The distribution of the digits to the right of the decimal point of π is not really far from the frequencies that would be expected. For example, in the first 2000 digits you'd expect 200 of each of the number 0 through 9. In fact, the distribution is as follows:

0	182	5	205
1	212	6	200
2	207	7	197
3	189	8	202
4	195	9	211

This gives a fairly low Chi-square value of 4.11.

In 1685, Adam Kochansky, who was a Jesuit employed as a librarian to King John of Poland, found what he thought was the definitive value of π:

$$\sqrt{\frac{40}{3} - \sqrt{12}}$$

How close was he?

A series that is a little more complex than those we've been investigating was discovered by Sharp in 1717. How fast does it converge?

$$\pi = 2 \quad 3 \left(1 - \frac{1}{3 * 3} + \frac{1}{5 * 3^2} - \frac{1}{7 * 3^5} + \ldots . \right)$$

I hope you have as much fun with these various series and other ideas presented in this article as I had. I've coded a few of the programs for you, and I think there is enough information for you to have lots of fun with the remainder of them. They should converge within the limits of your machine, within a few minutes or so, and they are all very easy to change to double precision, integerization of variables, etc.

C. Brian Honess
22 Shaftesbury Lane
Columbia, SC 29209 ■

COTTAGE COMPUTERY COMES OF AGE
continued from page 49

a typist with a good accuracy and reasonable typing speed, and we'll turn that person into a money-making typesetter in about *five* minutes. Or, more precisely, the Superbrain will!"

But is it really as simple as Simpson says? Indeed, type houses and printers throughout the country are following Cold Type's lead. "Some of our home keyboarders," reports Al Stein, an executive of an east coast house, "are right here in town, some of them aren't. But with our long-distance WATS lines, It really doesn't make much difference. Besides, our keyboarder in Denver brought us in a new 'local' typesetting account just last week."

Yet you might be wondering: Doesn't the nearly coast-to-coast shipping time involved in sending finished type galleys to the Denver client mean a delay for them as opposed to them dealing with a geographically-local type house? Nope! Not when the galleys themselves are sent *electronically!* "We put them on our laser scanner," explains Stein, "and transmit them by satellite link—it's cheaper, in this case, than by phone—to our Denver client, where they are de-scanned. Don't ask me how it all works, but you can sure ask me *if* it does! Let me tell you, I don't know how we ever got along without this kind of transmission."

And it all begins at home. In this case, in a two-bedroom townhouse in a Denver suburb, where a househusband prepares the morning meal for his family, then walks into his den, sits down at his personal computer, and becomes—a typesetter.

Or *whatever!*

Everlastingly at it,

Michael Herbert Shadick
Cedar Square West, Apt E-414
1515 South Fourth Street
Minneapolis, MN 55454 ■

```
410 PRINT,F2,B
420 NEXT F2
430 PRINT,"PRESS ANY KEY TO RUN AGAIN";
440 A$=INKEY$                    'STROBE THE KEYBOARD
450 IF A$="" THEN 440            'NO KEYS DOWN
460 RUN
```

INSPIRATION

Many of you have sent in ideas, suggestions, complaints and compliments, and some of you have asked where a software author gets his inspiration and the enthusiasm to keep hammering out algorithms and subroutines all day long. Well, when I need inspiration I get into my Diesel Rabbit and go up to the highway for a filet of fish sandwich and a vanilla milkshake. So where is the inspiration? It all started right here in Sterling, Illinois about 40 years ago when a traveling salesman named Ray convinced Earl, the local ice cream man, that he'd sell more milkshakes if he used frozen milk than he was selling by mixing milk and ice cream, because they'd be thicker. So many in fact that he'd have to raise his prices and use paper cups, just to keep some metal cups free for the mixer. (In case you haven't guessed, Ray was selling paper cups.)

Well, Earl sold more shakes alright, even after he raised the price from a dime to twelve cents, but he burned up his mixer because the mix was so thick. But Earl was a mechanic, and soon invented a big mixer with a heavy-duty motor in the middle and five spindles around the outside. Ray was so impressed with it that he quit his job with the paper cup company and started selling mixers for Earl. (Ray's wife wasn't impressed, she filed for divorce.)

The mixer business was a little slow at first, so Ray was surprised when a place in San Bernardino ordered several of them. Who in the world would need to make 20 or 30 shakes at a time? A couple of guys named McDonald, that's who. They were making terrific French fries, so the milkshake business was good too. Ray had a long talk with the McDonalds and finally got them to let him start franchising their operation. You probably know most of the rest of the story, if not see if your local library has a book titled *Grinding It Out* by Ray Kroc. Ray earns over a hundred million dollars a year, and it all started because the frozen milk burned up Earl's old mixer.

I was hired to write some inventory and production control programs on the first TRS-80 Model-I here in Sterling for the company that still makes the five spindle Multimixer for Earl Prince. Your local McDonalds has one or two of them, but you will find them anywhere shakes are made.

The point to this story is, it doesn't make much difference who you are or where you live, if you have access to a computer and the time to learn how to use it (and you do, or you wouldn't have read this far) you can be a valuable resource to any of your local industries.

Now, would you like to hear the story about the filet of fish sandwich and the tartar sauce gun?

Gordon Speer
3304 Woodlawn Road
Sterling, IL 61081 ■

EPSON MX-80 GRAFTRAX INITIALIZATION PROGRAM

Joseph Rosenman

Last month, I reviewed the EPSON MX-80 Graftrax modification. What did the Graftrax lack? Good support software. Although the program included in this month's issue doesn't constitute a full-fledged "Graftrax software driver," it will both assist in printer initialization and permit some experimentation with the Bit Graphics mode.

Some of you might recall my EPSON Initialization Program published in the April 1981 issue of *Computronics*. This program replaces the earlier initialization program for MX-80s that have had the Graftrax modification added. While both MX-80s (with and without the Graftrax option) are similar, there are sufficient differences (and additions) to justify a new program. This version incorporates a special machine language subroutine to create a block graphic border around the screen VERY QUICKLY (a purely esthetic consideration). Using machine language subroutines is nothing earth-shattering, but what is especially interesting about this particular subroutine is how it is used. The program is contained in DATA statements, just as you would find it in any other program. This method differs in that the program is not POKEd into any specific address. Instead, it is stored in a string variable. This technique permits any program (255 bytes long or less) to be stored in an address independent location. In other words, you need not reserve memory to run this program. In addition, this will run equally well on a 32K or a 48K machine without any modification. Furthermore, you could store some other machine language program in high memory (unrelated to the Printer Initialization program), without potential memory address conflicts. The address of the string location is passed to the appropriate DEFUSR statement through a calculation involving the VARPTR function. In addition to the Graphic Border subroutine, there is a special machine language print driver. This subroutine will permit any value to be passed to the printer (including a HEX 00H).

The Bit Graphics section is actually a test learning section. Any operations performed in this mode are executed immediately, rather than setting parameters. This section is very helpful when developing patterns of bit graphic codes, or in learning about the Bit Graphics mode of Graftrax. Both the 480 and the 960 mode are supported. The program allows the bit pattern is set by entering the binary pattern (of eight zeros or ones). The pattern can be repeated any (reasonable) number of times. Patterns can be printed one following the other, or with carriage returns separating the different bit graphic patterns.

```
1Ø REM Graftrax Initialization Program Version 1.Ø
2Ø REM 5/82    by J Rosenman
3Ø REM
4Ø CLS : PRINT @ 512 ,"Epson MX-8Ø Graftrax Initialization
Program"
5Ø PRINT @ 58Ø, "by Joseph Rosenman"
6Ø CLEAR 5ØØ : DEFINT A-N : DEFSTR P-Z
7Ø DIM A(22),H(8),P(22),R(24)
8Ø GOSUB 16Ø : ' Set up Variables, Arrays, etc.
9Ø GOSUB 3ØØ : ' Prepare Screen.
1ØØ GOSUB 41Ø : ' Get & Validate input.
11Ø GOSUB 54Ø : ' Process request
12Ø GOTO 1ØØ
13Ø REM
14Ø REM INITIALIZATION
15Ø REM
16Ø FOR I=1 TO 22 : READ P(I) : NEXT I
17Ø FOR I=1 TO 22 : READ A(I) : NEXT I
18Ø FOR I=1 TO 38 : READ N : S=S+CHR$(N) : NEXT I
19Ø FOR I=1 TO 14 : READ N : S1=S1+CHR$(N) : NEXT I
2ØØ DEFUSRØ=-1*(65536 - (PEEK(VARPTR(S)+1) +
PEEK(VARPTR(S)+2)*256))
21Ø DEFUSR1=-1*(65536 - (PEEK(VARPTR(S1)+1) +
PEEK(VARPTR(S1)+2)*256) )
22Ø T=CHR$(27) : ' ESC CHARACTER
23Ø U=CHR$(8) : 'BACKSPACE CHARACTER
24Ø Z=STRING$(6Ø," ") : M=898
25Ø GOSUB 97Ø : 'Initialize the strings.
26Ø RETURN
27Ø REM
28Ø REM PREPARE SCREEN
29Ø REM
3ØØ CLS : N=USRØ(N)
31Ø PRINT @ 74,"EPSON MX-8Ø Graftrax Initialization Program";
32Ø PRINT @ 129,"";  : J=129 : K=161
33Ø FOR I=1 TO 12
34Ø PRINT @ J,I;U;") ";R(I);
35Ø PRINT @ K,I+12;U;") ";R(I+12);
36Ø J=J+64 : K=K+64 : NEXT I : J=J+5 : JJ=J
37Ø RETURN
38Ø REM
39Ø REM GET & VALIDATE INPUT
4ØØ REM
41Ø J=JJ : N=USRØ(N) : PRINT @ J,"Enter Selection: "; :
J=J+18
42Ø PRINT @ J,CHR$(127); : FOR I=1 TO 6 : NEXT I
43Ø V=INKEY$ : PRINT @ J,CHR$(95);
44Ø IF V="" THEN FOR I=1 TO 1Ø : NEXT I : GOTO 42Ø
45Ø PRINT @ J,V; : J=J+1
46Ø PRINT @ J,CHR$(127); : FOR I=1 TO 6 : NEXT I
47Ø V1=INKEY$ : IF V1="" THEN PRINT @ J,CHR$(95); : FOR I=1
TO 1Ø : NEXT I : GOTO 46Ø
48Ø IF V1=CHR$(13) THEN N=VAL(V) ELSE PRINT @ J,V1; :
N=VAL(V+V1)
49Ø PRINT @ M,Z; : IF N<1 OR N>24 THEN PRINT @ M,"Sorry, that
option doesn't exist. Try again."; : FOR I=1 TO 5ØØ : NEXT I:
J=J-19 : PRINT @ M,Z; : GOTO 41Ø
5ØØ RETURN
51Ø REM
52Ø REM PROCESS REQUEST
53Ø REM
54Ø IF N=24 THEN CLS : PRINT "End of EPSON MX-8Ø Graftrax
Initialization Program" : END
55Ø IF N=23 THEN CLS : GOTO 63Ø
56Ø J=J-19 : PRINT @ M,Z; : LPRINT T;P(N);
57Ø IF A(N) <> Ø THEN PRINT @ M+2,"Option ";N;U;":   Final
argument is: "; : LINE INPUT N$ : LPRINT CHR$(VAL(N$));
```

```
58Ø N=USRØ(N) : PRINT @ M,Z;
59Ø RETURN
6ØØ REM
61Ø REM Graphics Mode
62Ø REM
63Ø CLS : N=USR(N) : PRINT @ 81,"Bit Graphics Mode MENU";
64Ø PRINT @ 129,STRING$(62,CHR$(14Ø));
65Ø PRINT @ 258, "1) 48Ø Bit Graphics Mode";
66Ø PRINT @ 29Ø, "2) 96Ø Bit Graphics Mode";
67Ø PRINT @ 322, "3) Return to Main Menu";
68Ø PRINT @ M, "Enter Selection :"; : LINE INPUT N$ :
N=VAL(N$)
69Ø IF N<Ø OR N>3 THEN PRINT @ M,Z; : PRINT @ M,"Sorry,
illegal choice. Try again."; : N=USRØ(N) : FOR I=1 TO 5ØØ :
NEXT I : PRINT @ M,Z; : GOTO 68Ø
7ØØ IF N=3 THEN CLS : GOTO 27Ø : ' WARM INIT MAIN.
71Ø I=USRØ(I)
72Ø PRINT @ M,Z; : PRINT @ M-64,"Enter Bit pattern (Ø off, 1
on) for character:";
73Ø PRINT @ M,"?";
74Ø FOR I=1 TO 8
75Ø Y=INKEY$ : IF Y="" THEN GOTO 75Ø
76Ø IF Y <> "Ø" THEN IF Y <> "1" THEN GOTO 75Ø
77Ø PRINT Y; : H(I)=VAL(Y)
78Ø NEXT I
79Ø PRINT @ M-64,Z; : PRINT @ M,Z;
8ØØ H=H(1)*128+H(2)*64+H(3)*32+H(4)*16+H(5)*8+H(6)*4+H(7)*2+H(1)
81Ø PRINT @ M,"Pattern formed, how many repetitions? ";
82Ø LINE INPUT N$ : N=VAL(N$) : I=USRØ(I)
83Ø IF N=Ø THEN PRINT @ M,Z; : GOTO 68Ø
84Ø IF N>255 THEN N2=INT(N/256) ELSE N2=Ø
85Ø N1=N-N2*256
86Ø LPRINT T; : IF N=1 THEN LPRINT "K"; ELSE LPRINT "L";
87Ø J=USR1(N1) : J=USR1(N2)
88Ø FOR I=1 TO N : J=USR1(H) : NEXT I
89Ø PRINT @ M,Z; : PRINT @ M,"Carriage Return to Printer?";
9ØØ Y=INKEY$ : IF Y="" THEN GOTO 9ØØ
91Ø PRINT Y; : IF Y<>"N" THEN IF Y<>"Y" THEN PRINT U; : GOTO
9ØØ
92Ø IF Y="N" THEN PRINT @ M,Z; : GOTO 68Ø
93Ø LPRINT " " : PRINT @ M,Z; : GOTO 68Ø
94Ø REM
95Ø REM Initialize ID strings.
96Ø REM
97Ø R(1)="8 Lines per inch     "
98Ø R(2)="7/72 Lines per inch  "
99Ø R(3)="6 Lines per inch     "
1ØØØ R(4)="n/216 Lines per inch "
1Ø1Ø R(5)="n/72 inches per line "
1Ø2Ø R(6)="n lines per page     "
1Ø3Ø R(7)="Home print head      "
1Ø4Ø R(8)="Print Italics type   "
1Ø5Ø R(9)="Print Standard type  "
1Ø6Ø R(1Ø)="Set Radio Shack mode "
1Ø7Ø R(11)="Set Standard mode    "
1Ø8Ø R(12)="Set compressed mode  "
1Ø9Ø R(13)="Cancel compressed    "
11ØØ R(14)="Set emphasized mode  "
111Ø R(15)="Cancel emphasized    "
112Ø R(16)="Set double strike    "
113Ø R(17)="Cancel double strike "
114Ø R(18)="Mask Out-of-Paper    "
115Ø R(19)="Set Out-of-Paper     "
116Ø R(2Ø)="Cancel the M.S.B.    "
117Ø R(21)="Set the M.S.B.       "
118Ø R(22)="Reset the Printer    "
119Ø R(23)="Bit Graphics Menu    "
12ØØ R(24)="Exit Program         "
121Ø RETURN
122Ø REM
123Ø REM Data values - Graftrax codes & machine language
code.
124Ø REM
125Ø REM Graftrax activation codes.
126Ø DATA Ø,1,2,3,A,C,<,4,5,":",;,P,Q,E,F,G,H,8,9,<,=,@
127Ø REM Extra input flag for escape codes.
128Ø DATA Ø,Ø,Ø,1,1,1,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
129Ø REM
13ØØ REM Subroutine to paint border around video display.
131Ø REM This subroutine is stored in the S character.
132Ø REM
133Ø DATA 62,143   :' BORD   LD    A,143
134Ø DATA 33,Ø,6Ø  :'        LD    HL,3CØØH
135Ø DATA 6,64     :'        LD    B,4ØH
136Ø DATA 119      :' LOOP1  LD    (HL),A
137Ø DATA 35       :'        INC   HL
138Ø DATA 16,252   :'        DJNZ  LOOP1
139Ø DATA 17,63,Ø  :'        LD    DE,3FH
14ØØ DATA 6,14     :'        LD    B,14
141Ø DATA 62,17Ø   :'        LD    A,17Ø
142Ø DATA 8        :'        EX    AF,AF'
143Ø DATA 62,149   :'        LD    A,149
144Ø DATA 119      :' LOOP2  LD    (HL),A
145Ø DATA 25       :'        ADD   HL,DE
146Ø DATA 8        :'        EX    AF,AF'
147Ø DATA 119      :'        LD    (HL),A
148Ø DATA 35       :'        INC   HL
149Ø DATA 8        :'        EX    AF,AF'
15ØØ DATA 16,248   :'        DJNZ  LOOP2
151Ø DATA 62,188   :'        LD    A,188
152Ø DATA 6,64     :'        LD    B,4ØH
153Ø DATA 119      :' LOOP3  LD    (HL),A
154Ø DATA 35       :'        INC   HL
155Ø DATA 16,252   :'        DJNZ  LOOP3
156Ø DATA 2Ø1      :'        RET
157Ø REM                     END
158Ø REM
159Ø REM Machine Language Direct Out Printer Driver
16ØØ REM Contained in the S1 character.
161Ø REM
162Ø REM
163Ø DATA 2Ø5,127,1Ø :'      CALL  ØA7F  ;Get BASIC Variable
164Ø DATA 235      :'        EX    DE,HL
165Ø DATA 33,232,55 :'       LD    HL,37E8H
166Ø DATA 62,63    :'        LD    A,63
167Ø DATA 115      :'        LD    (HL),E
168Ø DATA 19Ø      :' LOOP   CP    (HL)
169Ø DATA 32,253   :'        JR    NZ,LOOP
17ØØ DATA 2Ø1      :'        RET
171Ø REM           :'        END
172Ø END
```

Joseph Rosenman
35-91 161 Street
Flushing, NY 11358 ■

POCKET COMPUTER CORNER

S. M. Zimmerman, Ph.D. and L. M. Conrad

This Month: Solving Simultaneous Equations with the TRS-80 and Sharp Pocket Computers

Simultaneous equations have few applications by themselves, but are often needed as part of the solution procedure for business, engineering, and scientific problems. The problems associated with solving small sets of simultaneous equations are not great. However, as the number of equations increase, manual solutions become more and more time consuming and difficult to complete with accuracy.

This program we have developed for the pocket computer can handle up to 10 simultaneous equations. We ran a test of ten equations on our pocket computer and found 32 minutes were needed to complete the task. This included three minutes and 15 seconds to print the input data. The same data set was run through a simultanous equation program on our Model I TRS-80 and the amount of time needed was 50 seconds, including ten seconds to print the input data.

If you are not familiar with simultaneous equations, we are going to review an example and some background of the technique. We will then walk through a sample run and interpretation of output. Finally, we will examine the program in detail to show where different activities are performed.

Background for Simultaneous Equations

The procedure used to solve the simultaneous equations starts with the laying out of the coefficients and constants of the equations into a matrix. We created the following set of equations to use as an example to illustrate the steps in the process:

```
Equation #
   1        10 * X1 + 5 * X2  + 4 * X3 = 50
   2            X1 + .5 * X2 + 9 * X3 = 30
   3         2 * X1 + 8 * X2          = -100
```

The lay out of the above coefficients and constants in a matrix looked like the following:

```
                        columns
Matrix # 1        10      5       4       50
        rows       1      .5      9       30
                   2      8       0      -100
```

The first step in the solution process is to look at the first column, starting with the first row, to find the first nonzero value. In this case the 10 in row 1, column 1, is a ten, which means this row is selected.

The first row is divided by the value ten found in row 1, column 1, and the new first row is entered into a new matrix below:

```
                    1       .5      .4      5
Matrix # 2          -       -       -       -
                    -       -       -       -
```

The next step is to get a zero in row 2 column 1, and row 3 column 1. This is done for row 2 by subtracting the first row of the new matrix, times the value in row 2 column 1, from the second equation and entering the new equation as row 2 of the new matrix:

```
old row 2:      1       .5      9       30
-1* new row 1:-1        -.5     -.4     - 5
                ------------------------------
new row 2       0       0       8.6     25
                ==============================
```

Now the same thing is done for row 3:

```
old row 3:      2       8       0       -100
-2* new row 1:-2        -1      -.8     -10
                ---------------------------------
new row 3       0       7       -.8     -110
                =================================
```

The complete Matrix #2 follows:

```
                    1       .5      .4      5
Matrix # 2          0       0       8.6     25
                    0       7       -.8     -110
```

The process now continues with the second column (.5,0,7) starting with the second row a nonzero coefficient is sought. In this case the value in the second row is zero and we must add a new step. The new step is to move the row with the nonzero value into the second row and the row which was in row 2 is moved to where the new row came from. The new matrix is as follows:

```
                    1       .5      .4      5
Matrix # 2 a        0       7       -.8     -110
                    0       0       8.6     25
```

The process of dividing through by the value in the current row column (2,2) position, seven, into the values in row two is the next step:

```
                -          -          -          -
Matrix # 3      Ø          1     -.114285714   -15.71428571
                -          -          -          -
```

The next step is to get a zero in row one column two, and row three column two. This is done for in a the same manner used to get the zeros in the first column. We have performed these steps and printed the final Matrix # 3 below:

```
                1          Ø     Ø.457142857    12.85714286
Matrix # 3      Ø          1     -.114285714   -15.71428571
                Ø          Ø         8.6           25
```

The process is now repeated for the last column of coefficients, column three. Since all the steps are the same, we have performed the steps and written the results below:

```
                1          Ø          Ø          11.52823921
Matrix # 4      Ø          1          Ø         -15.3820598
                Ø          Ø          1          2.906976744
```

The answers are shown on this matrix. The value of X1 is 11.52823921, the value of X2 is -15.3820598, and the value of X3 is 2.906976744. As you will see from our sample run these results are exactly the same as the computer produced. We found no round-off errors. It took several tries to eliminate all of the errors from our manual calculations.

Running the Program

Since the program is not part of a group, we did not identify it in any way. The program may be run in the RUN or DEFinable MODEs by typing RUN and hitting the ENTER key. The first thing you will see is:

```
NO EQ?
```

This question asks the number of equations to be solved. We decided for comparison purposes to use the three equations from our example, so we typed 3 and hit the ENTER key.

The next question was:

```
EQ# 1.
COEFF: 1.
?
```

We answered 10 and hit ENTER. The next thing we saw was:

```
COEFF: 2.
?
```

This time we typed 5 and hit ENTER.

Inputting the data from here on is the same as before, except the constant is called for by name. Be careful and actually type in the value of 0 when such is the case. The pocket computer indexes to the next line when no character is sent as the answer to a question, in a manner different from other computers we know of.

After all data is input the computer produces the following results on either the printer or display. We have listed the following results as it came off our printer.

```
EQUATIONS
#1.
          1Ø.
           5.
           4.
          5Ø.
#2.
           1.
          Ø.5
           9.
          3Ø.
#3.
           2.
           8.
           Ø.
        -1ØØ.
  11.52823921
  -15.382Ø598
  2.9Ø6976744
```

If you obtain SINGULAR for the output of the program, it means there is no solution possible. This happens when all the remaining coefficients in any given column are zero.

Examination of the Program

The program is organized into eight main parts. Each part, the line numbers associated with the part, and the function of the part are identified below:

Part	Line numbers	Function
I	1-3	To input the equations
II	4	To print the input
III	6,17	The loop for each equation
IV	7-8	To select a nonzero coefficient
V	9-1Ø	Switch equations if necessary
VI	11-12	Divide to get 1 in current row/column position
VII	13-16	Subtract to get Ø's in current column
VIII	18	To print results

You may wonder why we made line 4 a subroutine. This was done to enable us to add the ability to print a working matrix whenever we wished. For example, you can add a GOSUB 4 at the end of line 6 and obtain the working results of each cycle.

The ideal way to store and control the data in this program is with a double subscripted variable. Since A(i) is the only subscritped variable in our pocket computer, we had to partition this variable to serve this purpose. Data storage for this program starts with A(10). The program leaves 105 memories available. Twenty six less 10, plus 105, leaves 121 useable memory location for data. A 10 by 11 matrix needs 121 data storage areas, a 11 by 12 matrix needs 132 spaces. We feel we are near the maximum number of variables the pocket computer can handle.

QUESTIONS AND ANSWERS

Conducted by Hubert S. Howe, Jr.

QUESTION

from Brian Peterson, 6807 N. Sheridan Rd., Chicago, IL 60626: The TRS-80 Pocket Computer-II supports PEEK, POKE, and CALL commands. I have tried contacting Radio Shack for information on the machine language used in the computer. They gave me some story about the machine language being "proprietary information." What's going on here? Since when is a machine language on a computer proprietary information?

ANSWER

I would take it to mean that Radio Shack doesn't know what the machine language of the Pocket Computer II is, either. Radio Shack does not manufacture this device. Sharp manufactured the Pocket Computer I. I don't know who manufactures the Pocket Computer II, but I believe it is either Sharp or Panasonic. In any event, the machines probably use something like a calculator chip. Maybe somebody reading your letter will know more.

QUESTION

from Stanley Esposito, Jr., 830 E. Lutton St., New Castle, PA 16101: I own a TRS-80 Model 1 4K and am just being introduced to the computer field. I would like to upgrade the system which uses cassettes, and I would also like to learn the BASIC, RPG and Cobol Languages. Where does a beginner begin?

ANSWER

First, I would recommend that you upgrade to 16K, which you can do by purchasing 4116-type dynamic RAM chips and installing them yourself. Several companies sell these chips for under $20.00 for 16K.

Next, I would recommend that you learn BASIC well. It is the most widely used programming language for microcomputers, and there are many fine books explaining aspects of BASIC programming. Before learining RPG or Cobol, I would recommend that you read books on the subject. It makes little sense to learn these languages unless you can then use them on your computer. These languages are rarely used on microcomputers, although they might be important in the future. Reading about them will certainly help, and might save you valuable time. ■

POCKET COMPUTER CORNER

continued from previous page

Program Listing

```
1:INPUT "NO EQ?";A:D=1Ø
2:FOR B=1 TO A:PAUSE "EQ#;B:FOR C=1 TO A:PAUSE "COEFF:";C
  :D=D+1:INPUT A(D):NEXT C:D=-D+1
3:INPUT "CONSTANT?";A(D):NEXT B:GOSUB 4:GOTO 6
4:H=1Ø:PRINT "EQUATIONS":FOR J=1 TO A:PRINT "#";J:FOR I= 1 TO
  A+1:H=H+1:PRINT A(H):NEXT I:NEXT J:RETURN
6:FOR B=1 TO A:C=B:D=1Ø+(B-1)*(A+1)+B
7:IF A(D)=ØLET D=D+A+1:C=C+1:IF C=A+1PRINT "SINGULAR":END
8:IF A(D)=Ø THEN 7
9:I=11+(C-1)*(A+1):IF C=B THEN 11
1Ø:FOR E=1 TO A+1:H=A(I):G=(B-1)*(A+1)+1Ø+E:A(I)=A(G):
   A(G)=H:I=I+1:NEXT E
11:F=A(1Ø+(B-1)*(A+1)+B)
12:FOR E=1 TO A+1:G=(B-1)*(A+1)+1Ø+E:A(G)=A(G)/F:NEXT E
13:FOR D=1 TO A:G=1Ø+(D-1)*(A+1)+B:IF D=B NEXT D:GOTO 17
14:E=-A(G)
15:FOR F=1 TO A+1:G=(D-1)*(A+1)+1Ø+F:H=(B-1)*(A+1)+1Ø+F
   :A(G)=A(G)+EA(H):NEXT F
16:NEXT D
17:NEXT B
18:FOR B=11+A TO 11+A+(A-1)*(A+1) STEP A+1:PRINT A(B):NEXT B
```

SUMMARY

Simultaneous equations are used to help solve many different business, engineering, and scientific problems. The time needed to solve small sets of simultaneous equations is not great. However, as the number of equations increase, manual solutions become more and more time consuming and difficult to complete with without making errors.

The program developed for our pocket computer can handle up to ten simultaneous equations. The time increases with an increase in the number of equations. Our test of ten equations took 32 minutes. For more than ten equations we recommend the use of larger faster computer.

Steven M. Zimmerman,Ph.D.
College of Business
University of South Alabama
Mobile, Alabama 36688

Leo M. Conrad
Imagineering Concepts
P.O. Box 9843
Mobile, Alabama 36691-0843 ■

BUILD WITH THE ELECTRIC PENCIL

continued from page 50

you will not be able to remember which disks have been updated or how many of he patches have been made to each of your disks. I have added a small self-adhesive sticker to each disk on which I have pencilled in the number of the last patch made.

Jerome I. Weintraub
690 Mountain View Road
El Cajon, CA 92021 ■

PAYROLL SIMPLIFIED

VERSA-- PAYROLL

★ THE ULTIMATE PROFESSIONAL PAYROLL SYSTEM

★ HANDLES ALL PAYROLL FUNCTIONS AND REPORTS

★ QUICK QUARTERLY AND END OF YEAR SUMMARIES

★ PERFECT FOR A SMALL BUSINESS

★ EXPANDS TO HANDLE LARGE CORPORATE PAYROLLS

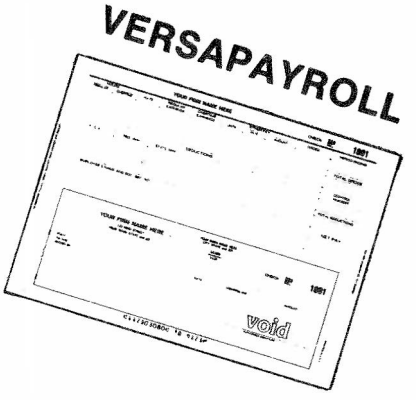
FOR YOUR TRS-80* MODEL I, II, III, 16 • APPLE * • I.B.M.* • XEROX* • ATARI*
ALL MICROSOFT BASIC COMPUTERS

* TRS-80 is a trademark of Tandy Corp. - * APPLE is a trademark of Apple Corp. - *I.B.M. is a trademark of I.B.M. Corp. - *XEROX is a trademark of Xerox Corp. - *ATARI is a trademark of Atari Inc.

HOW IT WORKS . . . .

VERSAPAYROLL is a complete menu driven payroll system that grows as you or your business grows. Your VERSA-PAYROLL acts as a simple payroll system keeping track of all government required payroll information. Just enter the employees, VERSAPAYROLL will perform all the necessary payroll calculations automatically and display to totals on your screen. The user has complete control to accept the totals, to print or not print out a check and to post or not post the total to our **VERSALEDGER** system.

- VERSAPAYROLL automatically prints out your PAYROLL checks. **(IF YOU WANT IT TO)**
- VERSAPAYROLL allows you to override any payroll deduction. **(IF YOU WANT IT TO)**
- VERSAPAYROLL automatically posts all checks written to our VERSALEDGER system. **(IF YOU WANT IT TO)**
- VERSAPAYROLL allows the user to print out PAYROLL checks one at a time. **(IF YOU WANT IT TO)**
- VERSAPAYROLL allows the user to print out all your PAYROLL checks at the same time. **(IF YOU WANT IT TO)**
- VERSAPAYROLL gives you a summary of any employee's year to date payroll totals or all employee totals at any time. **(IF YOU WANT IT TO)**
- VERSAPAYROLL will allow you to correct any error made at any time and automatically refigure all totals. **(IF YOU WANT IT TO)**
- VERSAPAYROLL works in every state. **(IF YOU WANT IT TO)**
- VERSAPAYROLL automatically calculates all federal and states taxes. **(IF YOU WANT IT TO)**
- VERSAPAYROLL allows for all of the standard deductions plus state, city and three miscellaneous deductions. **(IF YOU WANT IT TO)**
- VERSAPAYROLL prints all government required reports. **(IF YOU WANT IT TO)**
- VERSAPAYROLL permanently stores all PAYROLL transactions. **(IF YOU WANT IT TO)**
- VERSAPAYROLL HAS AN ALMOST UNLIMITED CAPACITY . . . .



INTRODUCTORY PRICE
$99.95

Can handle up to 300 employees on a TRS-80 MODEL I, 600 employees on a TRS-80 MODEL III, 1200 employees on a TRS-80 MODEL II, 500 employees on an APPLE II, 600 employees on any single density 8" CP/M computer and almost unlimited capacity on hard disk systems.

CAN BE USED WITH 1 or MORE DISK DRIVES (AND 48K)

VERSAPAYROLL HAS BEEN CREATED WITH THE FIRST TIME COMPUTER USER IN MIND

COMPUTRONICS INC
MATHEMATICAL APPLICATIONS SERVICE

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA AND MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO

NEW TOLL-FREE
ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

VISA AMERICAN EXPRESS

24 HOUR ORDER LINE
(914) 425-1535

*** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE***

HOW MUCH MONEY DO YOU OWE?

VERSA-- PAYABLES

★ THE ULTIMATE ACCOUNTS PAYABLE SYSTEM

★ HANDLES ALL ACCOUNTS PAYABLE FUNCTIONS

★ QUICK PERIODIC SUMMARIES AND REPORTS

★ PERFECT FOR PERSONAL OR BUSINESS USE

★ EXPANDS TO HANDLE LARGE CORPORATE PAYABLES

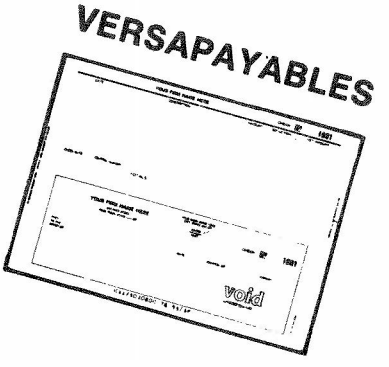
FOR YOUR TRS-80* MODEL I, II, III, 16 • APPLE* • I.B.M.* • XEROX* • ATARI*
ALL MICROSOFT BASIC COMPUTERS

* TRS-80 is a trademark of Tandy Corp. - * APPLE is a trademark of Apple Corp. - * I.B.M. is a trademark of I.B.M. Corp. - *XEROX is a trademark of Xerox Corp. - * ATARI is a trademark of Atari Inc.

HOW IT WORKS . . . .

VERSAPAYABLES is a complete menu driven accounts payable system. It keeps track of all information related to how much money you (personally) or your company owes. It prints all necessary checks and statements on easily obtainable tractor feed forms (or on plain paper). Prints all summary reports to keep you in touch with the flow of money going out of your hands (or leaving your company). In short, VERSAPAYABLES is designed to keep track of current and aged payables. The system maintains a complete record of each vendor, helps determine which transactions to pay by due date within certain cash requirements and prints checks automatically with a detailed check register.

- VERSAPAYABLES prints out your checks. **(IF YOU WANT IT TO)**
- VERSAPAYABLES prints out a detailed check register. **(IF YOU WANT IT TO)**
- VERSAPAYABLES allows for full or partial payments. **(IF YOU WANT IT TO)**
- VERSAPAYABLES prints out vendor mailing labels. **(IF YOU WANT IT TO)**
- VERSAPAYBLES prints out all commonly used ACCOUNTS PAYABLE reports to give you a total picture of money you or your company owes. **(IF YOU WANT IT TO)**
- VERSAPAYABLES integrates with **VERSALEDGER. (IF YOU WANT IT TO)**
- VERSAPAYBLES HAS AN ALMOST UNLIMITED CAPACITY . . . .
  - 400 vendors and transactions per month on sinlge density 5¼" disk drives such as the TRS-80 MODEL I
  - 600 per month on the APPLE II
  - 2400 per month on the TRS-80 MODEL III
  - 6000 per month of the TRS-80 MODEL II
  - 3000 per month on single density 8" CP/M
  - Almost unlimited capacity on hard disk drive systems



INTRODUCTORY PRICE

$99.95

Above capacities are estimates and depend on disk space available and your vendor-transaction mix.

VERSAPAYABLES HAS BEEN CREATED WITH THE FIRST TIME COMPUTER USER IN MIND

COMPUTRONICS
MATHEMATICAL APPLICATIONS SERVICE

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA AND MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO

NEW TOLL-FREE
ORDER LINE
(OUTSIDE OF N Y. STATE)
(800) 431-2818

*** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE***

VISA

24 HOUR ORDER LINE
(914) 425-1535

THE COLLECTOR!

VERSA-- RECEIVABLES

★ THE ULTIMATE ACCOUNTS RECEIVABLE SYSTEM

★ HANDLES ALL ACCOUNTS RECEIVABLE FUNCTIONS

★ QUICK PERIODIC SUMMARIES AND REPORTS

★ PERFECT FOR PERSONAL OR BUSINESS USE

★ EXPANDS TO HANDLE LARGE CORPORATE RECEIVABLES

FOR YOUR TRS-80* MODEL I, II, III, 16 • APPLE* • I.B.M.* • XEROX* • ATARI*
ALL MICROSOFT BASIC COMPUTERS

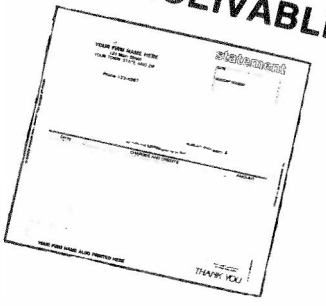
*TRS-80 is a trademark of Tandy Corp. - *APPLE is a trademark of Apple Corp. - *I.B.M. is a trademark of I.B.M. Corp. - *XEROX is a trademark of Xerox Corp. - *ATARI is a trademark of Atari Inc.

HOW IT WORKS . . . .

VERSARECEIVABLES is a complete menu driven accounts receivable system. It keeps track of all information related to who owes you or your company money. It prints all necessary statements, invoices and all summary reports to keep you in touch with the flow of money owed to your company. In short, **VERSARECEIVABLES** is a complete invoicing and monthly statement generating system which keeps track of current and past due receivables.

- VERSARECEIVABLES invoices your customers. **(IF YOU WANT IT TO)**
- VERSARECEIVABLES prints customer mailing labels. **(IF YOU WANT IT TO)**
- VERSARECEIVABLES generates monthly (or periodic) statements at any time. **(IF YOU WANT IT TO)**
- VERSARECEIVABLES uses commonly available preprinted statements and invoices. **(IF YOU WANT IT TO)**
- VERSARECEIVABLES allows partial payments on open invoices. **(IF YOU WANT IT TO)**
- VERSARECEIVABLES prints out all commonly used ACCOUNTS RECEIVABLE reports to give you a total picture of money owed to your company. **(IF YOU WANT IT TO)**
- VERSARECEIVABLES keeps a history of each account, both current and aged. **(IF YOU WANT IT TO)**
- VERSARECEIVABLES is ideal for doctors, lawyers, small and large businesses.
- VERSARECEIVABLES HAS AN ALMOST UNLIMITED CAPACITY . . . .

VERSARECEIVABLES



INTRODUCTORY PRICE
$99.95

400 customers and transactions per month on single density 5¼" disk drives such as the TRS-80 Model I
600 per month on the APPLE II
2400 per month on the TRS-80 MODEL III
6000 per month on the TRS-80 MODEL II
3000 per month on single density 8" CP/M
Almost unlimited on hard disk drive systems

Above capacities are estimates and depend on the customer-transaction mix and the amount of disk space available.

VERSARECEIVABLES HAS BEEN CREATED WITH THE FIRST TIME COMPUTER USER IN MIND

COMPUTRONICS INC
MATHEMATICAL APPLICATIONS SERVICE

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA AND MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO

NEW TOLL-FREE
ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

VISA AMERICAN EXPRESS

24 HOUR ORDER LINE
(914) 425-1535

*** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE***

THE COMPUTRONICS INC.

WABASH WARRANTY FLEXIBLE DISKS

SPECIAL INTRODUCTORY OFFER

wabash®
5-YEAR WARRANTY
8" Maxi-Myte
Flexible Disks

The only 8" diskettes guaranteed to perform for a full five years.

Single Side / Double Density

$39.95 EACH BOX OF 10

wabash® 2-YEAR WARRANTY
5¼" Mini-Myte Flexible Disks

The only 5¼" diskettes guaranteed to perform for a full two years.

Single Side / Double Density

$34.95 EACH BOX OF 10

COATING The Coating process has been recently improved by our revolutionary SR-3000 manufacturing process, resulting in a completely uniform surface, with excellent adherence to the polyester base. The result: consistent signal quality which is crucial to all applications, on all appropriate drives.

PUNCHING State-of-the-Art equipment at the Paoli plant assures precision punching and assembly for every Wabash diskette. Certification and initialization are also care fully supervised at this stage, guaranteeing satisfactory use of Wabash diskettes for years to come.

Ask us about our other in-stock and ready-to-ship products including printer ribbons, pressure sensitive labels, paper, media storage equipment, and more!

CALL TODAY OR SEND COUPON

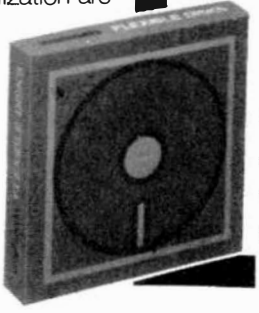
800-631-2818 24-HOUR HOTLINE 914-425-1535

THE COMPUTRONICS INC. 50 N. Pascack Road Spring Valley, N.Y. 10977

☐ Please send me ____ boxes of 8" Maxi-Myte Flexible Disks at **$39.95** per box. Please add $3.00 for shipping.

☐ Please send me ____ boxes of 5¼" Mini-Myte Flexible Disks at **$34.95** per box. Please add $3.00 for shipping

☐ Please send me more information about other Computronics products.



☐ Check enclosed ☐ VISA ☐ ☐ AMERICAN EXPRESS

Credit Card No. _____ Exp. _____

Name _____

Address _____

City/State/Zip _____

Phone (____) _____

ADVERTISING DIRECTORY

8	ABM Products	1-800-854-1555
5	Alpha Products	1-800-221-0916
61	Anitek Software Products	305-259-9397
9	Aspen Software Company	505-281-1634
13	Bealin Corp.	301-490-2477
10	Bible Research Systems	512-346-2181
34-35	BT Enterprises	516-981-8568
Cover 4	The Business Division	305-830-8194
43	Compass Systems, Inc.	603-329-5603
11	Compusoft Publishing	800-854-6505
29, 31	Computech	201-364-3005
59	Computer Plus	617-486-3193
51	Computer Shopper	1-800-327-9920
59	Computerware	714-436-3512
39	Cosmopolitan Electronics Corp.	313-397-3126
61	Data Systems	305-788-2145
8	EAP Co.	817-498-4242
4	Eighty System Newsletter	
61	ETS Center	216-946-8479
25	Allen Gelder Software	59
59	GTI Electronics	717-386-4032
37	Hacks	713-455-3276
Cover 2-3	H & E Computronics	1-800-431-2818
64-67	H & E Computronics	1-800-431-2818
61	Illustrated Memory Banks/IMB	413-663-9648
21	Kwik Software	417-326-7154
21	Lemons Tech Services	417-345-7643
59	Micro Architect	
3	Micro Systems Software	1-800-327-8724
43	Nanos Systems Corp.	317-244-4078
12	Pel/Tek	215-947-2334
3, 17	Powersoft	800-527-7432
22-23	Racet Computes	714-997-4950
68	Radix Technologies	703-385-0900
45	The Softcore Software Co.	513-294-3383
33	Software Options, Inc.	1-800-521-6504
6	SOURCELiB Systems	800-223-1087
41	Virginia Micro Systems	703-491-6502

The Original Magazine for Owners of the TRS-80™ MicroComputer

MODEL I • MODEL II • MODEL III • POCKET COMPUTER • COLOR COMPUTER

Software for TRS-80 Owners

H & E COMPUTRONICS INC.

Monthly Newsmagazine for TRS-80 Owners

MONTHLY NEWS MAGAZINE

- PRACTICAL APPLICATIONS
- NEW EXPANDED BUSINESS SECTIONS
- GAMBLING
- GAMES
- EDUCATION
- PERSONAL FINANCE
- BEGINNER'S CORNER
- NEW PRODUCTS
- SOFTWARE EXCHANGE
- MARKET PLACE
- QUESTIONS & ANSWERS
- PROGRAM PRINT OUTS

...and MORE!

YOUR CHOICE FREE

with your Subscription or Renewal

A.

NANOS SYSTEMS CORP

TRS-80 At Your Fingertips Complete quick reference guide to basic, assembly language and graphic codes —all at your fingertips for all TRS-80 computers (specify computer).

Assembly Language Section N/A for Color Computer

OR

B.

A Word Processor, Data Management System and Cleanup (A maze game) All on cassette. (Add $3 for diskette, add $5 for modified MOD-II diskette version —N/A on color computer or pocket computer.)

H & E COMPUTRONICS INC.

50 North Pascack Road
Spring Valley, New York 10977

CALL TOLL FREE 800-431-2818
(Outside of New York State)

24 HOUR ORDER LINE
914-425-1535

master charge VISA

☐ One Year Magazine Subscription $24 ☐ New ☐ Renewal
☐ Two Year Magazine Subscription $48.00 ☐ New ☐ Renewal
☐ Sample Issue $4 ☐ Mod II Newsletter Subscription $18

Your Choice: TRS-80* at Your Fingertips ☐ or Word Processor/ Data Management ☐
Model I ☐ Model II ☐ Model III ☐ Color Computer ☐ Pocket Computer ☐

NEW! NEW! MOD-II NEWSLETTER 18/Year (or 12 issues)

Name _____ Address _____ City _____

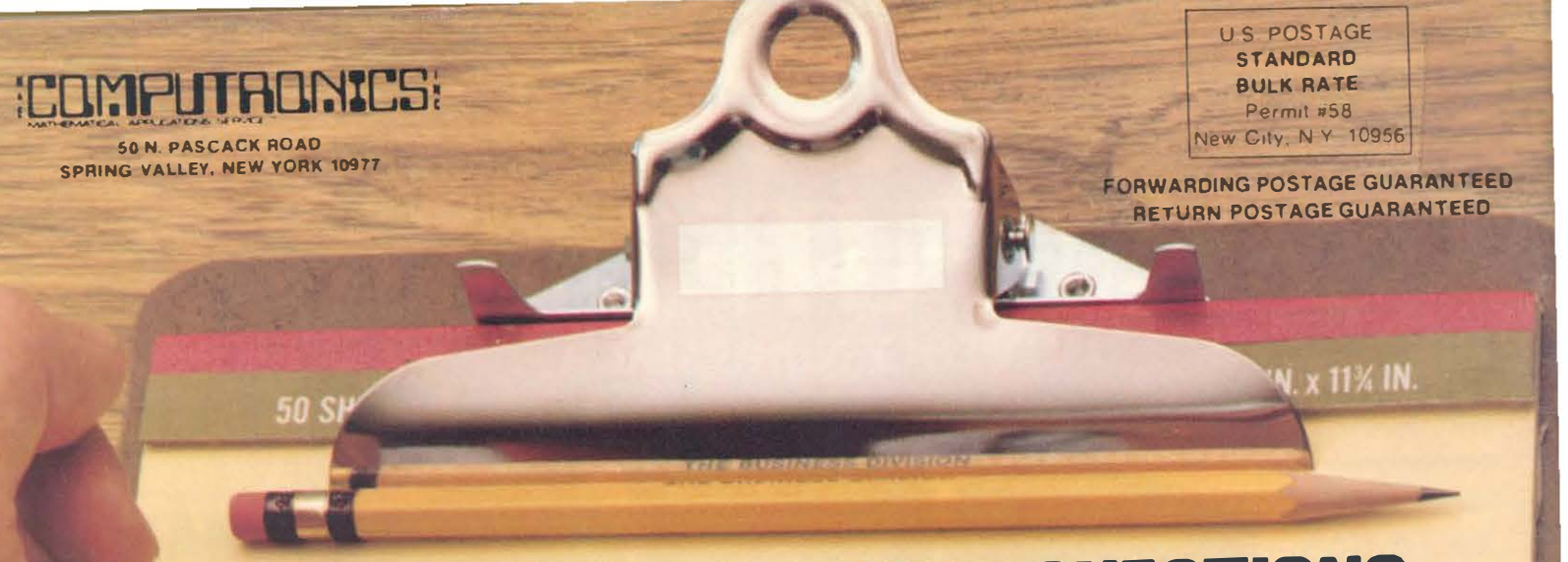
State _____ Zip _____ Signature _____

Credit Card Number _____ Expiration Date _____

Add $12/Year (Canada, Mexico)—Add $24/Year Air Mail outside of U.S.A., Canada, and Mexico
All Prices and Specifications Subject to Change

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp.

COMPUTRONICS
MATHEMATICAL APPLICATIONS GROUP
50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977



PLEASE ANSWER THESE 5 QUESTIONS

	YES	NO
1. Do you have information that must be kept organized and accessible?	☐	☐
2. Do you ever need to perform statistical analyses?	☐	☐
3. Would you like to have a tool that will allow you greater flexibility in managing your own or your company's money?	☐	☐
4. Could you use a mail list program that will be easy to use, maintain an unlimited number of names, allow you great sorting flexibility, and even interact with a word processor?	☐	☐
5. Do you own or have access to a TRS-80 microcomputer?	☐	☐

If you answered "yes" to any of these questions, we can be of assistance to you. Our Maxi Series of applications programs are designed to give you maximum versatility with a minimum of hassle. These programs were created for business use, but you'll find yourself using them for personal applications as well.

Thorough support — Of course, each program comes with in-depth, user-oriented documentation, and is menu-driven to make it easy to use. When necessary, the Maxi programs are compatible with each other, and, whenever pertinent, are interactive with the major word processing and spreadsheet programs published by other manufacturers. Also, we maintain a telephone support line to provide you with any assistance you might require.

Maxi Manager *by Dale Kubler*

Maxi Manager is a remarkable data base manager. Its fast machine language sort complements its large data storage capacity. The sophistication of its data entry, management, and printing capabilities makes Maxi Manager a versatile tool for many applications.

The program now includes Maxi Utility, which allows you to rescue files on diskettes that have been damaged by excessive wear or misuse and lets you expand, add, or delete fields from an existing data base. Let the unmatched capabilities of Maxi Manager handle your data management!

Model I & Model III. Minimum 1 disk drive required . . . 012-0196 $149.99

Maxi Cras *by Dale Kubler*

(Check Register Accounting System)

Maxi Cras is a system that will computerize check writing, recording and analysis for business and personal finance. The system features 223 income and expense accounts, each of which will handle an unlimited number of transactions. Extensive register and report printing capabilities make Maxi CRAS an indispensible tool for managing money effectively.

Model I & Model III. Minimum 2 drives required 012-0145 $99.95

Maxi Mail *by Dale Kubler*

Maxi Mail is a powerful mail list management system that:

1. Is easy to use.
2. Has virtually unlimited storage capacity.
3. Interfaces with the major word processors to generate form letters and other text.
4. Prints mailing labels up to four-across in any format desired.
5. Has virtually unlimited coding capabilities with thirteen fields of information for each record.

Maxi Mail is the most sophisticated user-oriented mail list program available.

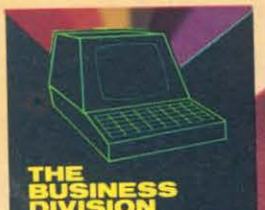
TRS-80 Model III only. 2 disk drives required 012-0148 $99.95

Maxi Stat *by David Walonick*

Maxi Stat is the most useful statistical analysis package on the market today. It was developed to allow maximum flexibility in designing customized analysis. Maxi Stat handles the three main components of statistical analysis:

1. Complete menu-driven codebook creation and editing.
2. Menu-assisted data entry.
3. User-created control files to describe the statistical analyses to be performed and printed out on the variables of your choice.

Model I & Model III. Minimum 2 drives required 012-0153 $199.95



A Division of
Scott Adams, Inc.

Send $1.00 for our 16 page booklet "Getting The Most From Your Micro" All 16 pages are packed with indepth explanations and printout samples from the Maxi Series of applications programs.

THE BUSINESS DIVISION
BOX 3435
LONGWOOD,FL 32750
(305) 830-8194