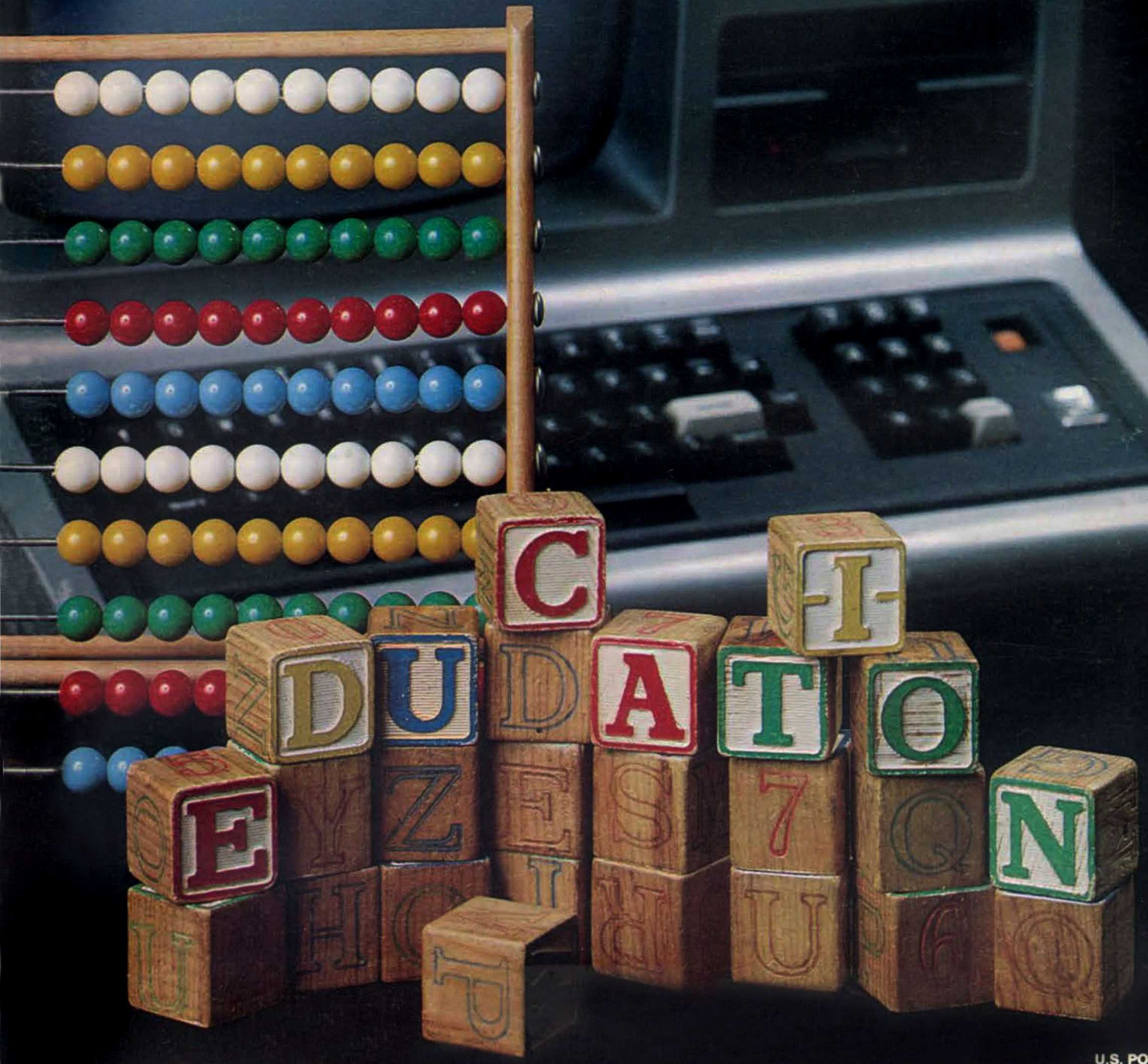
NOVEMBER, 1982
ISSUE NUMBER 51

PRICE U.S. $2.95
£2.00

THE ORIGINAL MAGAZINE FOR TRS-80™* OWNERS

H&E COMPUTRONICS INC.



Cover Photo by Harry Peterson
H&E COMPUTRONICS INC.
50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977
*TRS-80™ IS A TRADEMARK OF TANDY CORPORATION

U.S. POSTAGE
STANDARD
BULK RATE
Permit #58
New City, N.Y. 10956

FORWARDING POSTAGE & RETURN POSTAGE GUARANTEED

"MAXI STAT Is The Most Versatile Statistical Analysis Program Available . . . On Any Micro."

Dr. Steven E. Mayer, PhD., Industrial Psychologist — Maxi User

Focus On: Maxi Stat

Who can use Maxi Stat?

Anyone who needs an analysis of statistical data. Maxi Stat has successfully replaced dedicated mainframe and time-sharing statistics programs for business, financial and marketing researchers, educational administrators, hospitals, medical schools and medical research facilities, agricultural testing, and social scientists.

What does it do?

If you're familiar with SPSS (Statistical Package for the Social Sciences) and what it can do on a mainframe, you have a good idea of what Maxi Stat will do for your TRS-80.

Maxi Stat offers many of the best features of SPSS (modified for microcomputers) as well as features you can't get in any other system, at any price. Maxi Stat is one of the most useful analysis packages available — anywhere.

Read on to discover what this remarkable new tool can do for you . . .

Maxi Stat offers:

- menu-driven operation
- user-created codebook
- the ability to handle up to 255 variables
- machine language code allowing speedy data entry
- fast and easy editing
- hundreds of tasks performed at one time
- reports you can custom design for your specific needs
- ample screen prompts to guide user
- frequent automatic backup of data to help protect against power failures and system crashes
- all phases of research analysis, including:
  - write subfiles
  - frequency distribution
  - descriptive statistics
  - crosstabs & chi-square
  - correlation and linear regression
  - t-test
  - multiple linear regression
  - analysis of variance
  - multiple variable response

Summary

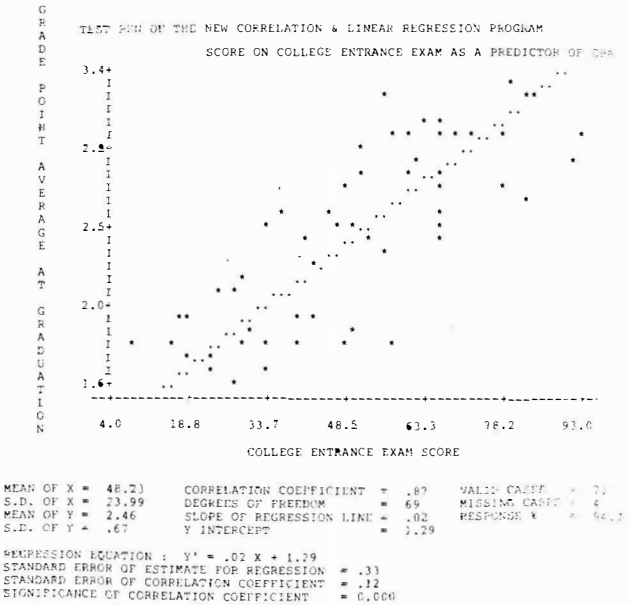
See for yourself what hundreds of other professionals have already discovered — Maxi Stat is THE solution to your analysis problems, and it's never further away than your TRS-80. Pull the plug on your time-sharing system and discover the power and convenience of Maxi Stat today!

THE BUSINESS DIVISION

Look who's already using it!

University of Alaska, Fairbanks, AK. • **Florida Hospital,** Orlando, FL. • **Capital Research Services,** Topeka, KA. • **Center for Financial Studies and Development,** Claremont, CA. • **U.S. Army Psychological Testing Unit,** Wahiawa, HI. • **University of Nottingham Dept. of Social Administration,** Nottingham, U.K. • **Defense Intelligence School,** Washington, D.C. • **Dept. of Recreation and Park Administration, University of Missouri,** Columbia, MO. • **Children's Hospital of Philadelphia,** Philadelphia, PA. • **Kingston City Schools,** Kingston, N.Y. • **Universidad de Monterrey,** Monterrey, Mexico • **University of Michigan, Dept. of Pediatrics,** Ann Arbor, MI. • **Kodak Camera (Australia, Asia),** Coburg, Victoria, Australia • **The Williams Group Advertising and P.R.,** Dallas, TX. • **Medical College of Pennsylvania,** Philadelphia, PA. • **Parkway Day School,** Philadelphia, PA. • **University of Illinois,** Chicago, IL. • **Selkirk Health District,** Nelson, B.C. Canada • **Human Communications Systems,** Reston, VA. • **California Research Center,** Los Angeles, CA. • **WaltRich Marketing Consultants,** Orlando, FL. • **Rutgers University,** Piscataway, N.J. • **Bethal Theological Seminary,** Brooklyn Park, MN • **Seton Hall University, Stillman School of Business,** S. Orange, N.J. • **Plan-Test Associates,** Phoenix, AZ. • **General Mills Corporation,** Minneapolis, MN. • **Shearson, Hayden Stone Investments,** Hollywood, CA. • **Energy Systems Research Institute,** Raleigh, N.C. • **Minnisota Historical Society,** St. Paul, MN. • **Purdue University,** W. Lafayette, ID. • **Nova Scotia Agricultural College,** Truro, N.S., Canada • **Agrico-Chemical Co.,** Wilmington, DL. • **Marketing Associates,** Charlotte, N.C. • **University of Ottawa, School of Psychology,** Ottawa, Ont., Canada • **Rock House Publishing,** Kowloon, Hong Kong • **Center for Applied Social Sciences,** Boston, MA. • **Blake and Dickinson Marketing Research,** Manchester, N.H. • **United Nations,** New York, N.Y., and hundreds more!

Maxi Stat prints many different report styles.



THE BUSINESS DIVISION
A Subsidiary of Scott Adams, Inc

By David Walonick

Maxi Stat is available now for TRS-80 Models I and III with 2 disk drives, 48K RAM, and printer.

$199.95

Soon to be available for IBM PC

ORDER FROM YOUR FAVORITE DEALER or CALL TOLL FREE (800) 327-7172 (ORDERS ONLY PLEASE)
SHIPPING & HANDLING ARE EXTRA - PRICES SUBJECT TO CHANGE WITHOUT NOTICE
BOX 3435, LONGWOOD, FL 32750, (305) 830-8194 (QUESTIONS)

NOVEMBER 1982 **ISSUE NUMBER 51**

CONTENTS

FEATURES

REGULAR DEPARTMENTS

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:

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

International Distributor

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

BITS AND PIECES

Howard Y. Gosman

ON THE COVER

The computer has replaced the abacus. Unfortunately, too many Radio Shack owners don't know it. Our programmers are constantly speaking to Radio Shack computer owners who know very little about their computer beyond where to find the on/off switch. A little bit of knowledge can go a long way. The TRS-80 *by far* outperforms the abacus.

Adults seem to have a bigger problem than children in learning how to use a computer. Recently, I brought home a Radio Shack color computer. Within days, my six-year-old was writing simple programs. I started her off with a game called POP-CORN (the best game designed for children as young as 4 years old).

After POP-CORN, we switched to the Model III. A company called PDI has some excellent software for children 4 and up (including adults). Memory Builder is an excellent start to familiarize your child with the key board. PDI has a variety of other programs to teach your child basic skills. For the adults, PDI has a program called STEP BY STEP (for junior high to adult) which is an excellent step by step workbook approach to learning BASIC on the TRS-80 and interact with your computer (for more information on PDI programs, see our catalog).

COMPUTRONICS CATALOG #10

CATALOG #10 is on the way and should be in your hands by November 1. CATALOG #10 is our largest catalog yet, and the first to include color printing. All current subscribers to the *H & E COMPUTRONICS, Inc. Monthly Newsmagazine* (see your mailing label) will receive a FREE copy of the new catalog. An abridged version of this catalog will also appear in the November issue of *80-MICROCOMPUTING*.

continued on page 6

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 NEWSMAGAZINE 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 51 (OR LESS). THE NUMBER FOLLOWING THE DASH TELLS YOU THE LAST ISSUE THAT YOU WILL RECEIVE. For example, if your subscription number is 16429-51, your subscription expires with this issue (issue #51).

THE MAGIC IS AT YOUR FINGERTIPS.

Dosplus II magic. For you and your business.

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 incredibly 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.

DOSPLUS II DOSPLUS first in quality! First in the industry!

POWERSOFT
A Division of Breeze QSD, Inc.
11500 Stemmons Fwy., Dallas, Texas 75229
To order call toll free 1-800-527-7432
For product information (214) 484-2976

MICRO SYSTEMS SOFTWARE, INC.
4301-18 Oak Circle
Boca Raton, FL 33431
To order call toll free 1-800-327-8724
For product information (305) 983-3390

THE CRYSTAL BALL

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

COLOR COMPUTER NOTES

RADIO SHACK is now selling a new version of the COLOR COMPUTER, but no one really knows about it, and Radio Shack is not really anxious for owners of the older version to know about the new version. Why is this? If you have purchased a COLOR COMPUTER recently, you'll find that it contains a modification that allows the addition of memory to a maximum total of 64K. If you have an older COLOR COMPUTER, the maximum memory size is only 32K, *and you cannot get your computer upgraded!* Which one do you have? Apparently, Radio Shack intends to upgrade the new COLOR COMPUTER even further. We expect to see Radio Shack release a new disk operating system that may add several features essential for a serious business-oriented system, including Random Access file capability and perhaps even support of a rumored COLOR COMPUTER Hard Disk Drive. With a new operating system and 64K of memory, the COLOR COMPUTER may even rival the Model II!

Radio Shack's software for the COLOR COMPUTER is also being improved. COLOR BASIC 1.1 has now been released, and it has some excellent new features for owners of this machine. You can now produce graphics on a line printer, without any additional software. If you have a line printer that has a 7- or 8- bit option (such as the Radio Shack Line Printer VII), all you have to do is set the 7/8 bit switch to 8 bit. If you use a joystick, the TV will no longer display a string of characters when you press the "Fire-When-Ready" button—but if you use Radio Shack's CHESS or CHECKERS programs, you will now have to press the SPACE BAR when the program prompts you to press the "Fire-When-Ready" button, and if you use their BUSTOUT program, you press the space bar to get the next ball. Under COLOR BASIC 1.0, the screen didn't display an S (for Search) or F (for Found) when the computer encountered an "ungapped" file on cassette—now it does. Finally, you can now have full 255-byte records in your data files without the risk of losing information.

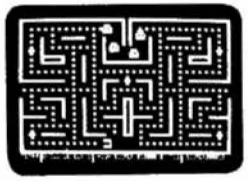
Where can you get a COLOR COMPUTER Assembler/Disassembler for only $3? The September 1982 issue of *80 U.S.* contains an Assembler and a Disassembler, both written in Extended BASIC, along with complete and clear operating instructions.

SERVICE CONTRACTS

The cost of servicing your computer, as many of you well know, usually ranges from exorbitant to astronomical. In businesses especially, the customary way to avoid crippling repair costs is to get a form of "repair insurance": the service contract. The service contract states that, for a single set fee and a set period of time, the servicing company will do any repairs necessary on the covered equipment, for no additional cost. However, rather than concentrating on protecting the users of microcomputers, many service companies have been more concerned with protecting themselves. Instead of paying an astronomical repair bill, you pay a service contract fee that is frequently more astronomical than any conceivable repairs could be. In the past, if you've shelled out for complete coverage of your computer, you probably have been tempted to hope that something major *does* go wrong with your machine, just so that you have a chance of getting your money's worth of repairs! And if your contract calls for on-site service (the repairman comes to your location), then the costs can be even higher. This is quite a dilemma, especially for the business user who can't afford to have a computer out of service even for a few days. Most businesses have got to have a repairman who can come to fix a problem *immediately*. If you do have a service contract, and then don't need any repairs, then you've just paid out a large sum for nothing (except "security"). If you don't have a service contract, then you will not only pay great sums of money for repairs, you'll also pay a

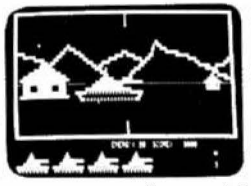
continued on page 6

All photos are actual TRS-80 screens.



SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You run 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. Price: A



ARMORED PATROL

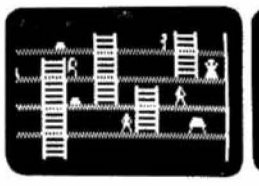
A realistic tank battle simulation. Your view is a 3--D perspective of an alien landscape. Maneuver your T-36 tank to locate and destroy enemy tanks and robots that lay hidden, ready to assault you. Clever graphics create the illusion of movement and dimension. From Adventure International. With sound.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

REAR GUARD

Deadly waves of enemy Cyborg craft attack your fleet from the rear. You are the Mothership's sole defender. You have unlimited firepower but the Cyborgs are swift, nimble attackers. Your abilities are tested hard in this game or lightening fast action and lively sound from Adventure International. Price: B

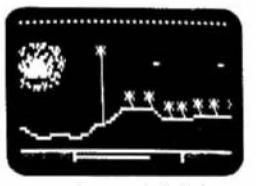
STRIKE FORCE

As the primary defender of a world of cities under deadly alien attack, your weaponry is the latest: rapid fire missiles, long range radar, and incendiary "star shells." Your force field can absorb only a limited number of impacts. A complex game of strategy, skill and reflexes from Melbourne House. Price: A



PANIK

Trapped at an enemy building site, your fate seems certain. Your laser is empty and evil Mzors are closing in. You'll have to climb ladders and think one step ahead of the various monsters. A challenging game for agile minds. From Fantastic Software with voice (Disk has larger vocabulary). Price: B



SEA DRAGON

Your submarine, the U.S.S. Sea Dragon, penetrates a mined enemy channel. Armed with missiles and torpedos, you engage the enemy while navigating unknown waters. Succeed or come to a salty end in this game. 29 screens of horizontally scrolling sea-scrape and sound from Adventure International. Price: B

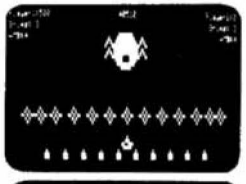
SAVE 10, 15, 20%

BOUNCEOIDS

Huge boulders careen off the walls. You're in the middle, in danger of being flattened. Keep your wits about you as you blast these "bounceoids" from the screen. Large ones break into many small ones. Clear a screen, and enter a fast-paced challenge stage with a chance for big bonus points. From the Cornsoft Group. Price: A

CATERPILLAR

An arcade favorite! Stop these multi-sectioned crawlers before they creep down through the mushrooms. Zap one and it splits into two smaller bugs, each with its own sense of direction. There are moths and tumble bugs too. It all adds up to lots of fun for kids and adults alike. From Sott Sector Marketing. With sound. Price code: A

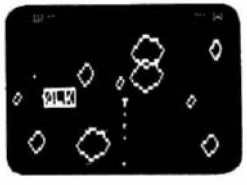


DEFENSE COMMAND

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

CRAZY PAINTER

You have to paint the floor white. We give you the paint and brush. Sounds easy? Hah! You'll be contounded by stray dogs, snakes, sloshing buckets of turpentine, even a ravenous "paint eater." A crazy, imaginative new game with ten selectable levels of skill for new or seasoned game players. Lot's of laughs. Price: A



SUPER NOVA

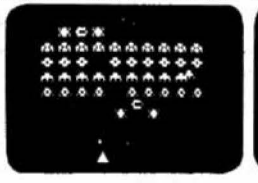
Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones). Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine. Price: A

METEOR MISSION II

As you look down on your view, astronauts cry out for rescue. You must maneuver through the asteroids and meteors. (Can you get back to the space station?) Fire lasers to destroy the asteroids, but watch out, there could be an alien Flagship lurking. Includes sound effects! Price: A

OUTHOUSE

You are the mighty protector of this small (but important) wooden structure. For reasons unknown, a bizarre gang of miscreants wish to vandalize, loot and otherwise destroy the little "halt moon house." Your patrol craft has lasers and smart bombs to deal with this terror. From SSM with sound. Price: A

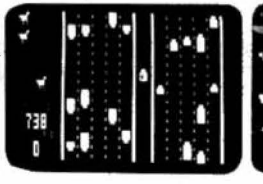


GALAXY INVASION

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! Price: A

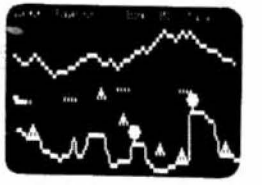
LASER DEFENSE

In this game of ICBM's, high-energy lasers and particle beams, you control the U.S. strategic defense satellite system. From your viewpoint high above the globe, you intercept Soviet nuclear missiles in flight and attempt to destroy their scattered missile silos. With sound from MED Systems. Price: B



CHICKEN

Will the chicken cross the road? That's up to you. Can you guide these helpless little chicks across the perilous 10 lane super highway to safety? Or will you bumble, blithering the blacktop with a storm of chicken feathers? A humourous yet challenging game of nerves from SSM with sound. Price: A



PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missiles attacks. With mites 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. Price: C

"If you purchase Alpha's Joystick you get the exquisite pleasure of enjoying (action games) to the limit of arcade-style realism."

–80 Microcomputing
80 Reviews, Jan '82

FEEL THE POWER...

THE ALPHA JOYSTICK

+ Features the famous Atari Joystick
+ Works with all Model I or III systems
+ Compatible with any other accessories
+ Saves your keyboard from abuse
+ Experiment in BASIC. Use A = INP(0)
+ Complete, ready to plug in and use
+ Model I: plugs into KB or E/I
+ Model III: plugs into 50 pin I/O bus

Price includes Joystick + Alpha Interface + Instructions + Demo Program listing. Please specify Model I or III.

ONLY **$39.95**

14 DAY MONEY BACK GUARANTEE

THE BEST FOR LESS

As you can see, all the best games from the top producers are joystick compatible. These games are fun without the joystick but we hope that you are one of the many thousands who enjoy the advantage of real joystick action.

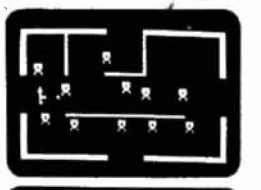
Now you can deduct up to 20% on the price of games: buy any 2 games deduct 10%, buy any 3 games deduct 15%, buy any 4 games deduct 20% from game prices.

TOP TEN

1. **SCARFMAN** - *Alltime favorite*
2. **ARMORED PATROL** - *Super 3D graphics*
3. **PENETRATOR** - *Rave reviews*
4. **STELLAR ESCORT** - *Fast and Challenging*
5. **CRAZY PAINTER** - *Unique game concept*
6. **PANIK** - *Remarkable Voices*
7. **DEFENSE COMMAND** - *Tough struggle*
8. **CATERPILLAR** - *Good rendition*
9. **ROBOT ATTACK** - *With voice*
10. **SEA DRAGON** - *Amazing "Seascape"*

STELLAR ESCORT

The latest super action game from Big Five. As the Federation's top space fighter you've been chosen to escort what is possibly the most important shipment in Federation history. The enemy will send many squadrons of their best fighters to intercept. With sound. Disk version has voices. Price: A



ROBOT ATTACK

Talks without a voice synthesizer, through the cassette port. With just a hand laser in a remote 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. Price: A

LUNAR LANDER

As a vast panoramic moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored -- if you land safely. You control LEM main engines and side thrusters. One of the best uses of TRS-80 graphics we have ever seen. From Adventure International. With sound. Price: A

Toll Free Order Line
800·221·0916
Orders Only, NY & Info call (212) 296-5916. Hours: 9-5 E.S.T.

TAPE: For Model I + III, 16K Level II
DISK: For Model I + III, 32K, 1 Disk
All games are joystick compatable or may be played using the arrow keys.

GAME PRICES

A: TAPE: $15.95 • DISK: $19.95
B: TAPE: $19.95 • DISK: $24.95
C: TAPE: $24.95 • DISK: $24.95

ALPHA Products
79-04 Jamaica Ave., Woodhaven, NY 11421

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

THE CRYSTAL BALL

continued from page 4

ridiculous hourly rate for the repairman to *drive to your location*. For example, Radio Shack's present policy for on-site, non-service contract repairs is typical. In addition to parts and labor for the repair itself, you will pay $52 per hour for driving time (with a 2 hour minimum charge) plus tolls and mileage. With charges like this, you'd better pray the repairman doesn't get caught in a traffic jam! On the other hand, a Radio Shack service contract for a single Model II computer will cost in excess of $1300 per year! That's a lot of money if the only repair you need is to have the disk drives realigned once or twice.

So what's the answer for a computer owner? Well, at the present time we would suggest the following: if there is *any* way at all that you can do without your computer for a while, either by borrowing or buying another computer as a "hardware" backup, then *don't* get a service contract. But, if you are in a business, like most businesses, that needs to have a reliable computer every day, then you have no real choice—you *have to have a service contract*. Your business could depend on it.

Well, we've stated the problem pretty completely. Most experienced users are well acquainted with this dilemma. And here's some good news: service companies are starting to notice that the complaints of users are getting louder and louder. When a company has to charge so much for insurance that their machine keeps running, it does not inspire the greatest confidence in prospective users. If they charge so much, then the machine is bound to break down a lot, right? In order to stay competitive in the present market, companies are making greater and greater concessions to the needs and desires of users. As an example: two and a half years ago, I asked a rather high-ranking Radio Shack computer person (the manager of one of Radio Shack's first computer centers—the most knowledgable member of that company that I've ever spoken to) whether Radio Shack would ever consider *leasing* equipment to businesses. The answer was: "Never! Not in a million years!" Well, Radio Shack *does* have a leasing program now, only two years later. They do *not* turn a deaf ear to the requests and suggestions of users. On the contrary, they have, more than any other company, shown a steadily increasing tendency to accomodate their customers. As an example: in our own business, we absolutely depend on seven TRS-80 computers. To be without a service contract for each computer would be *unthinkable*. Our local Radio Shack repair people *know* how essential our computers are to our business, and when we place a service call they invariably rush over to our office as fast as possible, usually the same day. Not only are they fast, they're also very competent. For *us* a service contract is well worth the cost.

At last, here's a prediction: in the very near future, we will see a dramatic reduction in the cost of service contracts, which will eventually result in an industry-wide movement toward more reasonable pricing for service. This will be partly due to large companies which are slowly becoming more sensitive to users' needs, and to third-party service organizations that do *nothing but* service other companies' equipment, providing competitive pressure on manufacturers to provide service at reasonable rates (we will even see a growing number of *independent* servicemen who run their own repair shop in much the same manner as radio and TV repair shops operate today. All of these developments will result in a lot of happy users, and even more business for computer manufacturers. It's in everybody's best interests.

BITS AND PIECES

continued from page 2

SELL YOUR TRS-80 AND BUY AN APPLE?

Why did you buy a TRS-80 instead of an APPLE? If you did, you made the right decision. Our staff of in-house programmers are becoming increasingly negative about the APPLE. Our programmers who learned programming on a TRS-80 and later had to learn how to program the APPLE,

continued on page 8

YOUR TRS-80®...OUR BASIC BOOK, THE PERFECT COUPLE

LEARNING TRS-80 BASIC starts where the Radio Shack® getting started manuals end, fulfilling the needs of elementary and intermediate level users. If you own or have access to a Model I, II/16, or III, our 544 page BASIC book is a necessity.

IT'S SIMPLE

We receive testimonials every day commenting on the simplicity of this book.

- "very well organized... careful step-by-step progression" **—R.G., Oxnard, CA**
- "extremely logical progression and excellent explanations" **—J.S., Pacifica, CA**
- "easy reading style"**—J.B., St. Louis Park, MN**

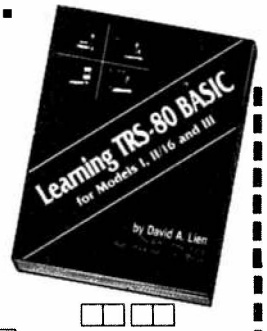
In fact we think it is so simple that....

WE GUARANTEE IT

Try LEARNING TRS-80 BASIC free for 15 days and if you don't agree that it is as simple as we and users like yourself are saying, then send back the book in saleable condition and pay nothing.

Call and order today. **(800) 854-6505** our 24 hour order hot line
(714) 588-0996 our Calif. (8am-5pm) order line

Please send me **LEARNING TRS-80 BASIC** on a 15 day **FREE** examination. At the end of that time, I will send payment, plus postage and handling, or return the book and owe nothing. On all prepaid orders, Visa or MasterCard charges, publisher pays postage and handling, send $19.95 only (Calif. residents add 6%) —same return guarantee. Offer good in USA only. Payment must accompany orders from P.O. Box numbers. Name of individual ordering must be filled in.



NAME _____

ADDRESS _____

CITY/STATE/ZIP _____

☐ CHECK ☐ VISA ☐ MASTERCARD

ENCLOSED ACCOUNT # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ EXPIRATION DATE ☐☐☐☐

SIGNATURE _____

COMPUSOFT™ PUBLISHING
P.O. Box 19669 Dept.M San Diego, CA 92119

CompuSoft Books are available at your local computer and book stores. Allow 4 to 6 weeks for delivery.

TRS-80 and Radio Shack are registered Trademarks of Tandy Corporation

MAILING LIST SYSTEM

For TRS-80* (*Tandy trademark) Model I & III

$89.95

- Simply to use...even for the novice.
- Maintain virtually an infinite number of disks all in continuous alph. or zip order...essential for large lists.
- Sort **2260** entries (2 full 40 track double density disks) in only 32K or an incredible **4640** entries (2 full 80 track disks) in only 48K!
- Super fast sort by alph. or zip order (8 sec. for 1000 entries)...both orders can exist simultaneously on disk.
- High speed recovery of entries from disk...pulls in over 11 per sec!
- Transfer old files to our system.
- Less than 5 digit zips have leading 0's appended.
- **Supports 9 digit zips, Canadian zips.**
- Zip order is "sub-alphabetized".
- System adjusts to any DOS.
- Backup data disks are easily updated as entries are created, edited, or sorted...extremely useful!!
- Optional reversal of name about comma.
- Permits telephone numbers, etc.
- Prints on envelopes or on labels, 1, 2, 3 or 4 across.
- Test label/envelope printing lets you make adjustments with ease.
- Master printout of your list in several formats.
- Selective printing by specific zips or by zip range.
- Editing is simple and fast...automatic search. Batch transfer of edited entries to backup disks.
- Provides for duplicate labels.
- Deleted entries have "holes" on disk filled automatically.
- Automatic "repeat" feature.
- Load and "scroll" through entries.
- Optional "ATTN:" line.
- Plenty of user defined fields with various options for **simultaneously** purging and selecting the printout.
- All Ø's in address labels are replace by easier to read 0's.
- Continuous display of numbers of labels/envelopes printed.
- Each disk entry automatically "remembers" how many mailings have been made.
- Primarily written in BASIC for **easy modification**...embedded machine code for those speed sensitive areas.
- Hardware requirements: 32K, printer, and 1 or 2 drives.

Precision Prototypes
410-F East Roca
Refugio, Texas 78377
512-526-4758

BITS AND PIECES

continued from page 6

hate it. They content that the APPLE version of BASIC is primitive and difficult both to learn and to use. Even worse, our programmers who first learned about computers by using the APPLE now want to do all of their programming on the TRS-80 (and the Model III is now everyone's favorite machine).

So how come the APPLE II is the world's largest selling computer? The answer is easy. Go into a Radio Shack store and ask the question, "Why should I buy your TRS-80 instead of the APPLE?" The usual answer is that "the TRS-80 is a very good computer." Now, try going into a *non*-Radio Shack Computer store and ask the opposite question: "why should I buy an APPLE instead of a TRS-80?" The usual answer is: "The APPLE and the TRS-80 are about the same price. The APPLE does everything that the TRS-80 does, but in color. Why do your check register and use VISICALC in black and white when you can do the same thing in color on the APPLE?"

Well, I just wanted all TRS-80 owners out there to know that you *did* make the right decision. The TRS-80 *is* better than the APPLE. APPLE BASIC is very primitive compared to the advanced BASIC used on the TRS-80 computers. In addition, you have a 64 (or 80) column screen on the TRS-80. The APPLE comes with a 40 column screen that is terrible for word processing and makes business programming nearly impossible. The TRS-80 comes with just about everything you *really* need in a computer. If you purchased an APPLE, you would have to purchase and install separate boards in order to use a parallel or serial printer. The problem becomes even worse when you try to actually make your APPLE print. Some people will tell you how great the APPLE graphics are for playing games, but most owners want to do a lot more than play games with their computer. My reply is, you definitely made the right decision in buying a TRS-80. You even saved about $400 (the difference in price for a comparable APPLE). If you want to play games, you can use the $400 you saved to purchase an ATARI or COLECO video game!

THE NEW RADIO SHACK CATALOG

The NEW Radio Shack computer catalog (RSC-8) is *stunning*. Radio Shack is finally coming of age. Let's just run through some of the highlights:

1. It's hard to believe, but Radio Shack is now selling LDOS. This is a good move on their part. LDOS is non-Radio Shack software (also sold through the COMPUTRONICS catalog). LDOS is an excellent alternative to TRSDOS, with many added features, including support of hard disk drives.

2. The K-8 Math Series is an excellent series of programs for computer-assisted education that should be in every house that has a TRS-80 and a child in grade school.

3. SUPERSCRIPSIT for the Model I and Model III adds all the major features that the original Model I/III SCRIPSIT lacked. Now the Model I/III version is nearly as good as the Model II version (although we at COMPUTRONICS still prefer the ELECTRIC PENCIL).

4. Radio Shack has introduced a 5-megabyte hard disk drive for the Model I or III. The price is $2500, or about $500 per megabyte. The cost to add another drive is $1995 making a total cost of about $4300 for 10 megabytes. However, non-Radio Shack hard disk drives tend to sell for about $2995 (for 12 megabytes), so the Radio Shack drives are presently priced a bit too high.

5. A $995 portable data terminal allows users to access the TRS-80 over the telephone.

6. Radio Shack now has a very popular item that has been asked for by many COMPUTRONICS readers: a printer switch that allows the user to connect two printers to one computer or two computers to one printer.

7. Radio Shack's biggest dud seems to be their new low-cost daisy wheel printer. It sells for $1495. My recommendation: stay away from it and buy the excellent quality $1995 version. The less expensive version is too slow and lacks many of the best features of the more expensive printer.

8. Radio Shack's other new printers are great! As stated above, their hard disk drives are very overpriced compared to the competition, but Radio Shack's new printers priced just right to provide strong competition for

continued on page 15

PROGRAM PREVIEWS

A. A. Wicks

This Month: VersaLedger II

A series of advertisements by H & E Computronics for a new "VERSA" series of business programs piqued my interest because business-oriented programs are the mainstay of this monthly review. I made a request to review VERSALEDGER II, partly because it seemed like a less complex program to start off the series, and partly because no business can be without a check accounting record of some kind, though they may never need complex payroll or inventory-type records. Then too, a good check record may also be used to advantage by an individual in this check-writing society that we live in.

VERSALEDGER II, written by Richard Kaplan, appears to be targeted for the users that I like to see benefit from small computer applications—businesses of from one to 200 employees, for instance, and grossing from a few hundred dollars to a few hundred thousand dollars per year. To place this in perspective with VERSALEDGER II, the program will support the records of up to 3000 checks and/or journal entries per month and 1000 general ledger accounts, depending upon equipment configuration. It will do much more along the way, of course, which will be described—so let's go through its operation in more detail.

VERSALEDGER I VS. VERSALEDGER II

VERSALEDGER II is an upgraded version of the original VERSALEDGER program, which has been available for a year now. In addition to VERSALEDGER's check register capabilities, VERSALEDGER II includes a general ledger, which can produce a myriad of reports (customized balance sheets & income statements, trial balances, account listings, and many more). The most outstanding feature of VERSALEDGER II, however, is that this program may be used as a stand-alone check register, as a stand-alone general ledger, or as a linked check register/general ledger. Any or all of VERSALEDGER II's features may be independently activated at any time.

WHO CAN USE VERSALEDGER II?

VERSALEDGER II will operate with all TRS-80 computers, the Apple II, Atari 800, Xerox 820, and IBM PC. It should also be noted that VERSALEDGER II will operate on "virtually" every CP/M-based computer that can read 8" single density disks. This is quoted from the manual, and is qualified with a small restriction due to the slight variations that may appear between CP/M systems. A configuration program is provided for CP/M users in order to customize VERSALEDGER II for most CP/M machines.

Because of the large number of different computers that may use VERSALEDGER II, considerable material has been presented in the manual that accompanies the program, in detailing the preparation of backup copies of VERSALEDGER II, and program initiation. These instructions are explicit, and the user should have no difficulty in this respect, or in subsequently running the program.

A big boost in this regard for Model I/III users is that the program comes on a disk with a Disk Operating System already on it—the eminent DOSPLUS, which also provides such comforts as lower case operation for Model I users, repeating keys, and full cursor control. I notice that more and more producers of programs are beginning to do this, and must agree that it is a great boon—especially for users with only one disk drive. This service of DOSPLUS operation is not extended to Model II users, however, who must place TRSDOS onto their VERSALEDGER II program disk. This should not be a problem with the instructions provided.

VERSALEDGER II is truly a "turn-key" software package. Initiating computer operation (boot-up) loads the program and displays the main menu. VERSALEDGER II is completely menu-driven—that is, all operations

are performed by the computer as a result of menu selection and screen dialog. A very nice touch is that VERSALEDGER II is supplied with a complete set of sample data files—anyone can load VERSALEDGER II and start experimenting within an hour—a set of check register entries and a general ledger chart of accounts has already been set up.

THE MENUS - QUITE UNUSUAL

The various functions of the system are divided into four major menus (additional subsidiary menus appear during the execution of certain functions). The MAIN MENU is concerned with the maintenance of check register files, the GENERAL LEDGER SUBMENU controls the functions of the general ledger, the FINANCIAL STATEMENT & CHECK REGISTER POSTING MENU allows you to print user-defined reports and to integrate the check register with the general ledger, and the SYSTEM MENU provides you with utility functions for starting new files, and defining report formats.

The menu routines are unusual. A movable pointer is used to indicate which option is being selected (on the Model I, an arrow is used; on the Model III a hand is shown pointing a finger at your selection; on the Model II, the option being selected is displayed in reverse video). Whenever the program returns to the menu, the pointer will indicate which option was last selected. The pointer is moved up and down using the arrow keys.

USING THE CHECK REGISTER

There are two major components of VERSALEDGER II: the CHECK REGISTER and the GENERAL LEDGER. Either component may be used individually, or the two functions may be linked. I will first discuss the CHECK REGISTER functions.

VERSALEDGER II is supplied with a pre-established check register chart of accounts. The user may choose to create his own unique chart of accounts, or he may wish to simply modify the pre-established chart of accounts for his own needs. A large business may have a great many different accounts that they may wish to maintain on their ledgers. An individual or small business operation may have only a few. With VERSALEDGER II, the choice is up to the user.

Every entry to VERSALEDGER II may be posted to any account(s) in the check register chart of accounts. For example, in a payroll check $200 might be posted to account 201 (SALARIES), -$20 might be posted to account 202 (FEDERAL TAX), and -$3 might be posted to account 203 (STATE TAX). When a check register printout (REVIEW CHECKBOOK) is performed, each entry which comprises this check will be printed, along with the total ($177). A listing may later be obtained of each account and its associated transactions over any given period of time. This is extremely useful at the end of a year for tax purposes.

Menu selection of the "Add to Checkbook" option permits entry of check information. As data is entered and completed <ENTER> increments the check number by one. Entry error correction is possible. It is also possible to skip check numbers, returning at a later time to enter these if desired. Deposits and bank adjustments (by typing D and B respectively), are labeled as entered. Of course, you have the option of having these in their own Account categories, too. Also, two deposits or adjustments may be entered with the same label — they will automatically be totalled when "Review Checkbook" is performed. Thus, one may be a negative amount to cancel the previous, but both may be observed for review. Very useful for breaking one deposit into cash, checks, or credit cards, for instance. Or, perhaps to reverse-out a previous bank adjustment, or reversing a check issued and then having payment stopped.

If you wish to print a check, the format is for a New England Business Systems (NEBS) check # 9020.

Locating a check register entry to be viewed, edited, or deleted can be done in two ways. The larger the check register, the longer it will take, ordinarily. But, with VERSALEDGER II, if you know the Item Number (a number which is appended automatically at entry time), then you may go

almost immediately to the entry to be retrieved. Check number 106 may be recalled as T106 (transaction 106) or I5 (the fifth item in the check register). As you may surmise, the first method will take as long to search as the position down the list of the transaction entails, which may be a short time, but it may also be long. You may use either method.

A complete review of the checkbook ledger may be obtained by menu selection. The resulting immediate listing will show the current account, month and year, and will include all checks, deposits and bank adjustments. You have the choice of sending this listing to the printer as well as the screen. Ten items at a time are displayed, and pressing <ENTER> causes the next "page" to appear. Completion of the listing will show initial balance, final balance, and outstanding check balance. The account file and month/year is also shown, of course.

The procedure "Account Development and Edit", which is used to maintain a check register chart of accounts, is rather extensive but far from complex in the performance of the function. Similar options are available for setting up a general ledger chart of accounts.

Some comment is in order regarding the SORT FILE option. (This function will sort a check register file or a group of general ledger journal entries by check number, by account number, or alphabetically according to the PAYABLE TO/DESCRIPTION or COMMENT fields.) The sort by check number option is very useful if a check is entered out of order, i.e., a handwritten check. When this is requested, the routine is loaded from disk, and a status display is provided giving a progress report on the sort. (Another useful feature is that the SORT FILE option is directly available as an option on three of the four main menus.)

USING THE GENERAL LEDGER

Users who have advanced accounting needs may activate the GENERAL LEDGER SUBMENU of VERSALEDGER II. This portion of VERSALEDGER II may be completely ignored by a user operating only a check register; yet, the same user may easily expand at any time to full general ledger capability.

VERSALEDGER II maintains a general ledger chart of accounts, which is independent from the check register chart of accounts. Information stored includes an account number, an account title, type of account (balance sheet/income/expense), a current balance, and a year-to-date balance. Of course, accounts may be added, deleted, viewed, and edited at any time with ease.

Journal entries may be posted to VERSALEDGER II in two ways. One option from the menu permits the user to enter each portion of an entry as a separate journal entry. (The user must be sure the transaction balances to zero.) Another option forces the user to enter balanced journal entries. Each portion of a journal entry is assigned the same journal entry number, and they are grouped together on the transaction register.

An advantage VERSALEDGER II has over virtually every other general ledger presently available is that account balances are immediately updated when a journal entry is made. The end of month procedure need not be run in order to update balances, as is the case with most accounting systems.

FORMATTING THE FINANCIAL REPORTS

VERSALEDGER II prints a multitude of reports (17 in all), including 4 types of trial balances (detailed or summary, monthly or YTD), a transaction register (a listing of all transactions entered in a given month), summary or detailed account listings, and user-definable income statements and balance sheets.

Most commercial general ledger packages available restrict the user considerably in formatting income statements and balance sheets. This is not so with VERSALEDGER II. The options available to the VERSALEDGER II user in generating income statements and balance sheets make this program perhaps the most versatile general ledger package available today—VERSALEDGER II enables the user to format these financial reports in virtually any way desired—almost

continued on page 13

BEGINNER'S CORNER

Spencer Koenig

This Month: The Beginner's Library (Which Books to Buy)

Hello again. It's time once more for inside information on where insiders get their information. Besides reading all of the latest periodicals around, either at the library (there are several hundred professional and semi-professional journals available) or by subscription (like the little gem in your hands), there are dozens of books that belong on your shelf, regardless of what your level is.

When I began my journey into the world of microcomputers, there was much less of a variety from which to choose. There was also much less information accessible written for the neophyte's point of view and experience. Today the problem isn't any easier. There are so many books for every level of programmer that it's too easy to make the wrong decision about which book to buy. Your alternative at this point is to read every review you can get your hands on — or maybe I can help.

In this article, I'm going to give you a list of books that should be on your shelf. If they're not, then plan to get them or leave subtle hints around on what you would like for the coming holidays.

Some of the books are as important today as when I first came upon them (in some cases more important), and some are very new books that should take the place of those less useful. The trouble with reviews is that they usually examine only one book at a time. This review article will examine quite a few, although not in as much depth.

In example 1, I have listed several books which are not only good reference materials but well written for readers at any level. The Osborne books are not really at the same level. Volume zero has been re-done and is a little friendlier than the usual style of most of the other Osborne books I've come across. Volume zero might also be less useful than volume 1 for many of you, especially if you have a text that deals with the same topics.

Those topics covered in Volume zero are: A brief overview of computers today; the parts of the computer that make the whole; choosing a computer; what software is and how to run it; computer languages; introduction to computer math; what happens inside the computer; and a section called "putting it all together", which deals with some terminology and concepts that at times get very technical. It's an inclusive summary of what goes on in the computer in much greater detail.

All in all, Volume zero doesn't talk down to you, but it is very thorough and expects you to be able to think a little on your own. It is arranged like the other books in the series, with more general concepts in bold type and more specific information (often more technical) set in regular type. The advantage of this format is that you can read (and understand) what you like, and you can go back for more as you understand more. The book costs $7.95, and its value is $8.00

I'm sorry, I didn't explain my rating system that I'll be using from now on. It's really very simple. A while back, some friends and I were discussing the latest movies that were around and giving them informal reviews. One of my objections to the way a movie was rated was that you couldn't tell if a film was worth the trouble of going to a particular theatre and spending a certain amount, or waiting for it to come to the local bargain theatre.

Out of that discussion I came up with this scheme: I started rating a film by the dollar value I thought it was worth. For example, the block buster "STAR WARS," came to the movies costing about $5.00, and I gave it a value of $5.00, which meant that it was worth spending the full cost. If you paid less, however, then you got a real bargain.

Another example would be "HUMANOIDS FROM THE DEEP." This flick came to my neighborhood costing $4.00, and I gave it a value of 75 cents. I know what you're thinking: "He had the nerve to go see that movie!"

I can explain. I happened to be in the mood for some really bad movies (a rare event, but true). You get that way when you've spent a lot of time seeing intellectual foreign (often depressing) German or French films (excuuuse me). Well, that day I decided to go escapist, and there wasn't much around which I either hadn't seen or was interested in. So I went to a double feature of "MAD MAX" and "HUMANOIDS FROM THE DEEP." To make matters more interesting, I went to the Matinee when all the kids were there being noisy and commenting on the film while it was going on, and usually adding better dialogue than that of the actors. To make a long story short, I had a great time.

I saw in one sitting what are probably the two worst movies of the last 10 years. Unfortunately, I can't go into detail here about why it was so much fun, but if you're interested, drop me a line and I'll be glad to answer you personally (send a SASE — I could get a mountain of requests).

If by chance "HUMANOIDS" should come to a $1.50 theatre as a double feature, and you're in the mood for movie-trash, then by all means have a good time. I believe it's destined to become a classic, with others like "ATTACK OF THE KILLER TOMATOES."

Well, now that you understand my rating system, you should know that most of the books in my list are worth more than the actual value at which they're sold. Volume 1 of the Osborne series is a prime example. This book is worth at least twice the value. I say this because I've seen books that cost twice as much but contain half as much information. In short, volume 1 is a real bargain.

Volume 1 deals with some of the specific timing specifications, as well as an introduction into programming in assembler for micros. It also covers some of the same

continued on page 14

PROGRAM PREVIEWS
continued from page 12

like VISICALC.

With VERSALEDGER II, headings on these financial reports are completely user-defined and may be printed at any position on a page. Subtotals and totals are again user-defined — any consecutive group of accounts may be totaled and printed at any position of a report.

THE LINKED CHECK REGISTER/GENERAL LEDGER

When both the check register and general ledger features of VERSALEDGER II are utilized, VERSALEDGER II can automatically link the two features. Summaries of each check register account may be posted to the general ledger at the end of a month. In this manner, a detailed record of a month's activity is available within the check register for audit trail purposes, and a summary is available within the general ledger.

THE VERSALEDGER II MANUAL

The VERSALEDGER II manual, which is more than 150 pages long, is comprised of seven major sections:

"An Overview of VERSALEDGER II" explains the use of the system's menus, and describes the interactions between all of VERSALEDGER II's functions.

"Learning VERSALEDGER II" will guide even the totally uninitiated computer-user through the process of making safe disk backups, entering the VERSALEDGER II program, viewing, printing, and manipulating data from the sample data files, and finally, initializing real data files, starting to enter real data, and using the system on a day-to-day basis.

"The Main Menu" describes the operation of all functions related to maintaining the check register and the check register chart of accounts.

"The General Ledger Submenu" describes every operation of the general ledger portion of the program.

"The Financial Statement & Check Register Posting Menu" explains how to generate the user-defined balance sheets and income statements, and how to post the check register to the general ledger files.

"The System Menu" describes the operation of utility functions such as initializing the system, starting a checkbook, general ledger account file, transaction files, and the special user-defined format files that are used by the system when generating balance sheets and income statements.

Finally, a lengthy Appendix contains such valuable auxiliary information as: technical information about VERSALEDGER II's disk files, how to make backups and format disks using DOSPLUS, how to get preprinted NEBS checks, an explanation of error messages, what to do if you have a problem, a summary of differences between VERSALEDGER I and VERSALEDGER II, and, most importantly, *51 consecutive pages of sample printouts generated from the sample data files.*

The options in the four menus are described in a format that provides the user with both an in-depth description of the option, and with a quick reference guide to the screen displays and operator responses for that option. The manual's description of each menu option is divided into three parts:

(1) A brief summary of the option, which includes the option's location (which menu it is found in), a brief statement of its function, and a one-paragraph description of its purpose.

(2) A depiction of every screen prompt and message included in that function, and examples of all possible operator responses.

(3) A section providing 'additional information', which consists of an in-depth description of all uses of the option, how the option relates to the rest of the program. There are also technical explanations for programmers, tips to make the program run faster, etc.

An extensive set of sample printouts is provided in the manual. These printouts trace the activity of a ficticious company over a 2-month period (using the same sample data which is provided to the user on disk). Any questions a user might still have after reading the manual would undoubtedly be cleared by examining these printouts.

continued on page 15

BEGINNER'S CORNER

continued from page 12

information in volume zero, but again in more detail. You begin to understand just how a computer interprets your instructions and how the clock of the CPU keeps things clear so the machine doesn't crash, etc. Concepts such I/O, the stack, computer math, ASCII, indexed addressing and so on are all explained in some detail. As in volume zero, the text is arranged in two kinds of type, so you can read what you understand and go deeper into the subject later on.

The Glossary from Running Press should definitely be in your hands when you begin getting into CPU's. I've bought this little jewel for a number of friends and associates when they were reading about something new that has a great deal of jargon. It could make the difference between understanding and maintaining interest or boredom and stopping dead in your tracks, because some word or idea wasn't fully explained.

Crash Course covers much of the same material as the Osborne texts, but it is easier to deal with. It's organized as a self-teaching guide, with short explanations every step of the way. It's well done and quick reading for any level. If there are any teachers out there, this book has your whole year's lesson plans worked out for you. If you're only looking to get additional information on a particular point, this book is the easiest with which to work.

The four books in example 1 contain all the information you will need to get a good start and a firm backround in computers. Joseph Rosenman confessed to me that, every now and then he still refers to Volume 1, whenever he needs to be refreshed on some inside information.

```
**********************************************************
EXAMPLE  1.   COMPUTERS IN GENERAL

                          P=price  V=What I think it's worth

An Introduction to Microcomputers vol Ø, The beginner's book
 Osborne/McGraw-Hill      by  Adam Osborne and David Bunnell
                          P= 7.95  V= 8.ØØ
An introduction to Microcomputers vol 1, Basic concepts
 Osborne/McGraw-Hill      by  Adam Osborne
                          P= 12.5Ø V= 25.ØØ
Running Press Glossary of Computer Terms, An insider's guide
to the language of the experts
 Running Press            by  John Prenis
                          P= 2.95  V= 5.ØØ
Crash Course in Microcomputers
 Howard Sams inc.         by Louis E. Frenzel jr.
                          P= 19.95 V= 22.95
**********************************************************
**********************************************************
EXAMPLE 2. BEGINNERS AT BASIC

Learning TRS-8Ø Basic for the Models I, II/16, and III
 Compusoft                by  David A. Lien
                          P= 19.95  V= 35.ØØ+
Learning level II
 Compusoft                by  David A. Lien
                          P= 15.95  V= 16.ØØ

TRS-8Ø Data File Programming Model I/III
 (also called Data File Programming in Basic)
John Wiley and Sons inc.  R.S. Cat. No. 62-2Ø85
                          by Finkel and Brown
                          P= 11.95  V= 14.95

Basic Programmer's Notebook
 Howard W. Sams & Co. Inc.  by E. R. Savage
(a good book, overpriced) P= 14.95  V= 7.95
**********************************************************
```

As you can see, in example two there are some excellent books available for the Basic beginner. Some, however, are good (worth having), but I think are still overpriced. On occasion I go to a computer show and see some books on sale. I suggest that you do the same, and save yourself a lot of money in the process.

I suggest you pick up either of the two texts by David Lien. Either one would probably be all you'd need to learn the language or the machine. Lien's style is excellent, and his reputation for being complete is renowned, as I'm sure you have seen or heard from others. I intend to suggest its use as a reference and text at any school where I teach.

The book by Savage is a collection of Basic subroutines that can help you write better programs. The text also contains advice about how to write better programs and to stay out of trouble (avoiding bugs) to begin with. It's a worthwhile book. I wish it weren't as expensive.

TRS-80 Data File Programming is one of the best books on the subject. It is well written and easy reading. I have a lawyer friend who uses it to learn not only the language but also techniques that time and time again have saved him effort. The techniques described are applicable to game writing as well as inventory control. If you think about it, in some cases there really isn't much difference between games and inventory control. The topics covered are: clarity; readability; logic; data entry; error checking routines; sequential data file handling; and random access. There are tons of examples.

```
**********************************************************
EXAMPLE 3.    INTERMEDIATE TO ADVANCED

Programming Techniques for Level II Basic
 R.S. Cat. No. 62-2Ø62          by William Barden Jr.
                          P= 4.94  V= 9.95

Intermediate Programming for the TRS-8Ø Model I
 Howard W. Sams & Co. Inc.      by David L. Heiserman
                          P= 9.95  V= 1Ø.ØØ

Fast Basic, Beyond TRS-8Ø Basic (Model I/III)
J.  Wiley & Sons                by Gratzer and Gratzer
                          P= 14.95 V= 15.ØØ

Basic Faster and Better
 IJG Computer Services          by Lewis Rosenfelder
(The best is overpriced)  P= 29.95 V= 19.95
```

continued on page 52

PROGRAM PREVIEWS

continued from page 13

The format of this manual is excellent from the standpoint of readability and organization. There should be no question in the reader's mind as to any procedure or discussion—the text is very understandable in this respect, even to a person not particularly adept at financial recordkeeping. The three-part explanations of all functions transforms the manual into both a tutorial manual and a quick reference guide—a format which will please users at all levels of computer literacy, from a complete novice to a professional programmer. The manual rates a 8 on our usual scale of 1 to 10.

TECHNICAL SUPPORT

VERSALEDGER II has full customer support should you encounter any difficulties. This is an increasing and welcome trend among the better program producers, and indicates an interest in more than just selling a program. A Customer Service number is provided in the documentation, for any calls of this nature.

CONCLUSION

VERSALEDGER II is probably one of the easiest programs around for any level of financial recordkeeping, from simply maintaining a personal check register to performing client writeups for multi-million dollar companies.

The fact that the computer asks for minimum data, the user provides it, and the computer and printer do everything else, makes it ideal for relatively unskilled computer operators. The fact that it can perform sophisticated financial functions makes it ideal for people with more extensive computing requirements. Finally, the fact that it can be operated at any level between a simple check register and an extensive general ledger makes VERSALEDGER II ideal for anyone who needs a record of financial activity.

VERSALEDGER II—H & E Computronics, Inc. Spring Valley, N.Y.—Introductory Price until December 31, 1982: $149.95

A. A. Wicks
30646 Rigger Road
Agoura, California 91301 ■

BITS AND PIECES

continued from page 8

other manufacturers. All Radio Shack printers are excellent quality and extremely reliable (especially in comparison to some of their original printers). Radio Shack gets an A-plus for their selection of printers. One warning about choosing a printer: if you plan to use your printer in a business application, *buy a full sized printer* (with a 14-inch wide carriage). Most business software is designed to create reports on a printer capable of printing 132 characters per line at 10 pitch. The $1795 All-Business printer prints at 220 characters per second and is capable of correspondence quality print.

The new Radio Shack catalog does show excellent progress by Radio Shack. They are finally meeting the competition head on. ■

LETTERS TO THE EDITOR

Cassette Merge

In response to the request from Mr. William Byberg and your reply to him in reference to a Model III cassette merge program (August 1982 issue), perhaps the following will help:

Both the Model 1 and the Color Computer have had cassette merge utilities published in various magazines. Since Radio Shack maintains ROM compatibility between the Model 1 and Model 3, at least for the most part, the Model 1 merge should work on the Model 3. (The pointer addresses are the same for this particular use).

I would suggest that Mr. Byberg try this short routine:

1. Write down the contents of 16633 and 16634 after loading the first program.

2. If 16633 is 2 or more, subtract 2 from it; poke this value into 16548. Also poke 16549 with the value in 16634.

3. If 16633 is less than 2, add 254 to the value in 16633 and poke it into 16548. Also subtract 1 from the value in 16634 and poke it into 16549.

4. CLOAD the second program.

continued on page 19

ARRAY OF HOPE FOR BASIC PROGRAMMERS (PART 3)

Arne Rohde

5.2 Non-sequential Searching

Irrespective of the optimization techniques employed, sequential searching can become very slow as array size increases. Even with less than 10 elements in an array a non-sequential search may be faster than a sequential search, although there may be other reasons for using sequential searching instead. One of the common reasons is that the array must be stored in a particular sequence, but sometimes a search is performed on a value other than the normal key, and when more than one array element may fulfill the search conditions. For example finding all custumers who have purchased more than a certain number of items in a particular month would normally be done by sequential searching if this was not the normal access requirement for the particular array, in which case it could be sorted beforehand.

Many of the non-sequential search techniques require that the array be sorted into a particular sequence on the key being searched for. Some of the sorting techniques will be covered in a later article, but we have already seen how a binary tree has an implicit sequence which can be useful for searching. For most of the following search algorithms, we shall assume that the array has been sorted in sequence before the search is executed.

5.2.1 Binary Search

One of the better known non-sequential searching techniques is the binary search. The search derives its name from the fact that for each key comparison the remaining unsearched portion of the array is divided into two parts. The search can be illustrated easily with a small array containing a number of keys in ascending sequence. We shall assume an array with 7 elements contains the key values:

1 4 9 16 25 36 49

If we are searching for the key value 9, we first look at the central element in the array, in other words the one with key value 16. Since the required key value is lower, we have restricted the remaining search to those elements with a lower key value, or the first 3 elements in the array. The next element to be compared is the center element of the portion remaining to be searched, the element with key value 4. Since the required key value is higher, the remaining search is restricted to the elements with key value larger than 4 but lower than 16. There is only one element fulfilling this condition, and this is the required element.

To find the element 3 comparisons were required, exactly the same as if the search had been sequential, starting from the beginning of the array. However this was the worst case for the binary search, whereas the worst case for the sequential search would require 7 comparisons, with an average of 3.5 comparisons for finding a random key. With the binary search, assuming the key is found, one key value requires one comparison (key 16), two require two comparisons (4 and 36) while the remaining four require three comparisons. Three comparisons are also required to ascertain that the required key is not in the array. The average number of comparisons for finding a key with the binary search is thus 17/7 or approximately 2.4, one less than that for a sequential search.

For arrays of this size, the binary search will not offer any measurable advantages, since more complex calculations are required before each comparison can be made. If instead of 7 elements we need to search through 127 elements, the average number of comparisons for the sequential search will be 63.5 but for the binary search it will be (1*1 + 2*2 + 4*3 + 8*4 + 16*5 + 32*6 + 64*7)/127, or 769/127 which is approximately 6 comparisons on average. The worst case for the binary search is 7 comparisons, whereas the sequential search requires 127 for the worst case. On avarage the sequential search requires 10 times more comparisons to find a particular key, and with these array sizes the difference in speed can be quite noticeable.

To search an array A containing AE elements (with subscripts from 0 to AE-1) sorted into key sequence for the key value AR, the following code could be used:

```
100 IL=-1: IH=AE: REM Low and high searched
110 I=(IH+IL)/2: REM Find mid point for search
120 IF I=IL OR I=IH THEN PRINT "Key";AR;"not found":
GOTO 150: REM Subscript already used
130 IF A(I)>AR THEN IH=I: GOTO 110
ELSE IF A(I) < AR THEN IL=I: GOTO 110
140 PRINT "Key";AR;"found at position";I
150 etc.
```

It is also assumed here that the subscript values I, IH, and IL are defined as integers. IL points to the lowest subscript which has already been compared, and IH to the highest subscript which has been compared. Initially these values are just outside the range of the array, since none of the elements have been compared. If we look at the 7-element array again (AE=7 and AR=9), the execution sequence will be:

```
100 IL=-1:IH=7
110 I=3
120
130 IH=3 since A(3)>9
110 I=1
120
130 IL=1 since A(1)<9
110 I=2
120
130
140 "Key 9 found at position 2"
```

The code for the binary search is relatively compact and should be easy to follow, but there are a few points to note concerning the efficiency of the code. The first is that either one or two comparisons are made to check the key value, and two comparisons will be made for the required

continued on page 18

JUST LIKE FIRE

Michael Herbert Shadick

We've all heard the same or similar claims to the effect that computers are somehow capable of changing us as human beings, or even that they will eventually (horror of horrors) render us somehow less than human. Such claims are, of course, not only complete hogwash, but do in fact fly in the face of human history. Indeed, to even *utter* claims such as these is to overestimate the computer's power over us, and, even more significantly, to grossly *underestimate* our own powers and capabilities.

I ask you: has the coming of the computer age changed us, or even *threatened* to change us, for example, physically? The answer is (with the possible exception that we might be becoming slightly more sedentary, cerebral creatures)—of course not! Likewise, computers cannot change us in any other way—not emotionally, not spiritually, not even, if you will, metaphysically! Unless, of course, we *wish* to change in any (or all) of those areas. And then, the computer can serve as a most effective tool in helping us bring about whatever changes we desire in ourselves.

Computers, then, are no less—and no more than *tools*. The same can be said of the hammer, or of the loom, or of the telephone, or of any of humanity's other inventions. For each of them, in their own ways, have altered the course of human history. Yet it isn't our race's inventions *per se* which change things, but rather, the ways in which we *use* them. That's a mighty important distinction, which the so-called "computer critics" invariably fail to appreciate. It was, is, and always will be we who use our inventions, *not* vice versa. The computer is just another tool in our hands—albeit doubtless one of the most potent ones our species has ever possessed.

Fire, in the early days of its conquest by our prehistoric predecessors, was probably accuses of the same things which computers are accused of today. Certainly the conquest of fire *did* change the lives of its conquerors! But the species itself—*our* species—was then, and is now, relatively rigid and rather fixedly inflexible. We—each and every one of us—have the same basic needs, the same wants, and the same desires as those of our parents, our great-grandparents, or of our ancestors from however far back one wishes to go. The underlying configurations of human existence remain constant from generation to generation. The only aspects of our individual and collective existence which *do* change, then, are the ways in which we *meet* our needs, wants, and desires. That is, the *tools* we use to get what we want out of life.

There has never been a better tool, nor one with greater and more far-reaching capabilities and potentialities, than the computer. So in that regard, computery will doubtless continue to have increasingly profound effects on our lifestyles. But to say that the computer will *change* us, altering who and what we are—I wouldn't bank on it, not for a nanosecond! Ours is, after all, a mighty (and mighty *durable*) species. We have conquered ever more sophisticated natural phenomenon, and have turned those forces of nature into tools at our beck and call. Fire itself was one of our race's earliest major conquests. Electrical energy, as utilized in today's computers, is one of our latest major conquests—but, without a doubt, not our last. Fire and computers have this in common: they represent natural forces which have always been around us, but which we have now placed in our hands. They—the forces, whatever they may be—are not about to take control of the very species which conquered them!

So the next time some self-styled skeptic tells you that computers are "taking over" or some such inanity, you might reply with a statement to this effect:

"Yes, computers are beginning to effect every facet of our lives . . . just like fire!"

Everlastingly at it,

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

continued from page 16

key value. This tends to slow down the search slightly, compared to the sequential search which only uses a single comparison. Another factor is that a specific test is made to terminate the search when the key is not found. This was not necessary for the sequential search, and again it will slow down the binary search. The loop CAN be coded in a single line to speed up processing slightly, but the legibility suffers accordingly. The single line, replacing lines 110, 120 and 130 in the above code, could be as follows:

```
11Ø I = (IH + IL) / 2:
    IF IH - IL <= 1 PRINT "Key"; AR; "not found": GOTO 15Ø
    ELSE IF A(I) > AR THEN IH = I: GOTO 11Ø
    ELSE IF A(I) < AR THEN IL = I: GOTO 11Ø
```

The termination condition has been changed so that it requires only a single IF statement, and the key comparisons can be made in the ELSE portion of this comparison. If speed is the main objective then this version, perhaps with deletion of unnecessary spaces, could be used. For even faster speed the termination test could be moved, since it is executed unnecessarily when the key is found and on the first pass. The reason for the present position is that the code will execute correctly if the array contains zero elements, and if it were moved after the key comparisons it would have to be duplicated, once for a lower key and once for a higher key.

In a test to compare the speed of a sequential and a binary search an array containing the integers in ascending sequence was constructed. A search was then made with 50 random integers, most of which would be found in the array, and with different array sizes. The following times were obtained, but since the search keys were generated with the RND function, they cannot be reproduced exactly. The times, however, are representative of the time differences which can be expected as the array size increases.

# array elements	Time in seconds to find 5Ø key values	
	Sequential	Binary
15	8.2	6.7
31	16.7	8.5
63	3Ø.6	1Ø.8
127	54.1	12.Ø
255	118.4	13.8
511	233.Ø	15.1
1Ø23	458.9	17.3

The sequential search times are approximately doubled for each doubling of the array size, whereas the binary search has an almost constant increment in time for each doubling of array size. These observations correspond closely to the theoretical values expected for the search times.

5.2.2 Combined Searching

The binary search requires a division by 2 to calculate the subscript of the next array element to be compared. In some programming languages this division or subscript calculation may not be simple, or it may be time consuming. To avoid this calculation a combined search method can be used, the first part of the search consists of comparing elements at fixed distances from each other until an element with a higher key value is encountered. The next part consists of a sequential search from this point until the required key value is found, or it is determined that the value cannot be found. The number of comparisons required is considerably less than that for a pure sequential search, but greater than that for a binary search. The ideal distance for the first part of the search would be the square root of the number of array elements, giving an average total number of compares equal to this value. Thus 10 compares would be required for an array with 100 elements on average, compared to 50 for a sequential search and slightly less than 6 for a binary search. The code in Basic is slightly more complex than either of the other methods since two search loops are required, and the method would probably not be useful as a Basic routine, but in assembler or other languages it could be a useful addition to the non-sequential search repertoire. Using our array and variable definitions as before the algorithm can be coded as follows:

```
1ØØ I = Ø: IS = SQR(AE) + 1: REM Make step non-zero
11Ø IF I + IS < AE:
    IF A(I + IS) <= AR THEN I = I + IS: GOTO 11Ø
12Ø IF I < AE: IF A(I) < AR THEN I = I + 1: GOTO 12Ø
    ELSE IF A(I) = AR PRINT "Key"; AR; "found": GOTO 14Ø
13Ø PRINT "Key"; AR; "not found"
14Ø etc.
```

The two search loops are in line 110 and line 120. The first loop steps through the array until a key value higher than the desired one is found. The sequential search then continues from the previous value which was not higher than the desired value. Obviously we do not need to extract the square root of the number of elements in order to use the method, and for arrays of reasonably static size a constant would be used instead. For arrays which vary considerably in size a simple calculation or array lookup could be used instead to find an appropriate step value. The square root is used here since it yields the optimum step value for any array size. As in the example it would be necessary to ensure that the step value is greater than zero, irrespective of the size of the array.

If we have an array with 15 elements containing the values from 0 to 14 (corresponding to the subscripts) and we need to find the element with key value 11 then the execution sequence for the routine will be as follows:

```
1ØØ I=Ø: IS=4
11Ø I=4 since A(4) < 11
11Ø I=8 since A(8) < 11
11Ø
12Ø I=9 since A(8) < 11
12Ø I=1Ø since A(9) < 11
12Ø I=11 since A(1Ø) < 11
12Ø "Key 11 found"
14Ø
```

5.2.3 Binary Tree Search

We have already looked at the structure of a binary tree, and how a binary tree can be traversed if the appropriate pointers are present for each element. For searching only the normal left and right pointers are required, but we shall assume that the negative pointers useful for traversing are

continued on page 20

LETTERS TO THE EDITOR

continued from page 15

5. Poke 16548,233 and poke 16549,66.

6. LIST the combined program.

The above routine assumes that the second program has line numbers higher than those of the first program. If not, a simple renumber routine can be used. In that case, it is necessary to load the second program, renumber it, save it back to cassette, then load the first program and proceed as above.

C. Russell Eurich
150 Sunrise Lane
Pottstown, PA 19464

Inserting Diskettes

In a recent issue, you refer to how important it is to close the drive door correctly in order to minimize disk thrashing. Could you describe the correct step-by-step procedure that insures that the disk will do just that — correctly lock into place?

Andy Chakires
1704 Miramar St.
Los Angeles, CA 90026

It's tough to describe this as a procedure, since there are so many different makes of disk drives that are all slightly different. The problem that occurs is that the diskette is not properly grasped by the plastic knob that grips the hub and holds it in place while it spins. If is slightly off center, the hub can be bent or frayed, and of course the diskette will not be read correctly.

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. ■

continued from page 18

present in the array. The same definitions as for the sequential traversal will be assumed. The code for a binary tree search could be as follows:

```
100 NXT = ROOT
110 IF NXT <= 0 PRINT "Key"; AR; "not found": GOTO 140
120 IF A(NXT,KEY) < AR THEN NXT = A(NXT,LEFT): GOTO 110
    ELSE IF A(NXT,KEY) > AR THEN NXT = A(NXT,RIGHT): GOTO 110
130 PRINT "Key"; AR; "found at location"; NXT
140 etc.
```

Again the loop could be reduced to a single line to the detriment of the legibility of the code. It can be done simply by appending line 120 to the end of line 110 with the word ELSE as separator.

Depending on the shape of the tree the binary tree search can be as fast as the binary search or as slow as the sequential search. If the length of any branch belonging to a particular node differs from any other branch belonging to the same node by no more than one, then the tree is said to be balanced. For balanced trees the search time will be approximately the same as for the binary search. The other extreme is obtained if the tree only consists of a single branch with a length equal to the number of elements in the tree. This would correspond to a linear linked list, and the search time would be equivalent to the time for a sequential search. To avoid the worst case example the tree can be reorganized so that it will be balanced and faster search times will be obtained. The reorganization requires processing time, and corresponds to the sorting of an array.

5.2.4 Direct Reference

The fastest search method possible is obtained when the array can be built so that the key value directly corresponds to the subscript of the corresponding array element. The search is then a direct reference to the particular array element required. If not all elements are used, then a value must be stored in the array to indicate if the particular element is used or not.

For example a company has about 200 employees, each of which is assigned a three digit number between 100 and 999. An array could be built containing about 200 elements, but a more efficient search could be obtained by using an array of 900 elements, corresponding to the employee numbers from 100 to 999. To find the corresponding array element the value 100 would be subtracted from the number, yielding a subscript in the range of the array. The array perhaps contains the weekly wage for each employee, and this value could be set to -1 if the corresponding employee number was unused. Admittedly a great deal of memory space would be wasted, but if speed is of more importance then memory space then the method can be recommended.

In the above example the array utilization is only about 22%, but in many cases where normal arrays are used the percentage would be much larger. There is no reason to have large amounts of unused memory in the computer if it could be used to advantage to speed up array searches. The example showed an obvious key transformation, but in other cases the transformation may not be quite as obvious. The only requirement is that the transformation yields a unique subscript value within the range of the array, in other words two distinct key values may not yield the same subscript value. Many identification numbers are obtained by appending a check digit to the end of a value within a certain range. The employee number in the above example could have been a four digit number, with the last digit being a check digit. In this case the first three digits in each number are still unique, and if the check digit has been verified by other methods then the first three digits could still be used to obtain a direct subscript value.

For very large arrays the method can be an order of magnitude faster than the binary search, besides having the advantage that the array does not have to be sorted beforehand. Thus if additions to the array or deletions are common, then the method becomes even faster compared to the binary search or the binary tree search. It is my impression that the technique is not used as often as it could be, despite the obvious advantages to be gained by using it, but perhaps as memory becomes cheaper and microcomputer addressing ranges expand the method will gain in acceptance.

5.2.5 Hash Code Calculation

A method of array organization which is closely related to the direct lookup is the hash code calculation. The difference is that in the direct lookup the key can be transformed to a unique subscript, whereas the hash code calculation will not yield a unique value. It can be used where the range of key values is large compared to the number of positions available in the array. It is used extensively in compilers for storing a table of names defined in the program.

The calculation required to convert the key value to a subscript value will vary from one application to the next, but it should be chosen so that the subscript values are reasonably evenly spaced throughout the available range. A common method is taking the remainder after division by a value, often a prime number, corresponding to the number of elements in the array. To avoid having too many keys yielding the same subscript value the array should be defined with more elements than actually needed. No fixed rules can be given, but arrays with more than 70 to 80% utilization will often give extended search times.

When two different key values yield the same subscript in the hash code calculation, a clash results, and a method must be found to distinguish the two keys and to provide storage for both the key values. The key itself is stored in the array, so that the actual value can be checked with the desired value, resulting in more space being required than for the direct lookup where the key value does not have to be stored. To resolve clashes the key can be passed through a new transformation to yield a new subscript value, or a separate portion of the array can be reserved and searched sequentially to find those elements which have resulted in clashes. If many clashes are expected, then each element can contain a link pointing to the next

continued on page 22

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

ARRAY OF HOPE FOR BASIC PROGRAMMERS
continued from page 20

element with the same hash code value.

In order to use hash code calculations the key need not be a numeric value, string values can be converted to a numeric value by performing a transformation on the ASCII values of each or some of the characters in the string. The length of the string can also be used in the calculation. The TRSDOS operating system contains several examples of hash code calculations. One of them is used to transform the file name of disk files to a one-byte value stored in one of the disk directory sectors known appropriately as the Hash code Index Table (or HIT) sector. The value is used to speed up the directory search for a particular file. If the calculated hash code is not found in the HIT sector then the file does not exist on that diskette, if it is found then the corresponding sector in the File Directory Entry (or FDE) sectors can be read directly. If several files on the diskette yield the same hash code then the HIT sector and FDE sector reads will have to be repeated until the appropriate file is found. The file name itself is stored in the FDE sector so that the value can be verified since clashes are not uncommon.

Another example of hash code calculation, which in this case is not used for table lookup, is the conversion of the file passwords to two-byte integer values stored in the directory. Since two bytes can contain 65536 distinct values the probability of a clash is much lower than for the HIT codes which can only contain 256 distinct values.

For the HIT codes the ASCII values of each byte of the file name is used. A sequence of Exclusive OR and rotate commands are used, a sequence which would be difficult to reconstruct in Basic. However a similar sequence can be used, but instead of rotating the value it can be multiplied by two before the next ASCII value is added, and if a certain value is exceeded it is subtracted from the calculated value to avoid overflow. Also the length of the string can be used as the start value for the hash code. The following code can be used to yield a value between 0 and 1023 for a string A$ of any length.

```
100 HC = LEN(A$): IF HC = 0 THEN 150
110 FOR I = 1 TO LEN(A$)
120 HC = HC + HC + ASC ( MID$ (A$, I, 1))
130 IF HC > 1023 THEN HC = HC - 1024: GOTO 130
140 NEXT I
150 etc.
```

Any value other than 1023 could obviously have been chosen to yield a hash code value within any desired range. Note that the value of HC in line 130 can be larger than 1023*2 so that 1023 may have to be subtracted more than once. For short arrays the above calculation requires more time than a sequential search, but for large arrays, especially dynamic arrays where the time required for inserting a new element in sorted sequence may be longer than the search time, the calculation time will be insignificant.

In the following example the elements with clashing hash code values are placed in a separate potion of the array, and either searched sequentially or through a linked list from the main part of the array. The array will consist of two dimensions with 1200 rows and two or three columns. The columns will contain the key, the data portion, and optionally a pointer to the overflow portion of the array. The largest calculated hash code will be 1023, so the elements from 1024 to 1199 will be used for overflow due to hash code clashes. The array has been initialized so that unused elements contain a key value of -1, and for the linked version, the unused links will also contain the value -1.

Since the overflow portion of the array can only contain a limited number of elements it may also overflow. If and when this occurs the hash code calculation and the array size should be revised to reduce the number of clashes. The search part of the routine can be coded as follows, first for the unlinked version:

```
10 HM = 1024: SM = 1200: REM Hash code and subscript
   maximum values + 1
   ...
100 IF A(HC, KEY) = AR
    PRINT "Key"; AR; "found at location"; HC:
    GOTO 160: REM Found at hash code position
110 IF A(HC, KEY) = -1 THEN 150: REM Key not found
120 HC = HM
130 IF A(HC, KEY) = AR
    PRINT "Key"; AR; "found in overflow location"; HC:
    GOTO 160: REM Found in overflow portion
140 IF A(HC, KEY) >= 0 THEN HC=HC+1: IF HC < SM THEN 130
150 PRINT "Key"; AR; "not found"
160 etc.
```

Note that if the required key is found at the expected position then only a single comparison is required. Thus if there are no clashes only a single comparison is required, irrespective of whether the value is found in the array or not. Thus for speed the hash code calculation method can compare with the direct lookup. If clashes do result then the calculation of the average number of comparisons is complicated, but it can be done if the number of clashes is known, together with the number of elements present in the overflow portion of the array.

The routine when a link is used is very similar to the above, but the number of comparisons will be reduced if a relatively large number of elements are present in the overflow portion.

```
100 IF A(HC, KEY) = AR
    PRINT "Key"; AR; "found at location"; HC:
    GOTO 130: REM Found at expected location
110 HC = A(HC, LINK): IF HC >= 0 THEN 100:
    REM Check at link location
120 PRINT "Key"; AR; "not found"
130 etc.
```

As can be seen the code for this case is simpler than that for the unlinked version, and the loop portion of the code could easily be written as a single line to improve the efficiency slightly. The average number of comparisons is again dependent on the number of clashes, and the number of elements in each link, but it should be very close to one except for very exceptional cases. The disadvantage compared to the unlinked version is, of course, the extra memory required to contain the links in the array.

continued on page 39

Hardware Spool!

CUE is a communications processor which rapidly accepts from your computer data to be printed and then goes off-line to drive the printer —independent of your computer, which is now free. CUE is better than software spooling since none of your computer's memory, processing power, or disk drives are tied up. CUE is available in parallel, serial, or with both in 2- and 4-port models (connect several printers or computers) with from 16K to 32 K memory. Use as an interface.

Prices begin at $299.

MicroCompatible

(404) 447-4805

3810 Oakcliff Industrial Court, Doraville, GA 30340

COLOR COMPUTER CORNER

Joseph Rosenman

This Month: 6809 Assembly Language and Galax Attack

How does the Color Computer do it? We all know that there is a microprocessor inside the Color Computer, known as the 6809. The 6809 processor is part of a family of microprocessor, manufactured by Motorola. The first in this family is the 6800 (used in the PET computer). The 6809 is an advanced version of the 6800 (they are both 8 bit CPUs). The newest in the family is the 68000 (used in the Radio Shack Model 16). The 68000 is a 16 bit processor, and is an extremely powerful processor.

OK, the Color Computer uses a 6809 processor. How can we learn about it? We can begin with a book on the subject. This month, I will review a book on 6809 Assembly Language. Next month, I will review the Radio Shack Color Computer EDTASM program. Assembly Language/Machine Language is the only way to really understand the inner workings of any computer. Many of the readers of the Color Computer column know of the Assembly Language for Beginners column that I have been writing in Computronics. The first four issues contain a great deal of information useful in learning any Assembly Language (the column is focused on the Z-80 microprocessor used in the Radio Shack Models 1, 2, and 3). If you have a serious desire to learn about assembly language, you might want to review these issues. These issues deal with concepts relevant to any assembly language.

6809 Assembly Language Programming

by Lance Leventhal
Osborne/McGraw Hill, 1981

There have been several books on different microcomputer assembly language programs published by Osborne, all with a reputation for thoroughness and accuracy. The 6809 assembly language book continues their tradition. This large paperback book include 22 chapters detailing the history and use of the 6809. Appendices detail the machine code. 6800 compatability and differences are described. Of course, numerous examples of the use of the machine instructions are provided throughout the book.

All Osborne books use a double intensity print system to aid readers. Important or main points are "highlighted" darker), so that the reader could skim through the book and pick out the main details. The normal print contains the explanations and examples. This system permits a rapid review of material. My impression of this book is that it is of the usual high standards I have come to expect from Osborne. Who is the book targeted for? Not the absolute beginner. Although the book begins with introductory material, I would not recommend it to the novice. If you have some experience in assembly language (that is, if you know something about hexadecimal math or have started to learn another assembly language), this book should prove excellent. As far as the highly trained are concerned, the book is useful. Having worked on several assembly languages in the past, I found that this book required a lot of reading to get to the facts. I would have preferred an approach that included a description of the processor architecture, followed by a table of assembler mnemonics and functions. I must stress, though, that most readers will find the book's detail to be a tremendous asset. If you intend to work with an 6809 assembler, I highly recommend this book.

GALAX ATTACK

from: Spectral Associates
141 Harvard Ave.
Tacoma, WA 98466
206-565-4607

System requirements: 16K or 32K system (non-extended Basic).
Supplied on: Cassette tape. Joysticks required.
Disk compatible: Yes.
Price: $21.95

Spectral Associates has come out with another high quality arcade game. Galax Attack is another variant of the legendary "Space Invaders." In this game there are three different ship types. Points gained from a hit are doubled if the alien ship was attacking when fired upon. The graphics are interesting and appropriate. The game maintains a fast pace, and has suitable sound effects. Your ship is controlled by the right joystick (missiles are fired by depressing the joystick button).

Problems? Just two. I find that the joystick is not very responsive. In addition, the missiles dropped by the alien ships are not very distinctive (they resemble the background stars). I suppose one could say that the missiles are "camouflaged." In any case, with the exception of these two deficiencies, the game is quite good. Needless to say, the game is in machine language and uses high resolution graphics.

Have you noticed the amount of Color Computer software around? I am happy to say that it has increased dramatically. Radio Shack has continued a slow but steady increase in Color Computer software. Far more important, however, is the increase in independent software and hardware sources. When I think back to the software available just one year ago, I find the changes quite satisfying. Now, a Color Computer user has a large choice of software. What about the Color Computer itself? Radio Shack has offered virtually no additions to the Color Computer. There is still a limit on the maximum memory size (32K). Mass storage is, well, not very mass (by today's standards, the Color Computer Floppy Disk is small potatoes). Radio Shack made its usual unfortunate choice with the Color Computer lower case. There are several lower

continued on page 39

PONY EXPRESS™

Finally

The most complete **mailing program** ever designed, used for many years by large corporations and organizations

Now Available for the Trs 80* Models I and III

Following are some of the features of

PONY EXPRESS

- ★ Unlimited amount of files
- ★ 1400 records on a Model III 40 track diskette
- ★ Fast machine code sort
- ★ Records can be sorted by name, zip code, zip and name, zip and address
- ★ Records can be printed on labels one to five across 132 character wide paper 80 character wide paper or with cheshire format
- ★ Print your own return address labels or any other one to five line message
- ★ Records can be selected by an unlimited amount of selection codes.
- ★ Automatic repeat routines for ease of entering records
- ★ Codes can be used instead of titles for greater efficiency during input
- ★ Unique correction and deletion routines for ease of updating files
- ★ Self-documenting and extensive documentation so that **anyone** can learn how to use **PONY EXPRESS** in less than an hour

Special Introductory Offer

$99.00

Call or Write to

COMPUTECH

975 Forest Avenue
Lakewood, New Jersey 08701
(201) 364-3005

Master Charge and Visa accepted,
add $2.50 Shipping and Handling.
N.J. residents add 5% sales tax.

*Trs 80 is a trademark of Tandy Corp.

BYTEWRITER
DAISY WHEEL PRINTER

***NEW** / NOT REFURBISHED*

LETTER QUALITY PRINTER AND TYPEWRITER IN ONE PACKAGE

ONLY

$795

plus shipping

The BYTEWRITER is a new Olivetti Praxis electronic typewriter with a micro-processor controlled driver added internally. No software driver needed.

Maybe we goofed by not charging more for a DAISY WHEEL PRINTER

What's wrong with it?

We guess everyone must be getting used to paying over $2000 for a new Daisy Wheel Printer and over $1500 for a refurbished Daisy Wheel Printer. Anything that costs less must be junk. Right?

WRONG!

The BYTEWRITER is not only cheaper it is better!

Following are some of the features of

BYTEWRITER

- ★ 10, 12, or 15 characters per inch switch selectable
- ★ Interchangeable daisy wheel - many different typestyles readily available
- ★ Correctable Electronic Typewriter operation with nothing to disconnect
- ★ Correctable film or nylon cartridge ribbon
- ★ Self test program built in
- ★ Only 14 internal moving parts for incredible reliability, ease, efficiency and accuracy
  Two keyboard positions for standard American type or special characters for foreign languages

Centronics compatible parallel input operates with **TRS-80, APPLE, IBM** and many others

Call or Write to

COMPUTECH

975 Forest Avenue
Lakewood, New Jersey 08701
(201) 364-3005

Master Charge and Visa accepted,
N.J. residents add 5% sales tax.

H&E COMPUTRONICS INC.

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

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

FROM

RACET COMPUTES LTD. . . . . .

- 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

*** 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

HSDS HARD DISK DRIVE SOFTWARE

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

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

FOR DEALER INFORMATION CALL: RACET COMPUTES

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

MODEL II SPEEDUP—FAST DISK I/O

THIS IS AN ENHANCEMENT FOR TRSDOS 2.0 THAT WILL RADICALLY DECREASE DISK ACCESS TIME.

- DISKS BOOT FASTER TO DOS
- IMPROVE DISK I/O UNDER BASIC

NEW

MODEL II ONLY $99.95

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

NEW

MODEL II ONLY $75.00

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:

COMPUTRONICS INC
MATHEMATICAL APPLICATIONS, INC.

VISA AMERICAN EXPRESS

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

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

24 HOUR ORDER LINE
(914) 425-1535

DEALER INQUIRIES WELCOME

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

PRACTICAL BUSINESS PROGRAMS

Steven M. Zimmerman, Ph.D. and Leo M. Conrad

THE GRAPHICAL METHOD OF LINEAR PROGRAMMING FOR SOLVING SIMPLE BUSINESS PROBLEMS

The Graphical method of linear programming is both a solution procedure for simple problems and a demonstration method to aid in the understanding of this type of problem. This program draws both line and area graphs associated with two dimensional problems. An adaption of the Simplex method, published in the September 1982 issue of this magazine, is then used to produce an analytical solution to the program.

The graphical method of linear programming allows the business person to make a decision concerning two variables. These variables may be relative to any decision the business person must make. For example, the number of first class salespersons versus the number of trainees to be assigned to a particular office, or the amount of a given product to stock versus the amount of a second product to stock in a retail store, or the dollars to budget for one research project versus a second project.

If you have a problem involving more than two decisions, then the Simplex method must be used. The reason the graphical method is limited to two decisions is that we can only plot in two dimensions. The graphical method is primarly a visualization method and secondly a solution procedure.

A Sample Problem

Assume you are the owner operator of the local Computer For Sale Store. You must decide the number of IBM Personal Computers and the number of Apple II computers you will stock. You know the contribution margin (difference between the selling price and the purchase price per unit) on the IBM PC is $1500 for a typical unit sold, while the contribution margin for the Apple II is $1600.

You are limited by the display and storage space you have available to stock only ten computers. In other words, the number of IBM computers stocked plus the number of Apple computer stocked must be less than or equal to ten.

In addition, you have a budget problem. Your combined credit and cash allows you to stock up to $12,000 worth of goods as measured by their purchase price. The cost of a typical IBM unit is $3,000 while the price of a typical Apple II computer is $2,500.

You are working under one more constraint: that you must stock a minimum of three IBM computers in order to keep the franchise. You assume that you will be able to sell all of the units you must stock in a given time period.

Setting up Our problem in terms of Equations

The objective of the problem is to maximize the sum of the contribution margins. We write the following equation:

```
Contribution times IBM's  + Contribution times Apples
Margin                      Margin

$1,500 I + $1,600 A
```

This equation is called the objective function.

The constraints are first stated in words and then as equations:

Number of IBM's plus Number of Apples less than or equal to room

```
I + A <= 10
```

Cost times IBM's Plus cost times Apples is less than or equal to budget.

```
$3,000 * I + $2,500 * A <= $12,000
```

The last constraint is that you must stock at least three IBM PC's:

```
1 * I + 0 * A >= 3
```

Using the Program

The program may solve both maximization problems, like the example, and minimization problems. In the case of maximization problems, all constraints must be less than or equal to, and in the case of minimization, all constraints must be greater than or equal to. In our sample problem all constraints fit the pattern except the last one.

In order to make the last constraint relative to stocking a minimum number of IBM PC's, we multiply both sides of the equation by -1 to turn the sign around. The constraints now look like the following:

```
-3  I - 0 B <= -3
```

The equation looks strange, but it will work, and that is what counts.

We are now ready to run the program. The first thing seen on the screen is:

```
GRAPHICAL METHOD OF LINEAR PROGRAMMING
DEVELOPED BY:
STEVEN M. ZIMMERMAN,PH.D.
LEO M. CONRAD   1982
OBJECTIVE FUNCTION A * X1 + B * X2 ...INPUT A & B?
```

X1 is used for the first variable, in our case the number of Apple II computers, and X2 is used for the second variable, the number of IBM PC's. The value of A is the contribution margin for IBM computers, $1,500, and the value for B is $1,600 the contribution margin for Apple

continued on page 30

MULTIDOS Model III Now Available!

WHAT DO YOU USE YOUR TRS-80® For?

★ Word Processing ?

MULTIDOS will permit you to 'grab' a formatted diskette, shove it into a track compatible drive, and write to it — no matter if it is single density, double density, or DBLDOS™ density. This is possible because of **MULTIDOS'** automatic density recognition which includes diskettes formatted by DBLDOS™ — similar to NEWDOS/80 2.0.

★ Basic Programming ?

"MULTIDOS' BASIC is unequaled, in both features and size, among TRS-80® Disk Basics.
— Paul Wiener 1982

Included is B BASIC/CMD which has all of the "BOSS" single step and trace features. Now you can insert 'break points' to turn on/off the trace and/or single stepping in your **BASIC** program. Now you can save the screen as formatted and look at variable values. Afterward, return the screen as before and continue your **BASIC** program.

★ Whatever — MULTIDOS Is Easy To Learn, Easy To Use

DOS commands may be repeated, even multiple DOS commands. Systems utilities never require more than 2 keystrokes per query.

Now That's Easy! To Make It Easier There is a 'Help File'!

MULTIDOS
Model I or Model III
Only $79.95
(Watch the Competition Squirm!)

MULTIDOS Now Available From
WARLOCK WAREHOUSE
1691 Eason, Pontiac, MI 48054
(313) 673-2224 or (313) 673-8700
— AND —
POWERSOFT
11500 Stemmons Fwy., Suite 125
Dallas, Texas 75229
(214) 484-9428

Cosmopolitan Electronics Corporation
P.O. BOX 234 • PLYMOUTH, MI 48170
(313) 397-3126
— Dealer Inquiries Welcomed —

COD - Cash or Certified Check
Allow 2 Weeks For
Personal Checks To Clear
ADD $3.00
Shipping & Handling
Michigan Residents
Include 4% Sales Tax

WHY $ IN DI$K . . . WE'VE PUT SENSE IN ÇASSETTE

RELIABLE LOW-COST HIGH-SPEED TAPING

NO-FUSS HIGH SPEED SOFTWARE

KWICOS (Mod 1, 4k to 48k) **$26**
KWIK Cassette Operating System for Mod 1. The easy-to-use Level II enhancement for reliable fast taping (select 1000-3000 baud). Features: save, load, verify, search, chain-load, catalog, and test-read of both BASIC and machine-code programs . . . plus: long pgm names, passwords, debounce, slow 'list', self 'backup', and more.

KOS3 (Mod 3, 16k to 48k) **$26**
The KWIK Cassette Operating System for Model 3. All 'KWICOS' features at 2200 baud, plus KWIK set of: clock display, Time, Date, Cassette high/low, I/O routing, etc.

KWIKIT (specify Model) **$12**
Mini-system for BASIC programs only. EasyLoad 1000 baud for Mod 1, 2200 baud for Mod 3. Many 'KWICOS' features.

KWINK (Model 1, 4k-48k) **$15**
Makes stand-alone fast-loading (2x-6x) copies of any standard 500 baud "SYSTEM" program. (At 6x, 3 minute program loads in 44 sec!)

KLOAD (Model 1) **$15**
Similar to 'KWINK', but for BASIC pgms only. (Specify 16-32-48k)

KLOAN (Mod 3, 16k-48k) **$12**
Makes 500 or 1500 baud copy of any other standard 'system' pgm.

KWIK Software
Box 328
Bolivar, MO 65613
Phone (417) 326-7154

NO-FUSS ANY-SPEED HARDWARE

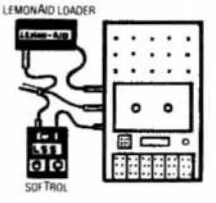
The amazing LemonAid Loader takes the 'finickies' out of Model I loads, but even more amazing . . . make KWIK copies of your Model I programs and load them flawlessly at 6× speed! Instant installation . . . simply plug between your cable and recorder. No internal batteries or external power needed.

New Model LLQ-2 Deluxe loader has volume controlled jack for tape listening without plug-pulling, using external speaker/earphone. Ideal for Model I, compatible with Model III. ppd $21.99.

New Model LQ-2 loader. Built-in low level audio monitor with fixed volume. Model I, Mod III compatible. $18.99

Model LL-1. Great little economy loader for the Model I. $14.99

Model LSS. Solid state SOFTROL (1) eliminates switch-hits, (2) Push button and slide switch control of CTR motor makes tape positioning a snap without plug pulling. (3) Cushioned motor-off delay pulls end of programs past CTR pinch rollers . . . no pinch-hits, plus (4) automatically gaps between saved programs. For all standard-plug recorders and computers. $18.99



LEMONS TECH
325 N. Hwy 65
P.O. Drawer 429
Buffalo, MO 65622
(417) 345 7643

Call either number 'til 10 pm most any day for in-door order. U.S. orders postpaid, add $3 for COD or overseas. Missouri residents add sales tax.

continued from page 28

computers. We input 1500,1600 as the answer to the above question.

Next on the screen will be:

```
FOR MAXIMIZATION ALL CONSTRAINTS MUST BE A*X1 + B*A2 <=Z
FOR MINIMIZATION ALL CONSTRAINTS MUST BE A*X1 + B*X2 >=Z
THE NON-NEGATIVITY CONSTRAINT FOR ALL VARIABLES ASSUMED
1 MAXIMIZATION, OR 2 MINIMIZATION ?
```

We have not spoken of the built-in constraint that all variables must result in answers which are greater than equal to 0. In business problems, this is a common limitation, and we have selected a solution procedure with this limitation built in.

Since our objective is to maximize profit, we type 1 and hit ENTER.

Next on the screen is:

```
NUMBER OF CONSTRAINTS ?
```

There are three constraints. One due to room, one due to budget, and one due to contract comment. We type 3 and hit the ENTER key.

The next question is:

```
COEFFICIENTS OF X1 AND X2 <= OR >= Z ?
```

The values of the three constraints are:

Equation #	Coefficient of X1 IBM's	Coefficient of X2 Apples	Constant
1	1	1	1Ø
2	3ØØØ	25ØØ	12ØØØ
3	-1	Ø	-3

The above question will appear three times. We answer with the numbers in the table above. When complete, we see this on the screen:

```
UPPER LIMIT OF X1 ?
```

The program is designed to handle a large variety of problems. You the user must select the scale which gives you the best picture for the problem under study. We have designed the program to recycle, so as to allow the user to try different scales to obtain the best picture for the purpose of the analysis. In this case we selected 10 as the upper value of X1, because this is where the line for the maximum number of computers to be stocked crosses the Apple axis. In a similar manner we selected 10 as the answer for the next question.

```
UPPER LIMIT OF X2 ?
```

The next question is relative to the frequency of plotting the graphs. If you select too small a number, the graph will be skimpy; if you slect too large a number, the amount of time needed to draw the picture will be very long. Start with the specification of 100 for this variable and then try different values to see which approach fits your needs the best.

```
INCREMENT (1ØØ+) ?
```

Next on the screen will be:

```
PRINTER (Y/N) ?
```

We have designed our printer copying routines for printers without graphic abilities. The screen results are much better than you can get from such printers. The results are of value and useful for reports if care is taken to select the scales to fit the abilities of the printers. We will assume that you have a printer, that it is running, and that you wish to have printer output of the results. In actual use we recommend that you spend some time in getting the scales set on the screen before using the printer to copy results.

The next question is:

```
TITLE & DATE ?
```

We make it a practice to identify and date all output which is sent through the printer. Put in any title and date you wish. The date should be put in the form of MO/DA/YR.

The next question is:

```
COPY LENGTH (3,6) ?
```

We have two copy routines in the program. Depending on the detail needed, you may prefer one to the other. Try both just to see what happens. We used 3 in our sample run.

Next on the screen will be:

```
AREA OR LINE PICTURE ?
```

Two graphing routines are available. One plots the constraints as a line, and the other whites out the area where the constraint restricts the solution from occuring. We have found both pictures useful at different times. It is possible to end up with a screen which has been completely whited out when selecting the AREA option. If this happens, we say the solution is degenerate, or there is no feasible solution to the problem.

We typed AREA at this time in our sample run and watched the picture being drawn. We almost had a unfeasible problem with the constraints we had defined in our sample problem. As you watch the picture being drawn, note how the first constraint is completly dominated by the second constraint. The amount of room available to stock computers is more than it needed.

After the picture has been drawn and a copy made by your printer, you are told to hit ENTER to continue. Do so and you will see on the screen:

```
ACTION MENU
    G     NEW GRAPH
    R     RECYCLE
    S     SIMPLEX SOLUTION
SELECTION ?
```

As noted, you are now able to redraw your picture again and again in order to select the best picture for the problem under study. You may also recycle or use the built-in Simplex procedure for solving the problem analytically. We selected this option, and now we will examine

the final simplex tableau which results.

The tableau looks like the following:

```
SOLUTION OPTIMAL
          X1        X2        S1        S2        S3
S1         Ø.ØØ      Ø.ØØ      1.ØØ     -Ø.ØØ      Ø.ØØ    5.8Ø
X1         1.ØØ      1.2Ø      Ø.ØØ     -1.ØØ      Ø.ØØ    3.ØØ
X2         Ø.ØØ      1.ØØ      Ø.ØØ      Ø.ØØ      1.2Ø    1.2Ø
        Ø.ØØ      Ø.ØØ      Ø.ØØ      Ø.64   42Ø.ØØ   %642Ø.ØØ
X1            = 3
X2            =1.2
IF AN X DOES NOT APPEAR IT IS ZERO
VALUE OF OBJECTIVE FUNCTION  642Ø
```

In the case of a maximization problem, the solution is found by looking down the first column for X1 and X2. In this case both are found. This means that both computers should be stocked. The number of IBM computers which maximize the contribution margin is 3.0. The number of Apple computers is 1.2. This means three IBM computers should be stocked and either one or two Apples. In mathematical problems contraints are fixed. In real life constraints are approximate. The two alternate decisions may be examined and the question reviewed.

Warning—Nothing's Perfect

The program's analytical solution is based on a Simplex algorithm, which is a step-by-step procedure for solving this problem. The Simplex algorithm is subject to a condition called "degenerate." When this condition exists, the algorithm will recycle forever or yield a false solution, not taking one or more of the constraints into account. Sometimes you can get around this condtion and obtain a useful result by putting the variables into the computer in a different order, or changing the order of the constraints. The solution will still be degenerate, but you will be able to obtain a workable result.

The degenerate condition is not the same as an impossible set of constraints. Degeneracy may exist even when a solution is possible. If the constraints are put in such that no solution is possible, the algorithm will also give unusable results. The graphical picture will tell you what is happening in this case.

```
1Ø CLEAR 5ØØ:CLS:REM      "LPGRAPH
2Ø CLS: PRINT "GRAPHICAL METHOD OF LINEAR PROGRAMMING":
PRINT "Copyright ": PRINT "STEVEN M. ZIMMERMAN,PH.D. ":I=Ø
3Ø PRINT "LEO M. CONRAD    1982"
4Ø INPUT "OBJECTIVE FUNCTION A * X1 + B* X2 ... INPUT A &
B";A,B
5Ø GOTO 62Ø
6Ø INPUT "NUMBER OF CONSTRAINTS";NE:C%=NE: FOR I=1 TO NE:
INPUT "COEFFICIENTS OF X1 AND X2 <= OR >= Z";A(I),B(I),C(I):
NEXT
7Ø INPUT "UPPER LIMIT OF X1";TX:SX=Ø
8Ø INPUT "UPPER LIMIT OF X2";TY:SY=Ø
9Ø INPUT "INCREMENT (1ØØ+-)";JJ: INPUT "PRINTER (Y/N)";P$:
IF P$="Y" THEN INPUT "TITLE & DATE";T$,D$: INPUT "COPY LENGTH
(3,6)";CL
1ØØ INPUT "AREA OR LINE PICTURE";AL$
11Ø CLS: FOR I=Ø TO 43: SET(22,I):NEXT
12Ø FOR I=22 TO 125: SET(I,43):NEXT
```

```
13Ø I=7: FOR J=64 TO 832 STEP 128:I=I-1: PRINT @J,
(SY+I*(TY-SY)/6);: NEXT
14Ø I=-1: FOR J=97Ø TO 1Ø15 STEP 9:I=I+1: PRINT @J,
(SX+I*(TX-SX)/5);: NEXT
15Ø GOTO 18Ø
16Ø PRINT @22,"PRESS ENTER TO CONTINUE";
17Ø Q$=INKEY$: IF Q$="" THEN 16Ø  ELSE 68Ø
18Ø S=(TX-SX)/JJ
19Ø IF AL$="AREA" THEN 25Ø
2ØØ FOR I=1 TO NE:HH=1: FOR X=SX TO TX STEP S:Y=999999:
IF B(I)<>Ø THEN Y=-A(I)/B(I)*X+C(I)/B(I) :REM THIS IS THE
CONSTRAINT EQUATION
21Ø E=23+((X-SX)/(TX-SX))*93:F=4Ø-((Y-SY)/(TY-SY))*37:
IF E>5 AND E<126 THEN IF F>Ø AND F<43 SET(E,F)
22Ø IF E<5 OR E>126 OR F<Ø OR F>43 PRINT @61,"***";:
FOR L=1 TO 3:R=RND(1ØØ): NEXT L: PRINT @61,"   ";
23Ø IF A(I)<>Ø AND HH=1 THEN H=INT(X): IF B(I)=Ø AND
H=INT(C(I)/A(I)) THEN FOR H=Ø TO 42: SET(E,H): NEXT H:HH=Ø
24Ø NEXT X,I: GOTO 49Ø
25Ø IF M%=2 THEN 37Ø
26Ø FOR I=1 TO NE:HH=1: FOR X=SX TO TX STEP S:Y=999999:
IF B(I)<>Ø THEN Y=-A(I)/B(I)*X+C(I)/B(I) :REM THIS IS THE
CONSTRAINT EQUATION
27Ø E=23+((X-SX)/(TX-SX))*93:F=4Ø-((Y-SY)/(TY-SY))*37:
IF E>5 AND E<126 AND F>Ø AND F<43 AND B(I)>=Ø AND C(I)>Ø
SET(E,F): FOR RX=F TO Ø STEP -1: SET(E,RX): NEXT : GOTO 36Ø
28Ø IF Y<Ø AND B(I)<>Ø AND C(I)>Ø : FOR RX=43 TO Ø STEP -1:
SET(E,RX): NEXT : GOTO 36Ø
29Ø IF E>5 AND E<126 AND F>Ø AND F<43 AND B(I)<>Ø AND C(I)<Ø
SET(E,F): FOR RX=F TO 43 : SET(E,RX): NEXT : GOTO 36Ø
3ØØ IF B(I)=Ø AND A(I)<>Ø AND X>(C(I)/A(I)) AND C(I)<Ø THEN
PRINT @61,"***";: FOR L=1 TO 3:R=RND(1ØØ): NEXT : PRINT @61,
"   ";: GOTO 36Ø
31Ø IF B(I)=Ø AND A(I)<>Ø AND X<(C(I)/A(I)) AND C(I)>Ø THEN
PRINT @61,"***";: FOR L=1 TO 3: R=RND(1ØØ): NEXT : PRINT @61,
"   ";: GOTO 36Ø
32Ø IF Y>Ø AND Y<TY THEN IF E<5 OR E>126 OR F<Ø OR F>43 THEN
FOR RX=42 TO Ø STEP-1: SET(E,RX): NEXT: GOTO 36Ø
33Ø IF A(I)<>Ø AND HH=1 THEN H=INT(X): IF B(I)=Ø AND
H=INT(C(I)/A(I)) THEN FOR H=Ø TO 42: SET(E,H): NEXT H:HH=Ø
34Ø IF Y<TY OR B(I)=Ø THEN IF Y>Ø AND A(I)<>Ø THEN FOR RX=Ø
TO 43: SET(E,RX): NEXT: GOTO 36Ø
35Ø IF C(I)<Ø AND Y>Ø AND A(I)<>Ø THEN FOR RX=Ø TO 43:
SET(E,RX): NEXT
36Ø NEXT X,I: GOTO 49Ø
37Ø REM FOR MINIMIZATION AREA PLOTTING
38Ø FOR I=1 TO NE:HH=1: FOR X=SX TO TX STEP S:Y=999999:
IF B(I)<>Ø THEN Y=-A(I)/B(I)*X+C(I)/B(I) :REM THIS IS THE
CONSTRAINT EQUATION
39Ø E=23+((X-SX)/(TX-SX))*93: F=4Ø-((Y-SY)/(TY-SY))*37:
IF E>5 AND E<126 AND F>Ø AND F<43 AND B(I)>Ø AND C(I)>Ø
SET(E,F): FOR RX=F TO 43 : SET(E,RX): NEXT: GOTO 48Ø
4ØØ IF E>5 AND E<126 AND F>Ø AND F<43 AND B(I)<>Ø AND C(I)<Ø
THEN SET(E,F): FOR RX=F TO Ø STEP -1 : SET(E,RX): NEXT:
GOTO 48Ø
41Ø IF B(I)=Ø AND A(I)<>Ø AND X<(C(I)/A(I)) AND C(I)>Ø THEN
FOR RX=42 TO Ø STEP -1: SET(E,RX): NEXT: GOTO 48Ø
42Ø IF B(I)=Ø AND A(I)<>Ø AND X>(C(I)/A(I)) AND C(I)<Ø THEN
FOR RX=42 TO Ø STEP -1: SET(E,RX): NEXT: GOTO 48Ø
43Ø IF Y>Ø AND Y<TY THEN IF E<5 OR E>126 OR F<Ø OR F>43 THEN
PRINT @61,"***";: FOR L=1 TO 3:R=RND(1ØØ): NEXT L: PRINT @61,
"   ";: GOTO 48Ø
44Ø IF Y>Ø AND  B(I)>Ø AND C(I)>Ø : FOR RX=43 TO Ø STEP -1:
SET(E,RX): NEXT: GOTO 48Ø
45Ø IF Y<TY AND Y>Ø AND B(I)<Ø AND C(I)<Ø : FOR RX=43 TO Ø
STEP -1: SET(E,RX): NEXT: GOTO 48Ø
46Ø IF Y<TY AND A(I)<Ø AND B(I)<Ø AND C(I)<Ø : FOR RX=43 TO Ø
STEP -1: SET(E,RX): NEXT: GOTO 48Ø
47Ø IF Y>Ø AND Y<TY AND B(I)<>Ø THEN FOR RX = Ø TO 43:
SET(E,RX): NEXT
48Ø NEXTX,I
49Ø IF P$<>"Y" THEN 16Ø
5ØØ LPRINT " "
51Ø LPRINT "TITLE:   ";T$:LPRINT "DATE:   ";D$
52Ø FF$="MAXIMIZATION": IF M%=2 THEN FF$="MINIMIZATION"
53Ø LPRINT FF$: LPRINT " ": LPRINT " THE OBJECTIVE FUNCTION
IS ";A;" * X1 + ";B;" * X2  "
54Ø LPRINT " THE CONSTRAINTS ARE ": FOR I=1 TO NE: LPRINT
"  ";A(I);" * X1 + ";B(I);" * X2 ";FZ$;" ";C(I): NEXT I:
SP=2: IF CL=6 THEN SP=1
55Ø E$="###,###.###": FOR I=3 TO 43 STEP SP: J=42-I:
IF I<43 THEN LPRINT USING E$;(SY+J*(TY-SY)/39);:
56Ø FOR X=Ø TO 1ØØ STEP 1.6: REM CHANGE 1ØØ TO 127 ON LINE
PRINTER II
57Ø IF POINT(X,I) THEN LPRINT "*";ELSE LPRINT " ";
58Ø NEXT X: LPRINT " ": NEXT I
59Ø FOR I=Ø TO 6:G(I+1)=SX+I*(TX-SX)/5: NEXT
6ØØ E$="            ###,###.###  ###,###.###  ###,###.###
 ###,###.###  ###,###.###  ###,###.###": LPRINT USING E$;
G(1),G(2),G(3),G(4),G(5): GOTO 16Ø  :REM ADD G(6) FOR LINE
PRINTER I
61Ø PRINT "SIMPLEX SOLUTIONS TO LINEAR PROGRAMMING PROBLEMS"
62Ø PRINT "FOR MAXIMIZATION ALL CONSTRAINTS ASSUMED TO BE
A*X1 + B*X2  <=Z"
63Ø PRINT "FOR MINIMIZATION ALL CONSTRAINTS ASSUMED TO BE
A*X1 + B*X2  >=Z"
64Ø PRINT "THE NON-NEGATIVITY CONSTRAINTS FOR ALL VARIABLES
ASSUMED"
65Ø INPUT " 1 MAXIMIZATION OR 2 MINIMIZATION ";M%: IF M%<1 OR
M%>2 THEN 65Ø
66Ø FZ$=">=": IF M%=1 THEN FZ$="<="
67Ø GOTO 6Ø
68Ø CLS: PRINT "ACTION MENU": PRINT "     G     NEW GRAPH":
PRINT "     R     RECYCLE": PRINT "     S     SIMPLEX
SOLUTION"
69Ø INPUT "SELECTION ";SS$: CLS: IF SS$="S" THEN 73Ø
7ØØ IF SS$="R" THEN 2Ø
71Ø IF SS$="G" THEN 7Ø
72Ø GOTO 68Ø
73Ø V%=2
74Ø R%=C%+1
75Ø CC%=V%+C%+1
76Ø IF M%=2 THEN R%=V%+1
77Ø X1(1)=A:X1(2)=B
78Ø IF M%=2 THEN 86Ø
79Ø FOR J=1 TO V%
8ØØ X(C%+1,J)=-X1(J)
81Ø NEXT
82Ø FOR J=V%+1 TO V%+C%+1
83Ø X(C%+1,J)=Ø
84Ø NEXT
85Ø GOTO 9ØØ
86Ø FOR I=1 TO V%
87Ø X(I,V%+C%+1)=X1(I)
```

```
880 NEXT I
890 X(V%+1,V%+C%+1)=0
900 FOR I=1 TO C%
910 X1(1)=A(I):X1(2)=B(I):X1(3)=C(I)
920 IF M%=1 THEN 980
930 FOR K=1 TO V%
940 X(K,I)=X1(K)
950 NEXT K
960 X(V%+1,I)=-X1(V%+1)
970 GOTO 1020
980 FOR K=1 TO V%
990 X(I,K)=X1(K)
1000 NEXT K
1010 X(I,CC%)=X1(V%+1)
1020 NEXT I
1030 FOR I=1 TO R%-1
1040 IF M%=2 THEN 1090
1050 FOR K=V%+1 TO CC%-1
1060 IF I=K-V% THEN LETX(I,K)=1
1070 NEXT K
1080 GOTO 1120
1090 FOR K=C%+1 TO CC%-1
1100 IF I=K-C% THEN LET X(I,K)=1
1110 NEXT K
1120 NEXT I
1130 IF P$="Y" THEN P$="H" ELSE P$="C"
1140 IF M%=1 THEN 1250
1150 FOR J=1 TO C%
1160 CL$(J)="S"+CHR$(J+48)
1170 NEXT J
1180 FOR J=C%+1 TO C%+V%+1
1190 CL$(J)="X"+CHR$(J-C%+48)
1200 NEXT
1210 FOR I=1 TO R%-1
1220 RL$(I)="X"+CHR$(I+48)
1230 NEXT
1240 GOTO 1580
1250 FOR J=1 TO V%
1260 CL$(J)="X"+CHR$(J+48)
1270 NEXT
1280 FOR J=V%+1 TO CC%-1
1290 CL$(J)="S"+CHR$(J-V%+48)
1300 NEXT J
1310 FOR I=1 TO R%-1
1320 RL$(I)="S"+CHR$(I+48)
1330 NEXT I
1340 GOTO 1580
1350 CL$(0)="  ":RL$(R%)="  "
1360 ZP$=" ###.## "
1370 AP$=" %    % "
1380 FOR J=0 TO CC%-1
1390 PRINT USING AP$;CL$(J);: IF P$="H" LPRINT USING AP$;
CL$(J);
1400 NEXT: PRINT: IF P$="H" THEN LPRINT " "
1410 FOR I=1 TO R%:REM  PRINT RESULTS SUBROUTINE
1420 FOR J=1 TO CC%
1430 IF J=1 THEN PRINT USING AP$;RL$(I);: IF P$="H" THEN
LPRINT USING AP$;RL$(I);
1440 PRINT USING ZP$;X(I,J);
1450 IF P$="H" THEN LPRINT USINGZP$; X(I,J);
1460 NEXT J
1470 PRINT
1480 IF P$="H" THEN LPRINT " "
1490 NEXT I
1500 IF M%=2 THEN 1540
1510 FOR I=1 TO R%: IF LEFT$(RL$(I),1)="X" THEN PRINT
RL$(I),"=";X(I,CC%): IF P$="H" THEN LPRINT RL$(I),"=";
X(I,CC%)
1520 NEXT: PRINT "IF AN X DOES NOT APPEAR IT IS ZERO":
IF P$="H" THEN LPRINT "IF AN X DOES NOT APPEAR IT IS ZERO"
1530 GOTO 1560
1540 FOR I=NE+1 TO CC%-1: PRINT CL$(I),"=";X(R%,I): IF P$="H"
THEN LPRINT CL$(I),"=",X(R%,I)
1550 NEXT
1560 PRINT "VALUE OF OBJECTIVE FUNCTION ";X(R%,CC%): IF
P$="H" THEN LPRINT "VALUE OF OBJECTIVE FUNCTION ";X(R%,CC%)
1570 RETURN
1580 REM START OF ANALYSIS
1590 REM   NEGATIVE INDICATOR?
1600 N=0
1610 FOR J=1 TO CC%-1
1620 IF X(R%,J)<0 THEN LET N=1:JJ=J: GOTO 1660 :REM JJ IS
PIVOTAL COLUMN
1630 NEXT J
1640 PRINT "OPTIMAL SOLUTION ": IF P$="H" THEN LPRINT
"OPTIMAL SOLUTION"
1650 GOSUB 1350 : INPUT "ENTER TO RECYCLE FROM START";DD$:
GOTO 10
1660 M=9999999:REM SELECTING ROW
1670 FOR I=1 TO R%-1
1680 IF X(I,JJ)=0 THEN X1(I)=999999999: GOTO 1700
1690 X1(I)=X(I,CC%)/X(I,JJ)
1700 IF X(I,JJ)<=0 THEN 1720
1710 IF X1(I)<M THEN M=X1(I):II=I:REM PIVOTAL ROW
1720 NEXT I
1730 PV=X(II,JJ)
1740 FOR J=1 TO CC%:REM DIVIDE ITH ROW OLD BY PIVOT
1750 X(II,J)=X(II,J)/PV
1760 NEXT
1770 RL$(II)=CL$(JJ):REM RELABEL ROW
1780 FOR I=1 TO R%
1790 IF I=II THEN NEXT I: GOTO 1580
1800 A=X(I,JJ)
1810 FOR J=1 TO CC%
1820 X(I,J)=X(I,J)-A*X(II,J)
1830 NEXT J,I
1840 GOTO 1580
```

Summary

This program solves the two-dimensional simplex-type linear programming problem both visually and analytically. The graphics allow the user to see the area where a feasible solution exists and how it is identified. The simplex solution procedure attached to the graphic analysis yields an algebraic solution.

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 ■

Disk Drives and Cases

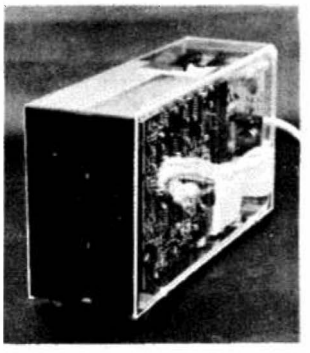
The B.T. Enterprises Mini-Floppy 5-1/4 inch Disk Drive Case is unique in that it is constructed of clear plexiglass. This innovative design is the first on the market which enables the user to actually see the Disk Drives working while using the computer. The case is constructed of a 1/4 inch durable 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

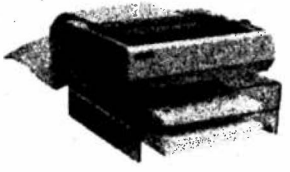


Bare Drives

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

108 Carlough Rd.
Bohemia, N.Y. 11716
(516) 567-8166 (voice)
(516) 588-8538 (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 the 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 $80.00 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

10B Carlough Rd.
Bohemia, N.Y. 11716
(516) 567-8155 (voice)
(516) 588-5836 (modem)
Dealer Inquiries Welcome
N.Y.S. Residents Add Tax

master charge
VISA
Welcome

Radio Shack is a trademark of Tandy Corp.

Toll Free Order Line (800) 645-1165

Prices subject to change

POCKET COMPUTER CORNER

Steven M. Zimmerman, Ph. D. and Leo M. Conrad

Ratios for the Real Estate Investor

The selection of ratios produced by our program are very popular among real estate investors. Included in the selection are:

1. Gross Rent Multiplier
2. Capitalization Ratio
3. Cash on Cash, return on investment (equity), ROI
4. Debt coverage ratio

We will detail how each of the ratios are calculated and some of the thinking processes used in applying them to investments. As with all tools, it is up to the investor to decide how best to use them for himself in his particular situation.

RATIOS CALCULATION METHODS

The Gross Rent Multiplier is the ratio of the investment divided by the gross rent income. This is a very useful fast method to tell if an investment is earning enough to justify a given price. In times of consistent financing and predictable business conditions, the gross rent multiplier can be a useful first cut tool. In times of innovative financing and unstable business conditions, the gross rent multiplier can be difficult to use.

Cap rates, or capitalization rates have been popular for both real estate investment and appraisal. The assumption when using cap rates is the investment (total investment including mortgages) should earn some value for many years, as if it were money placed in a bank forever. The calculation method consists of dividing the net yearly income (Gross income less expenses) by the total investment. No mortgage payments are taken into consideration.

CASH on CASH, return on investment, return on equity, or ROI may be calculated by dividing the cash flow per year, (Gross less expenses, less mortgage payments) by the cash investment. This method assumes a steady yearly cash earnings which is never true if after tax income is being considered.

Debt Coverage is simply the ratio of the mortgage to the overall investment. It tells the investor how much investor capital is needed as a ratio of the overall the investment. The lender will often specify the maximum coverage percentage which is acceptable from the lender's point of view.

CALCULATION OF THE FACTORS OF A MORTGAGE

To make the task of using the program a little easier, we have included the ability to calulate any one of the four factors of a mortgage. The method used to calculate each factor is detailed in Table 1 (see next column).

The only method that needs some explanation is that for finding interest per month. This task is usually a gross search task or an approximation task such as we have developed here. There is no direct method available.

Loan - principal	P=R*((1+i)^N-1)/(i*(1+i)^N)
Payment	R=P*i*(1+i)^N/((1+i)^N-1)
Months	N=LN (1/(1-(i*P/R))/LN (1+i)
interest	A five step approximation-where a series of five straight lines are fitted to the equation to find interest per period.

Where i=interest per month, N=number of months, R=the payment per month, and P is the loan or principal.

Table 1: Factors of a Mortgage Calculation Method

The approach we have used consists of finding two points on the interest function, at 25%/12 and 75%/12 interest per month. A straight line is fitted to these two points, and the point where the straight line crosses the X axis is calculated. Then a small amount is added and subtracted to the value of X at this point, and the process is repeated. Each time the process is repeated the interval is reduced.

USING THE PROGRAM

The main menu of this program may be called by typing R. in either the RUN or DEFinable modes.

```
MORTG RATIOS?
```

You select the choice you want by typing the entire word as shown. The reason the MORTG option was included in this program is that, in our experience, the investor often is missing some information about the mortgage. This option allows you to solve for any missing information.

If you type MORTG and hit ENTER the next thing you will see is:

```
1-YR 2-LO 3-PAY 4-%?
```

The option in this menu must be selected by its number. 1 means you want to solve for the number of years, 2 for the value of the loan, 3 for the monthly payment, and 4 for the annual percentage rate of the loan. Assume for our example you know the payments are $180.03 per month, the loan's value is $15,000 and that there is 15 years left on mortgage. Type 4 to select the annual percentage rate option and then answer the next three questions with the data just given. The results will be:

```
PRIN: 15000.
PAY: 180.03
11.99026322%
15. YEARS
```

When the program has finished printing on the printer or on the display (followed by an ENTER after each line) you will be returned to the main menu. This time select the RATIOS option and you will be asked the following question:

```
GROSS RENT?
```

Assume the gross rent is $200 per month or $2400 per year. Type 2400 and hit ENTER. The next question will be:

```
INVESTMENT?
```

Assume the investment is $25,000 including the mortgage. Type 25000 and hit ENTER. The next question is relative to annual expenses:

```
EXPENSES?
```

In this case assume expenses are equal to $2,150 per year. Type 2150 and hit ENTER.

The next question is relative to the mortgage. Assume the mortgage is the same as reviewed above. If this is true, all you need to do is hit the ENTER key at this point. The results should be:

```
GROSS RENT MULTI
     10.41666667
CAP RATE
            0.01
CASH/CASH
       -0.191036
DEBT COVERAGE
             0.6
GROSS RENT:2400.
INVESTMENT:25000

EXPENSES:2150.
PRIN: 15000.
PAY: 180.03
11.99026322%
15. YEARS
```

You may wonder why the CASH/CASH is a negative number. If you calculate the gross income (2400) less expenses (2150) less 12 times the mortgage payment (180.03), you get -1910.36, a negative number, the ratio must therefore be a negative number.

EXAMINING THE PROGRAM

The first three lines of the program are the main menu. In line 1 the selection is made, and if MORTG is selected the program is routed to line 50. In line 2 the program is routed to line 10 if RATIOS is selected. In line 3 the program is returned to line 1 if an unacceptable selection is made.

Line 10 inputs all the information needed by the program to calculate the ratios except debt information. Line 11 is designed to input mortgage information. If you have already calculated the mortgage in the alternative routine, you need only answer with the ENTER key to skip the entire line.

Lines 12 through 15 print the ratios of interest, calculate their values, and prints the results. Line 16 prints the input information for the record. This line should be left out if you are not using a printer to aid in speeding the output process.

Lines 40 through 47 are a series of four subroutines which do all the calculations for the interest calculations. Line 40 is to find the present value of the loan. Line 42 is to find the payments. Line 43 is to find the life while lines 44 through 47 are used to find the annual percentage rate.

In line 50 the menu for the mortgage is found. Lines 51 through 54 input information available, while lines 55 through 58 call the subroutine needed.

The printing for all mortgage information is performed in line 59. This statement is used for both parts of the program. You may not wish to direct the program to this statement when calculating ratios if you are not using a printer.

PROGRAM LISTING

```
1:INPUT "MORTG RATIOS?";A$:IF A$="MORTG"THEN 50
2:IF A$="RATIOS"THEN 10
3:GOTO 1
10:INPUT "GROSS RENT?";G:INPUT "INVESTMENT?";D:INPUT
  " EXPENSES?";E
11:INPUT "MORTGAGE?";P:INPUT "LIFE?";N:N=12N:INPUT "APR?";I
   :I=.01I/12:GOSUB 41
12:PRINT "GROSS RENT MULTI":PRINT D/G
13:PRINT "CAP RATE":PRINT (G-E)/D
14:PRINT "CASH/CASH":PRINT (G-E-12R)/(D-P)
15:PRINT "DEBT COVERAGE":PRINT P/D
16:PRINT "GROSS RENT:";G:PRINT "INVESTMENT:";D:PRINT
   "EXPENSES:";E:GOTO 59
40:P=R*((1+I)^N-1)/(I*(I+1)^N):RETURN
41:R=P*I*(1+I)^N/((1+I)^N-1):RETURN
42:N=LN (1/(1-(IP/R)))/LN (1+I):RETURN
43:B=.25/12:T=.75/12:FOR U=1 TO 5:C=B
   :D=R-P*C*(1+C)^N/((1+C)^N-1)
44:E=T:F=R-P*E*(1+E)^N/((1+E)^N-1)
45:A=(D-F)/(C-E):B=D-A*C:I=-B/A:G=P*I*(1+I)^N/((1+I)^N-1)
  :T=I+.2/U^4:B=I-.2/U^4
46:IF B<0 LET B=.001
47:NEXT U:RETURN
50:INPUT "1-YR 2-LO 3-PAY 4-%?";A
51:IF A<>1 INPUT "LIFE?";N:N=12N
52:IF A<>2 INPUT "PRINT?";P
53:IF A<>3 INPUT "PAY?";R
54:IF A<>4 INPUT "APR?";I:I=.01I/12
55:IF A=3 GOSUB 41
56:IF A=2 GOSUB 40
57:IF A=1 GOSUB 42
58:IF A=4 GOSUB 43
59:PRINT "PRIN: ";P:PRINT "PAY: ";R:PRINT 1200I;"%":
   PRINT N/12;" YEARS"
60:GOTO 1
```

SUMMARY

This program is written for the investor interested in real estate ratios to aid in making investment decisions. These ratios should not be used by themselves, but in conjunction with other evaluation techniques they can be of value.

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

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

COLOR COMPUTER CORNER

continued from page 24

case modifications available which convert the Color Computer from reverse video to normal lower case (with descenders). They are not expensive. Why couldn't Radio Shack include a decent lower case on its machine? Remember, RS did the same thing with the Model 1. I can't say why. With a floppy disk and a printer, the Color Computer becomes a useful (and inexpensive) word processor. With programs like SPECTACULATOR, home finance control, inventory control, etc., the Color Computer becomes a very inexpensive yet versatile system. Unfortunately, since Radio Shack has decided (to the best of my knowledge) not to produce a full scale Expansion Unit, the potential power and usefulness of the Color Computer will remain limited. Considering the power of the 6809 processor, this is sad fact. It wouldn't surprise me if some independent company decided to produce an Expansion unit that would elevate the Color Computer to the scale of the more versatile personal computers around today. Only time will tell.

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

ARRAY OF HOPE FOR BASIC PROGRAMMERS

continued from page 22

Hash code arrays also have some disadvantages which may not be obvious from the above examples. One of these is that, as for the direct lookup, duplicate keys cannot easily be accomodated. Duplicate keys will, of course, result in a clash, and the duplicate could be stored in the overflow portion. This would necessitate a search in the overflow portion each time, and thus the method would be most appropriate with a linked array. With duplicate keys the speed advantages of hash code arrays are reduced. The second disadvantage is that the array cannot easily be accessed in key sequence. This may not be necessary or only used sparingly, and in these cases the array could be sorted when the key sequence traversal is required. In most cases the advantages probably outweigh the disadvantages, and hash code arrays, like direct lookup arrays, should probably be used more extensively than at present.

Arne Rohde
Pilevej 31
7600 Struer, Denmark ■

THE CARDWRITER

Anthony T. Scarpelli

This article is about a program that is not really needed. The program writes postcards. Most people use postcards because they are handy and inexpensive. How much effort does it take to write a postcard, you ask? Not much, but when you have a lot of them to write, this program can be used to save a lot of time. This particular program is directed to those who like to send in postcards to TV stations.

I believe we all watch TV. There is a great variety of programs available for viewing, and we all have our opinions as to their worth. They run from the best to the worst, each depending upon our own personal view. And some of us like to write to the TV station to express those views. Often it is not necessary to write a long letter, and sometimes we don't even have the time to to write such a letter. So that is where postcards come in and also this program.

What this program does is to prompt you for certain responses, and you answer with either a number or with text, whichever is required. It also allows for a four line comment. It then prints the information out onto the card along with your name and address, and finally prints out the address of the broadcasting station you are writing to. You can literally print out a multitude of cards in minutes with little effort on your part; you don't even have to look up the addresses of the TV stations, nor even your own.

Let's take a look at the program. Line 20 is our standard initializing line, and from lines 30 to 160 we set up all the string variables and arrays we will be using. Lines 30-50 are for our greeting, 60-110 are for our opinions, 120-150 are for the TV station addresses, and finally line 160 is for your own address. You'll notice here in line 160 that there is a bit of indenting on the second and third line of the address, but not on the first. This is because I TAB to the indentation during printing for the first line. This allows us to use one string variable; but many other methods can also be used if you don't like this one. You should also notice that in all the addresses there is a line feed (down arrow) between all the lines. You can put any addresses you want in these lines, and even expand on them. You might want to add to this list your local TV stations, for instance.

Line 180 asks if you need instructions in case this is your first time through or if you don't write cards very often. Between lines 200 and 490 we input all the information that will eventually be printed. Lines 210-220 are for the greeting, and lines 230-260 are for the subject. I use line feeds freely in many of my print statements, so that is why some of these lines have no line numbers. (Most of these comments I make is for those of you who are new to Basic.) Line 250 will print a string of hyphens so that when you enter a subject, you will not go over the number of characters desired for that line. A postcard will easily fit 50 characters in a line, so all printout has to be limited to that. Once you see a printout you'll know why you are limited to only 33 characters for the show title, and this seems to be enough for most shows.

Lines 270 to 290 are for the date, and you will notice that line 290 says LINE INPUT. This is a disk basic statement that allows you to input commas and quotes which are normally delimiters in input statements. If you don't have disk basic, then delete "LINE", but you won't be able to use commas in your dates. Lines 300-320 are for the time, and lines 330-350 are for your opinion. You probably have noticed that in the prompts I have included the words that will lead into your response. This is to trigger your memory. It helps to know how it will be printed.

Lines 360 to 460 are for your comment, and need more explanation. Line 380 again prints a line of hyphens to limit your first line of comment; you are allowed four lines, but only the first one is limited to less than 50 characters. Line 390 initializes our comment variable and the count, and the next line sets up a loop that scans the keyboard for a keypress. It stays in that line until a key is pressed, then falls through to the next line as soon as you enter a character. The first thing we do is to print it, then check to see if it is an (ENTER), whose ASCII code is 13. Line 430 checks to see if the keypress was a line feed (down arrow), and if it was, will print 50 hyphens, and then increment our count. If the count becomes five we leave the comment section. If our keypress was a backspace, we want to remove the last character from our comment string, and that is what the rest of line 450 does. Finally line 460 adds our character to the comment string and returns for another keypress. One problem with this routine that might be annoying to you is that there is no cursor as you type in your comment. If this bothers you, you can program one in here with a varitey of methods.

Lines 470 and 480 ask for the broadcasting company you will be sending the card to, and it allows for an extra address that hasn't beed previously stored. This is taken care of in line 490. Of course you should not use commas in your address unless you have disk Basic.

Lines 500 to 540 produce a pause to allow you to set up your printer. It also allows you to start over in case you made a mistake along the way, or want to change your opinion. Lines 560 to 770 print the card. Line 560 and 570 are straightforward, and 580 gets the length of the subject string. We need to know how long it is because it will determine how many words will be printed on a line before going to the next line. We want to fill up the card as much as possible, but we also want to limit the lines to 50 characters. So in lines 590 to 620 we use the length to determine how many words of the phrase "which was aired" will be printed along on the same line as the the subject string. We also set a flag count to be able to know later just how many words were used in the next printed line. That is, in lines 640 to 670, we have a similar process to fill up the line depending on what was printed before. All this is to make the card appear to have been typed by a human, and not a machine.

Lines 670 to 740 do the standard job of printing out the rest of the card. Line 750 and 760 allow you to turn the card over to print the address. And finally lines 780 and 790 allow you to start another card.

As is usual, even though the program is simple, a lot of work went into it to make a simple, time-consuming process, like cardwriting, into a simple, speeded up process. I hope that you can use it. I like it better than writing cards, myself. Now, if everyone used this faster process, maybe TV will change for the better because there will be more people writing in to let the station masters know that we care what is going on in the land of television.

```
1Ø REM -- TV RESPONSE CARD WRITER
2Ø CLEAR 5ØØ:DIM GR$(3),OP$(6),AD$(5)
3Ø GR$(1)="SIR:"
4Ø GR$(2)="SIRS:"
5Ø GR$(3)="MADAM:"
6Ø OP$(1)="EXCELLENT"
7Ø OP$(2)="QUITE GOOD"
8Ø OP$(3)="GOOD"
9Ø OP$(4)="ONLY FAIR"
1ØØ OP$(5)="PRETTY BAD"
11Ø OP$(6)="TERRIBLE"
12Ø AD$(1)="ABC-TV
133Ø AVE. OF THE AMERICAS
NEW YORK  NY  1ØØ19"
13Ø AD$(2)="NBC-TV
3Ø ROCKEFELLER CENTER
NEW YORK  NY  1ØØ2Ø"
14Ø AD$(3)="CBS-TV
524 W. 57TH ST.
NEW YORK  NY  1ØØ19"
15Ø AD$(4)="PUBLIC BROADCASTING SERVICE
475 L-ENSANT PLAZA, SW
WASHINGTON  DC  2ØØ24"
16Ø MA$="ANTHONY T. SCARPELLI
                98 FOXCROFT DR.
                SCARBOROUGH  ME  Ø4Ø74"
17Ø CLS:PRINT"** UNIVERSAL CARD WRITER **"
18Ø PRINT:INPUT"DO YOU WANT INSTRUCTIONS (Y/N)";X$
19Ø IF X$="Y" THEN CLS:GOSUB 81Ø
2ØØ PRINT:PRINT"ENTER YOUR UNIVERSAL GREETING:"
21Ø PRINT"DEAR...
  1--SIR:
  2--SIRS:
  3--MADAM:"
22Ø INPUT GR
23Ø PRINT:PRINT"ENTER THE SUBJECT:"
24Ø PRINT"YOUR SHOW..."
25Ø PRINT"  ";STRING$(33,"-")
26Ø INPUT SB$
27Ø PRINT:PRINT"ENTER THE DATE:"
28Ø PRINT"ON..."
29Ø LINE INPUT DT$
3ØØ PRINT:PRINT"ENTER THE TIME:"
31Ø PRINT"AT..."
```

continued on page 50

Assembly Language for Beginners (Part 9)

Joseph Rosenman

Since I KNOW that you dazzled all of your friends (not to mention yourselves) with last month's alphabet printing program, I will proceed with a description of different ways that the same task could be performed.

```
;
; Alphabet printing program 2 (reading from a table).
;
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.
;
PROG2   CALL    CLR     ;Clear the screen.
        LD      HL,TAB  ;Get the address of the letter table.
        LD      B,26    ;Number of letters in the alphabet.
;
LOOP    LD      A,(HL)  ;Get the current letter.
        CALL    DISP    ;Display it.
        INC     HL      ;Point to the next letter.
        DEC     B       ;One less letter to display.
;                       ;also, set the status flags.
        JR      NZ,LOOP ;If not zero, display next letter.
;
        JP      DOS
;
; Data area
;
TAB     DEFM    'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
        END     PROG2
;
7000  CD C9 01 21 13 70 06 1A 7E CD 33 00 23 05 20 F8
7010  C3 2D 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D
7020  4E 4F 50 51 52 53 54 55 56 57 58 59 5A .. .. ..

7000  ..!.......3.#. ..-@ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

As you can see, there is also a second "dump" starting at address 7000H, except that the second dump is in ASCII. Note that the table of the alphabet also appears intact in the dump. Are there any new instructions? No! Since I can't explain any new instructions, let's take a closer look at the memory image.

```
7000  CD C9 01    CALL 01C9H
7003  21 13 70    LD   HL,7013H
7006  06 1A       LD   B,1AH
7008  7E          LD   A,(HL)
7009  CD 33 00    CALL 0033H
700C  23          INC  HL
700D  05          DEC  B
700E  20 F8       JR   NZ,F9H (-7)
7010  C3 2D 40    JP   402DH

7013  41 42 43 44 'ABCD'
7017  45 46 47 48 'EFGH'
701B  49 4A 4B 4C 'IJKL'
701F  4D 4E 4F 50 'MNOP'
7023  51 52 53 54 'QRST'
7027  55 56 57 58 'UVWX'
702B  59 5A       'YZ'
```

Now why would a CALL 01C9H turn into a CD C9 01? Of course, the CD is the "CALL" portion. The number CDH is the machine code for a CALL instruction. This is always followed by the two byte address (there for, a CALL instruction is always three bytes long). Why does the address 01C9H turn into C9 01? The reason is all sixteen bit (2 byte) values are stored "backwards." The two bytes in a sixteen bit number are identified as the MSB for Most Significant Byte, and LSB for Least Significant Byte. In the address 01C9H the MSB is 01, and the LSB is C9H. Since two byte values are stored in reversed order (LSB then MSB), the number is stored as C9 01. Is this also true for other sixteen bit values? Yes. Note that:

```
the  LD  HL,7013  becomes:  23 13 70
and  CALL  0033   becomes:  CD 33 00.
```

Now let's learn about a new instruction: DJNZ.

```
;
; Alphabet printing program 3 (reading from a table).
;
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.
;
PROG3   CALL    CLR     ;Clear the screen.
        LD      HL,TAB  ;Get the address of the letter table.
        LD      B,26    ;Number of letters in the alphabet.
;
LOOP    LD      A,(HL)  ;Get the current letter.
        CALL    DISP    ;Display it.
        INC     HL      ;Point to the next letter.
        DJNZ    LOOP    ;DEC B, if not zero - repeat.
        JP      DOS
;
; Data area
;
TAB     DEFM    'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
        END     PROG3
;
7000  CD C9 01 21 12 70 06 1A 7E CD 33 00 23 10 F9 C3
7010  2D 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E
7020  4F 50 51 52 53 54 55 56 57 58 59 5A .. .. .. ..
```

The DJNZ is a special instruction, which exists only in the Z-80 instruction set (it isn't in the 8080 or 8085 instruction set). The DJNZ stands for "Decrement B and Jump if not zero." The Jump portion of this instruction is just like the JR (127 bytes back or 128 bytes forward). What is more, it only decrements the B register. If B

contains a zero AFTER the decrement, the jump falls through (to the next instruction). Otherwise, it jumps to the target address.

One new instruction deserves another! Greet the LDIR (Load Direct on Incrementing Register), PUSH, and POP. What will the program do? Fill the screen with all of the characters from 20H to BFH (the values from 80H to BFH are the TRS-80 graphics characters). The program should briefly pause in between each "screen fill."

```
;
; Screen display of character set
;
        ORG     5200H
VIDEO   EQU     3C00H   ;Start of Video RAM.
DELVAL  EQU     0500H   ;Delay value
DOS     EQU     402DH   ;DOS re-entry location.
;
PROG4   LD      A,20H       ;First character to save.
PROG4A  LD      HL,VIDEO    ;First location in Video.
        LD      DE,VIDEO+1  ;Second location in Video.
        LD      BC,400H     ;Size of Video RAM
        LD      (HL),A      ;Save character in Video.
        LDIR                ;Fill rest of Video.
        CP      0BFH        ;Last character?
        JP      Z,DOS       ;If done, exit.
        INC     A           ;Next character.
        CALL    DELAY       ;Brief Pause
        JP      PROG4A      ;Repeat with next character.
;
DELAY   PUSH    AF          ;Save A register.
        LD      HL,DELVAL   ;Get delay value.
DEL1    DEC     HL          ;One less to count.
        LD      A,H         ;Get MSB of count.
        OR      L           ;Mash the MSB and LSB together.
        JR      NZ,DEL1     ;If both aren't 0, repeat.
        POP     AF          ;Done, get A back.
        RET
;
        END     PROG4
```

```
5200  3E 20 21 00 3C 11 01 3C 01 00 04 77 ED B0 FE BF
5210  CA 2D 40 3C CD 1A 52 C3 02 52 F5 21 02 05 2B 7C
5220  B5 20 FB F1 C9 .. .. .. .. .. .. .. .. .. .. ..
```

Gasp and dismay! I'm supposed to understand THAT!? You bet. First, lets trace through two complete iterations (excluding the DELAY subroutine). These two "passes" will fill the screen with blanks, then with exclamations marks.

Program Trace

```
 1 PC=5200 A=?? BC=???? DE=???? HL=???? SP=7100  LD  A,20H
 2 PC=5202 A=20 BC=???? DE=???? HL=???? SP=7100  LD  HL,3C00H
 3 PC=5205 A=20 BC=???? DE=???? HL=3C00 SP=7100  LD  DE,3C01H
 4 PC=5208 A=20 BC=???? DE=3C01 HL=3C00 SP=7100  LD  BC,400H
 5 PC=520B A=20 BC=0400 DE=3C01 HL=3C00 SP=7100  LD  (HL),A
 6 PC=520C A=20 BC=0400 DE=3C01 HL=3C00 SP=7100  LDIR
 7 PC=520E A=20 BC=0000 DE=4000 HL=3FFF SP=7100  CP  0BFH
 8 PC=5210 A=20 BC=0000 DE=4000 HL=3FFF SP=7100  JR  Z,04
 9 PC=5212 A=20 BC=0000 DE=4000 HL=3FFF SP=7100  INC  A
10 PC=5213 A=21 BC=0000 DE=4000 HL=3FFF SP=7100  CALL 5225
                <Delay routine>
11 PC=5216 A=21 BC=0000 DE=4000 HL=3FFF SP=7100  JP  5202
12 PC=5202 A=21 BC=0000 DE=4000 HL=3FFF SP=7100  LD  HL,3C00H
13 PC=5205 A=21 BC=0000 DE=4000 HL=3C00 SP=7100  LD  DE,3C01H
14 PC=5208 A=21 BC=0000 DE=3C01 HL=3C00 SP=7100  LD  BC,400H
15 PC=520B A=21 BC=0400 DE=3C01 HL=3C00 SP=7100  LD  (HL),A
16 PC=520C A=21 BC=0400 DE=3C01 HL=3C00 SP=7100  LDIR
17 PC=520E A=21 BC=0000 DE=4000 HL=3FFF SP=7100  CP  0BFH
18 PC=5210 A=21 BC=0000 DE=4000 HL=3FFF SP=7100  JR  Z,04
19 PC=5212 A=21 BC=0000 DE=4000 HL=3FFF SP=7100  INC  A
20 PC=5213 A=22 BC=0000 DE=4000 HL=3FFF SP=7100  CALL 5225
                <Delay routine>
21 PC=5216 A=22 BC=0000 DE=4000 HL=3FFF SP=7100  JP  5202
22 PC=5202 A=22 BC=0000 DE=4000 HL=3FFF SP=7100  LD  HL,3C00H
                And so on...
```

continued on next page

The numbers to the left are only for reference purposes. What follows is a description of what each statement is doing. Since this program uses the PC, SP, BC, DE, HL, and A registers, the trace includes each register. In addition, the assembly language instruction is shown on the right.

1) Get the first character to display (a blank) in A.
2) Get the first video location in HL (16 bit value).
3) Get the second video location in DE.
4) Get the size of VIDEO RAM in BC.
5) Put the character into the start of Video Ram (contents of HL)
6) Fill Video Ram with the Character.

What? One instruction can do that? Yes. The LDIR instruction uses the HL, DE, and BC registers all at once. Remember, LDIR stands for Load on Incrementing Register. How does it work? HL contains the Source address, DE contains the Destination address, and BC contains the count. The instruction causes the byte pointed to by HL to be moved into the address pointed to by DE. Then, HL and DE are incremented, and BC is decremented. If BC does not equal zero, the process is repeated. When BC equals 0, execution continues with the next instruction. So, lets say we wanted to copy 20 bytes from 6000H to 7000H. The action of LDIR would be:

```
6000  30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
6010  40 41 42 43 .. .. .. .. .. .. .. .. .. .. .. ..

7000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7010  00 00 00 00 .. .. .. .. .. .. .. .. .. .. .. ..

 HL   DE   BC
6000 7000 0014
6001 7001 0013
6002 7002 0012
6003 7003 0011
6004 7004 0010          This table contains the
6005 7005 000F          contents of the 3 registers
6006 7006 000E          involved in the LDIR instruction.
6007 7007 000D          The changes shown illustrate
6008 7008 000C          what happens during the execution
6009 7009 000B          of a SINGLE LDIR instruction.
600A 700A 000A
600B 700B 0009
600C 700C 0008
600D 700D 0007
600E 700E 0006
600F 700F 0005
6010 7010 0004
6011 7011 0003
6012 7012 0002
6013 7013 0001
6014 7014 0000

6000  30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
6010  40 41 42 43 .. .. .. .. .. .. .. .. .. .. .. ..

7000  30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
7010  40 41 42 43 .. .. .. .. .. .. .. .. .. .. .. ..
```

Now what would happen if we loaded HL with 7000H, DE with 7001H, and BC with 14H? Well; LDIR would move from 7000H to 7001H, then from 7001H to 7002H, then from 7002H to 7003H, etc. So, memory would then contain:

```
7000  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
7010  30 30 30 30 .. .. .. .. .. .. .. .. .. .. .. ..

 HL   DE   BC
7000 7001 0014
7001 7002 0013
7002 7003 0012
7003 7004 0011
7004 7005 0010
etc.
```

When used in this way, the LDIR can quickly "fill" a memory area with a specific byte. Whatever byte is in the address pointed to by HL at the start of execution will be copied into all of the addresses following (up to DE+BC). Pretty slick, isn't it? Anyway, back to our program trace.

7) Is the value in A the last character? (If so, set the zero flag).
8) If we are up to the last character, skip ahead to DONE.
9) Since we're not done, increment A so that it points to the next character.
10) Call the delay routine so that we can see what's on the screen.
11) Let's do it all again.
12) Get the first Video location.
13) And the second video location.
14) Count (size of video).
15) Save the current character in the first location.
16) Fill-er-up!
17) Are we done yet? (No).

And so on.

Now, it's time to examine the Delay routine. There are two new instructions in this subroutine: PUSH and POP. Both PUSH and POP make use of the SP register. SP stands for "Stack Pointer." It looks like it's time to discuss the mysterious "Stack." A stack is a special data structure used in computers. (A Data Structure is a "system" of dealing with information that is formally described, and highly consistent.) A stack is actually a table or list of variables. Generally, you can access (read to or write from) any variable in this list (also known as an array). In the case of a stack, you can only access one end of the list. Whenever you add something to the stack, it "Pushes" everything down. Since the first thing added to the stack is always at the bottom of the list, it is the last thing to be accessed when things are taken off of the stack. This data structure is also known as a LIFO — an acronym for "Last In/First Out." Adding something to a stack is known as "PUSHing," and taking something off of a stack is known as "POPing." All values that are PUSHed or POPed must be sixteen bits long. Any double register can be PUSHed or POPed (AF, BC, DE, HL, IX, IY) except the SP and the PC. Since the SP

contains the current address of the stack, it would make no sense to save the address of the stack IN the stack. In the case of the PC, it is pushed onto the stack every time a CALL instruction is executed, and poped off of the stack whenever a RET instruction is executed.

Where is the stack? It could be anywhere in free RAM. I haven't included the necessary instructions to create and restore stacks yet so that I could keep the programs as simple as possible. Generally, the DOS or BASIC will have reserved a Stack with at least 10 free bytes (enough for 5 PUSHES and/or CALLs). Does that mean a program with more then 5 CALLS can't run? Not necessarily. If the CALLs are all one right after the other (without any RETs in between), then it DOES mean that the program may not work. In the case of the above programs, the value of the SP is actually fictitious. Since we didn't actually set the value of the SP (and can't know what it is), I set it for an area of unused RAM (where we might very well have placed it).

I'm afraid that the description of the DELAY program will have to wait until next month. In the mean time, you might want to experiment with different delay values. Try a value like 5000H instead of 500H. For those daring souls, I will pose an exercise. Re-write the program so that it displays all of the characters in reverse order (starting with the graphic block BFH, and ending with a blank). The only hint I will offer is that you should use the LDDR instruction (Load on Decrementing Register). It works just like LDIR, except that HL and DE are decremented after each move. Good Luck!

Corrections

One final order of business. I just received my copy of Computronics with Assembly Language # 6. There are several mistakes in it that need correction (some my fault, some not). First, an excuse. I have been writing in 8080 assembly language recently, and I see that I confused Z-80 and 8080 mnemonics several times (the Z-80 is a superset of the 8080 machine language). On page 54, I used the command MNEMONIC JNZ. This is the 8080 equivalent of the Z-80:

```
Label   Command Argument
        JP      NZ,Address.
```

Also on this page, there is the mysterious LDPPP instruction! Don't ask me what it does, since I have no idea. It was supposed to be a LD. (Maybe it stands for "Load Direct on Positive Phase Processing?!") Anyway, there is no such instruction on the 8080, 8085, Z-80, or any other CPU I know of. Apologies to all concerned.

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

PHONE BILL SORTING PROGRAM

Dan Keen and Dave Dischert

Some people are involved in many different things. Recently we met a man who owned seven small companies which he operated out of one office.

At the end of the month when the telephone bill arrived, each long distance call had to be checked to see which company that amount should be billed to. Each number had to be put in a separate column and then each list was totalled.

A log book could be kept, but this was a pain, and it would still require someone to go through it manually and tally all amounts for each company.

Since most of the long distance calls were to the same numbers each month, we created a program which allows him to set up company files and store the numbers that are frequently called by each company.

When the monthly bill comes in, the secretary simply goes down the list and enters each phone number and its date, time (number of minutes), and amount.

She can then review the list in the computer and check it for any typographical errors.

When it is satisfactory, the computer will sort the entries, printing out a list for each company, including the date, time, phone number, amount and grand total for that company.

A miscellaneous list is created and totalled to show all the numbers which appeared on the phone bill but which were not on file with any of the companies.

OVERVIEW

This program allows you to store all regularly called phone numbers into separate files, each file representing a particular business or use. Several files can be set up for different businesses, or perhaps two files can be created, one to represent business calls, the other for personal calls made from the same telephone.

HOW TO USE THE PROGRAM

The top of the screen will show all the current company files on the diskette which you have created. Below that the MAIN MENU will be displayed. The video display would look like this assuming there have been two files created so far, one for a campground business and the other for the owner's personal calls.

```
THE FOLLOWING FILES ARE ON THIS DISK:
-------------------------------------------------------
1 CAMP
2 PERSONAL

<A>DD NUMBERS                    <C> ADD COMPANY LIST
<S>EARCH/EDIT/DELETE FILE        <E>NTER PHONE BILL
<R>E RUN PHONE BILL              <K>ILL OLD BILL
```

An invisible option, <Q>uit is not displayed, but will respond if a "Q" is typed. This will cause the program to end and return you to BASIC READY.

<C> ADD COMPANY LIST

This is the very first option you must choose when you initially run the program. You are prompted with:
ENTER COMPANY NAME (OR JUST HIT <ENTER> KEY TO QUIT)

If the name conflicts with one previously entered, it will indicate that there is a conflict and the program will not accept your input.

It will again prompt you with ENTER COMPANY, and you can keep going. When you are finished, simply hit the <ENTER> key in response to entering a name. The program allows up to 9 companies to be added. They do not have to be added all at once. This option can be selected any time you wish to add another company.

The computer will automatically take the company name you type in and make it equal a filename which is in the correct form that the computer requires to enable it to store that file on diskette. There must be eight characters. If you enter more than eight, only the first eight (starting from the left) will be used. If you enter less than eight, the remainder will have asterisks tacked onto it. An extension of /FIL representing "file" is concatinated. Do not use numbers or spaces when entering a company name. For example, let's say you typed in a company named REPAIR. The filename on the diskette will appear as REPAIR**/FIL.

<A>DD NUMBERS

All the company files currently in storage (created by using the <C> ADD COMPANY option) are displayed on the screen along with a number beside them (1, 2, 3, etc.). You are prompted with ENTER COMPANY NUMBER TO WORK ON. Type the corresponding number. As with many inputs in this program, only one keystroke is necessary (you don't have to hit <ENTER> after you type the number). This helps speed you through the program.

ENTER PHONE NUMBER (HIT <ENTER> TO QUIT). Type in a phone number. If you type in the area code here, then you will have to also type in the area codes when you enter each phone number on your monthly bill, or the computer will not find a match. So get in the habit of using area codes either all of the time, or none of the time. You could choose just to use area codes on numbers which do not have the same code as you.

You can continue to enter numbers. To quit, simply respond to the ENTER NUMBER prompt by hitting the <ENTER> key.

Enter phone numbers in the form of 609-886-1511, using hypens between number sections.

<S>EARCH/EDIT/DELETE FILES

Selecting this option results in another menu, EDIT MENU. Options are:

<D>ELETE A COMPANY NAME
<P>HONE NUMBER FILE (EDIT OR CHANGE INFORMATION IN A FILE)
<Q>UIT

If you choose the <D>ELETE: The company file names and consecutive numbers are displayed. ENTER COMPANY NUMBER YOU WANT TO DELETE OR <Q>UIT. In case you want to bail out and not erase anything, you can always hit the "Q". This whole program is loaded with these "Q"uit bail outs at critical points. Finally, you are quizzed ARE YOU SURE YOU WANT TO DELETE? (Y/N). Enter a Y for yes, N for no. Once erased, there is no way to recover any information in that company file.

If you choose the <P>HONE NUMBER EDIT: ENTER COMPANY NUMBER YOU WANT TO WORK ON. A list is displayed.

The top of the screen will show you instructions as to available selections at this time:
SELECT LETTER: <A>DVANCE, <B>ACKUP, <E>DIT, <D>ELETE, <Q>UIT

Picture in your mind a long list of phone numbers. The TV screen is a window, allowing you to view one number at a time. To look ahead to the next number, hit the letter "A". To go back to the last number, "B". If you wish to delete the number currently displayed, "D". If you want to change the number, hit "E".

<E>NTER PHONE BILL

ENTER DATE, PHONE NUMBER, TIME, COST. Do not enter a dollar sign for COST, simply the amount. Just go down the entire phone bill, entering the information for each number. Entering Area Code is optional. When you are done, just hit <ENTER> when asked DATE.

NOW TO REVIEW THE LIST. DO YOU WANT TO VIEW LIST ON:
<P>RINTER
<V>IDEO
<S>KIP TO SORTING BILL (Note item number to edit if needed.)

At this time you may wish to look over the numbers you just typed in to see if you made any errors. If you choose P or V, the list is shown, then the screen displays:

HIT <ENTER> TO SORT, <Q>UIT, <E>DIT FILE

If all looks well, then just hit <ENTER> and the computer will begin to sort out the phone numbers to their respective companies. Choosing the <S>KIP TO SORT option as previously mentioned also brings you to this point.

If you choose <E>DIT, ENTER FILE NUMBER OF LINE TO EDIT OR ENTER A ZERO TO QUIT will be displayed. The four lines of information (that is, date, number, time, amount) are shown. Choose a number from 1 to 4 representing the line you wish to change. The old value will be shown and you will be asked to enter the new value.

SORT. The sort routine will now print out on the line printer each company name and all the phone numbers and costs associated with that company. If there are any numbers on the list you entered that are not on file, then they will be placed in under a MISCELLANEOUS heading and their cost totalled.

If there is more than one company file that contains the same phone number, the sorted list will bill that phone number to the first company it encounters. For example, if company file number 1 and number 3 both contain the number 123-1234, then number 1 will get the bill attributed to company 1.

A phone bill file is created, saving all these numbers and information you just entered. You can get another printout any time you wish by selecting the <R>ERUN PHONE BILL OPTION.

Before you <E>NTER another phone bill, you MUST use the <K>ILL OLD BILL option. This will erase the bill and clear the way for you to enter a new one. If you fail to do this, it is possible for phone numbers in last month's bill to be included erroneously into this month's bill. That would happen if last month's phone list was longer than this month's. So be sure to Kill before entering a new phone bill.

The <K>ILL option will prompt you with ARE YOU SURE? (Y/N). This gives you a chance to bail out at the last minute.

PROGRAM LISTING

```
10 REM
                MODEL I & III VERSION
                PHONE BILL PROGRAM
                CREATED 08/06/81
                UPDATED 11/08/81

20 CLS : CLEAR 15000 : DIM CO$(50),F1$(250),PN$(250),MO$(12):
M$="$$####.##" : M1$="######" : M2$="###" : GOSUB 1000 :
GOSUB 1020
30 CLS : CLOSE : GOSUB 1030 : A$(0)="<A>DD NUMBERS" :
A$(1)="<C> ADD COMPANY LIST" : A$(2)="<S>EARCH/EDIT/DELETE
FILE" : A$(3)="<E>NTER PHONE BILL" : A$(4)="<R>E RUN PHONE
BILL" : A$(5)="<K>ILL OLD BILL" : C=0
40 FOR B=704 TO 959 STEP 32 : PRINT @B,A$(C); : C=C+1: NEXT :
C=0 : A=672 : IF D$( 1)="" THEN GOSUB 5100
50 A$=INKEY$: IF A$="A" THEN 100 ELSE IF A$="C" THEN 200 ELSE
IF A$="S" THEN 300 ELSE IF A$="E" THEN 500 ELSE IF A$="R"
THEN 600 ELSE IF A$="K" THEN 1300 ELSE IF A$="Q" THEN CLOSE :
END
60 A=A+32 : PRINT @A,STRING$(25," "); : FOR X=1 TO 50 : NEXT
: PRINT @A,A$(C); : C=C+1 : FOR X=1 TO 50 : NEXT : IF A$ <>""
THEN TM=0
70 IFC=6 THEN A=672 : C=0
80 TM=TM+1 : IF TM=500 THEN GOSUB850 : GOSUB 1030
90 GOTO 50
100 REM
                ADD NUMBERS

110 CLS : PRINT TAB(23)"ADD PHONE NUMBERS" : PRINT
STRING$(64,"-") : GOSUB 1030 : GOSUB 1040
120 PH$="" : LINEINPUT"ENTER PHONE NUMBER (HIT <ENTER> TO
QUIT) ";PH$
130 IF PH$="" THEN CLOSE : RUN
140 FOR DE=1 TO LOF(2) : GET 2,DE : IF LEFT$(F1$,1)<>"*" THEN
NEXT
150 RSET F1$=PH$ : PUT 2,DE : GOTO 120
200 REM
                ADD COMPANIES

210 GOSUB 1010 : CLS
```

```
22Ø CO$="" : LINEINPUT"ENTER COMPANY NAME (HIT <ENTER> TO
QUIT) ";CO$
23Ø IF CO$="" THEN CLOSE : RUN
24Ø FOR DE=1 TO NF : IF LEFT$(CO$(DE),LEN(CO$))=CO$ THEN
PRINT"NAME CONFLICTS WITH AN EXISTING FILE " : GOTO 22Ø ELSE
NEXT
25Ø CN$=CO$+STRING$(1Ø,"*")
26Ø FI$=LEFT$(CN$,8)+"/FIL"
27Ø FOR DE=1 TO LOF(1) : IF LEFT$(F$(Ø),1)<>"*" THEN NEXT
28Ø LSET F$(1)=CO$ : LSET F$(2)=FI$ : PUT 1,DE : NF=NF+1 :
CO$(NF)=F$(Ø) : GOTO 22Ø
3ØØ REM
                SEARCH EDIT

31Ø CLS : PRINT TAB(25)"EDIT FILES" : PRINT @256,"<D>ELETE
COMPANY NAME
<P>HONE NUMBER FILE EDIT
<Q>UIT" : GOSUB9ØØ
32Ø IF IK$="P" THEN 4ØØ ELSE IF IK$="Q" THEN 3Ø ELSE IF
IK$<>"D" THEN 31Ø
33Ø GOSUB 1Ø3Ø : PRINT TAB(15)"ENTER COMPANY TO DELETE 'Q' TO
QUIT "; : GOSUB 91Ø : IF IK$="Q" THEN CLOSE : GOTO 3Ø ELSE
GOSUB 1Ø5Ø : IF IK=Ø THEN 33Ø
34Ø GOSUB 1Ø1Ø
35Ø PRINT"ARE YOU SURE YOU WANT TO DELETE THIS FILE? (Y/N) "
: GOSUB 9ØØ
36Ø IF IK$="Y" THEN KILL FI$ : CO$(IK)="*" : LSET
F$(Ø)=STRING$(42,"*") : PUT 1,CO(IK) : CLOSE : RUN ELSE CLOSE
RUN
4ØØ REM
                EDIT PHONE NUMBERS

41Ø GOSUB 1Ø3Ø : GOSUB 1Ø4Ø : CLS
42Ø DE=1 : PRINT"SELECT A LETTER :  <A>DVANCE, <B>ACKUP,
<E>DIT, <D>ELETE, <Q>UIT" : GOSUB 1Ø6Ø
43Ø GOSUB 91Ø : IF IK$="A" THEN GOSUB 44Ø ELSE IF IK$="B"
THEN 45Ø ELSE IF IK$="D" THEN GOSUB 49Ø ELSE IF IK$="E" THEN
GOSUB 48Ø ELSE IF IK$="Q" THEN CLOSE : GOTO 3ØØ ELSE 43Ø
44Ø IF DE>LOF(2) THEN 43Ø ELSE DE=DE+1 : GET 2,DE : GOSUB 46Ø
: GOTO 43Ø
45Ø IF DE<3 THEN 43Ø ELSE DE=DE-1 : GET 2,DE : GOSUB 46Ø :
GOTO 43Ø
46Ø PRINT @16Ø,CHR$(31); : PRINT @256,"PHONE NUMBER #";DE-1;"
   ";F1$ : RETURN
48Ø INPUT"ENTER NEW NUMBER ";NE$ : RSET F1$=NE$ : PUT 2,DE :
DE=DE-1 : RETURN
49Ø LSET F1$="** DELETE **" : PUT 2,DE : DE=DE-1 : RETURN
5ØØ REM
                ENTER PHONE BILL

51Ø X=Ø : GOSUB 77Ø : CLS : PRINT"HIT <ENTER> FOR 'DATE' TO
QUIT" : PRINT STRING$(64,"=")
52Ø PRINT @256,; : O$(Ø)="" : FOR DE=ØTO3 : PRINT D$(DE+1),
: LINEINPU TO$(DE) : IF O$(Ø)="" THEN CLOSE : GOTO 6ØØ ELSE
NEXT : X=X+1 : PRINT @256,;CHR$(31);
53Ø LSET F2$(Ø)=O$(Ø) : RSET F2$(1)=O$(1) : RSET F2$(2)=O$(2)
: LSET F2$(3)=MKS$(VAL(O$(3))) : PUT3,X : GOTO 52Ø
6ØØ REM
                COMPUTES PHONE BILLS (SORT)

6Ø5 GOSUB 77Ø

61Ø X1=Ø : CLS : PRINT "NOW TO REVIEW THE LIST DO YOU WANT
TO VIEW LIST ON :

<P>RINTER
<V>IDEO
<S>KIP TO SORTING BILL
          (NOTE THE ITEM NUMBER TO EDIT IF NEEDED)" :
GOSUB 91Ø ELSE CLOSE : GOTO 3Ø
62Ø IF IK$="P" THEN GOSUB 69Ø : GOTO 635 ELSE IF IK$="S"
THEN 64Ø ELSE IF IK$<>"V" THEN 61Ø
63Ø GOSUB 121Ø : FOR DE=1 TO LOF(3) : GET3,DE : GOSUB 122Ø :
NEXT
635 Q$="" : PRINT"HIT <ENTER> TO SORT, <Q> TO QUIT, <E> TO
EDIT FILE" : GOSUB 91Ø : IF IK$="E" THEN 7ØØ ELSE IF IK$="Q"
THEN CLOSE : GOTO 3Ø ELSE IF IK$=CHR$(13) THEN 64Ø ELSE
GOTO 635
64Ø CLS : PRINT"STANDBY. SORTING YOUR NUMBERS" : X=LOF(3) :
FOR DE=1 TO X : GET 3,DE : PN$(DE)=F2$(1) : NEXT
65Ø FOR DE=2 TO NF : FI$=RIGHT$(CO$(DE),12) : CLOSE 2 : GT=Ø
: LPRINT LEFT$(CO$(DE),3Ø) : GOSUB 111Ø : GOSUB 75Ø
66Ø FOR DL=1 TO LOF(2) : GET 2,DL : F1$(DL)=F1$ : NEXT
67Ø FOR D=1 TO X : FOR D1=1 TO LOF(2) : IF LEN(F1$(D1))>1
THEN IF F1$(D1)=RIGHT$(PN$(D),LEN(F1$(D1))) THEN GET 3,D
: F2=CVS(F2$(3)) : LPRINT F2$(Ø),F2$(1),F2$(2),USING M$;F2
: PN$(D)=STRING$(12,32) : GT=GT+F2 : D1=Ø
68Ø NEXT D1,D : GOSUB 127Ø : GT=Ø : NEXT DE
685 LPRINT" " : LPRINT"MISCELLANEOUS NUMBERS" : LPRINT" " :
GOSUB 111Ø : FOR DE=1 TO X : IF RIGHT$(PN$(DE),1)=" " THEN
687 ELSE GET3,DE : F2=CVS(F2$(3)) : GT=GT+F2 : GOSUB 126Ø
687 NEXT : CLOSE : GOSUB 127Ø : GOTO 3Ø
69Ø GOSUB 124Ø : FOR DE=1 TO LOF(3) : GET3,DE : GOSUB 125Ø :
NEXT : GOTO635
7ØØ REM
                EDIT BILL FILE

71Ø INPUT"ENTER FILE NUMBER TO EDIT Ø TO QUIT";Q : IF Q=Ø
THEN 635
72Ø GET 3,Q : GOSUB 112Ø : PRINT : PRINT : PRINT"ENTER NUMBER
TO EDIT Ø TO QUIT" : GOSUB 91Ø : IK=VAL(IK$) : IF IK>4 THEN
72Ø ELSE IF IK=Ø THEN 635
725 IK=IK-1 : PRINT D$(IK+1);" OLD VALUE IS ";F2$(IK); :
INPUT" ENTER NEW VALUE ";NV$ : IF IK=Ø THEN LSET F2$(Ø)=NV$
ELSE IF IK=1 OR IK=2 THEN RSET F2$(IK)=NV$ ELSE IF IK=3 THEN
LSET F2$(3)=MKS$(VAL(NV$))
73Ø PUT 3,Q : GOSUB 112Ø : PRINT"IS THIS CORRECT Y/N " :
GOSUB 91Ø : IF IK$="Y" THEN 71Ø ELSE GOTO 72Ø
74Ø REM
          OPEN COMPANY PHONE LIST FILE

75Ø OPEN"R",2,FI$ : FIELD 2,12ASF1$ : RETURN
76Ø REM          OPEN FILE FOR BILL
77Ø OPEN"R",3,"PHONE/BIL" : FIELD 3,5 AS F2$(Ø),12 AS F2$(1),
3 AS F2$(2),4 AS F2$(3) : RETURN
8ØØ REM
                ENTER COMPANY NAMES

81Ø CLS : CO$="" : PRINT @32Ø,; : INPUT"ENTER COMPANY NAME
 (HIT <ENTER> IF DONE) ";CO$ : IF LEN(CO$)<8 THEN
CN$=CO$+STRING$(1Ø,"*")
```

continued on page 53

SOFTWARE REVIEWS

Super Utility Plus by Kim Watt

Joseph Rosenman

There are many utility programs for the Models 1 and 3. Quite a few of them are useful, quality programs. Even in this crowded market, Super Utility has made a well deserved name for itself. First, the facts:

Super Utility Plus
by Kim Watt
Breeze/QSD Inc.
11500 Stemmons Expwy., Suite 125
Dallas, Texas 75229

System: Model 1 or Model 3
RAM: 48K required
Disks: At least 1 disk required.
Language: Z-80 Machine Language.

Super Utility is a powerful and comprehensive disk access and modification utility. What is a disk utility? It is a program that permits the diskette to be analyzed, displayed, and altered. Many users of the the NEWDOS80 disk operating system (by Apparat) might be familiar with the SUPERZAP program. SUPERZAP is a disk utility that is included with NEWDOS80. I can say from a great deal of personal experience that Superzap is an extremely useful program. It has one distinct advantage over Super Utility: it is a CMD file. With that one exception, Super Utility is overflowing with extra features. I guess the best way to understand the scope of Super Utility is to examine the multiple Menus that it uses.

```
                Main Menu

1) Disk Zap              6) Tape Utilities
2) Disk Purge            7) Memory Utilities
3) Disk Format           8) File Utilities
4) Disk Backup           9) Configure System
5) Disk Repair           0) Exit Program
```

Each menu selection in turn calls up another menu. The menu "topics" are arranged in a logical fashion, and pretty much cover the gamut of utility functions. The next step in describing Super Utility is to describe the sub-menu details.

1) Disk Zap

1) Display sectors: Display any disk sector in Hex and ASCII.
2) Verify sectors: Make sure the data in the sector is correct.
3) Compare sectors: Determine whether any two sectors match.
4) Copy sectors (on the same disk or on any two disks).
5) Copy sector Data: This copy is similar to Copy Sectors, but the size is specified as a byte count rather then a sector count.
6) Zero sectors.
7) Reverse sector data: Byte 256 becomes byte 1, byte 255 becomes byte 2, etc.
8) Exchange sectors (swap a specific number of sectors).
9) String search: will search the disk for any length ASCII string or byte list (replacement is optional).
0) Sector search: Search for duplicate sectors.
A) Read ID address marks: Display track ID information.
B) Alter data address marks.

Note that the Display Sector option also allows you to modify the sector contents, and write it back to the disk.

2) Disk Purge

1) Kill selected files
2) Kill by category: by extension or attribute.
3) Remove system files.
4) Remove all passwords.
5) Disk Directory: with all of the trimmings.
6) Zero unused entries: cleans up unallocated FPDEs.
7) Zero unused granules: fills all unused sectors with 00s.
8) Change disk name.
9) Change file parameters: a fancy RENAME function.
0) Check directory: verifies that there are no errors.

3) Disk Format

1) Standard Format
2) Special Format: Allows you to custom configure the formatting.
3) Format without erase: will "clean up" the inter-sector gaps.
4) Build Format track (in memory).
5) Write Format track (from memory).
6) Software bulk erase: Wipe out all information from the diskette.

4) Disk Backup

1) Standard Disk Backup.
2) Special Disk Backup: Analyze a disk, then duplicate.

5) Disk Repair

1) Repair GAT sector.
2) Repair HIT sector.
3) Repair BOOT sector.
4) Read protect directory (read protection is a quick way to locate the directory in the TRSDOS compatible DOSes).
5) Un-read protect directory.
6) Recover killed files (as long as it wasn't killed by TRSDOS).
7) Move Directory (to any unallocated track).
8) Display Directory.
9) Check directory (thorough directory test).
0) Clear unused entries: zero out all unused FPDEs.

6) Tape Utilities

1) Read Tape.
2) Write Tape.
3) Verify Tape.
4) Copy Tape.

7) Memory Utilities

1) Display memory.
2) Move memory.
3) Exchange memory.
4) Compare memory: byte by byte comparison.
5) Fill memory.
6) Reverse memory: First byte becomes last, etc.
7) Test memory: Tests a range of memory with every possible bit pattern.
8) Jump to memory: transfers execution to a memory location.
9) String search: Search memory for an ASCII string or byte pattern.
0) Input byte from port.
A) Output byte to port.
B) Memory to sectors: dump memory to disk sectors.
C) Sectors to memory.
D) Memory to track (entire track).
E) Track to memory (reads all sectors and inter-sector gaps).

8) File utilities

1) Display file sectors.
2) Compare files.
3) Copy files: will copy a specified group of files all at once.
4) Disk Directory: provides a file oriented disk directory.
5) Free space: gives a summary of all space available on all drives.
6) Offset file: allows a file to be modified so that it will load at a different address.
7) File locations: provides a complete description of all files on a disk.
8) Drive status.
9) Sector allocation: Will determine what file is located at any given track and sector.
0) Build a file: pre-allocate a file on disk.
A) Clear a file: zero out all data sectors on a file.
B) Disk allocations: display allocation status for each track.
C) Compute hash code (for any filename).
D) Compute passwords: will encode or decode any password.

9) Configure System

Configuration can be "hard" or "soft." In "soft" configuration, the configuration table is altered. "Hard" configure requires that the Super Utility Plus disk program be zapped. This section of Super Utility is VERY VERY complicated. You can custom configure:

1) DOS type
2) "Real" track count
3) "Relative" track count
4) Directory location
5) Stepping rate
6) Initial drive delay
7) Software write protect
8) Disk density
9) Real track/relative track relationship
10) Track 0 density
11) Starting sector value
12) Data Address Mark (DAM)
13) Hi speed mod (Y/N)
14) Tie printer to display (Y/N)
15) Define high speed activate code
16) Define high speed deactivate code
17) Printer graphics (Y/N)
18) Lowercase (Y/N)
19) Epson MX-80 (Y/N)
20) Parallel (Y/N)
21) Line Feeds (Y/N)

0) Exit Super Utility

Super Utility doesn't require a system disk to be in drive 0 while it is running. When you are ready to exit Super Utility, you need to replace the system disk.

As you can see, Super Utility covers a very wide range of applications. What's more, it works! This is a very powerful and useful program. What would I suggest if Kim Watt asked me "How to improve Super Utility?"

1) Re-write the Configuration section so that it is clearer.
2) Permit configuration to be "hard" saved to disk without zapping.
3) Modify the "Special Copy" command so that "strange" information can be displayed/acted upon.
4) Permit the Special copy function to save the disk parameters and read them back.
5) Provide Super Utility as a CMD file (I know they won't, since CMD files are to easy to copy).
6) Use a larger print in the instructions.

By the way, the instructions are quite comprehensive. The program includes a 42 page instruction manual. What is more, there is a large section on particular systems (Model 1 or 3, NEWDOS80, DOSPLUS, LDOS, TRSDOS, etc.).

As you might have guessed by now, I believe that this is a very useful program. Since it greatly increases the power of your TRS-80, "no system should be without it." This is probably the best program of its type, and is well worth having.

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

THE CARDWRITER

continued from page 41

```
320 LINE INPUT TM$
330 PRINT:PRINT"ENTER YOUR OPINION:"
340 PRINT"WAS REALLY...
  1--EXCELLENT
  2--QUITE GOOD
  3--GOOD
  4--ONLY FAIR
  5--PRETTY BAD
  6--TERRIBLE"
350 INPUT OP
360 PRINT:PRINT"ENTER YOUR COMMENT,
  JUST <ENTER> IS NO COMMENT,
  TO ADD A LINE USE DOWN ARROW,
  4 LINES MAXIMUM:"
```

continued on page 52

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... C... 03

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 ... 1 year old $589. syst... 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... 6031... 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.

THE CARDWRITER

continued from page 50

```
37Ø PRINT"I THINK THAT..."
38Ø PRINT STRING$(36,"-")
39Ø CM$="":C=1
4ØØ A$=INKEY$:IF A$="" THEN 4ØØ
41Ø PRINTA$;
42Ø IF A$=CHR$(13) GOTO 47Ø
43Ø IF A$=CHR$(1Ø) THEN PRINT STRING$(5Ø,"-"):C=C+1
44Ø IF C=5 GOTO 47Ø
45Ø IF A$=CHR$(8) CM$=LEFT$(CM$,(LEN(CM$)-1)):GOTO 4ØØ
46Ø CM$=CM$+A$:GOTO 4ØØ
47Ø PRINT:PRINT"WHO IS THIS TO BE ADDRESSED TO:
  1--ABC
  2--CBS
  3--NBC
  4--PBS
  5--OTHER"
48Ø INPUT AD
49Ø IF AD=5 THEN LINE INPUT"SPECIFY NAME AND ADDRESS
USE DOWN ARROW FOR EACH LINE:
";AD$(5)
5ØØ PRINT:PRINT"PREPARE PRINTER AND CARD"
51Ø INPUT"ENTER <P> TO PRINT,
ENTER <S> TO START AGAIN";X$
52Ø IF X$="S" THEN RUN
53Ø IF X$="P" THEN 55Ø
54Ø GOTO 51Ø
55Ø PRINT
56Ø LPRINT"DEAR ";GR$(GR)
57Ø LPRINT"   YOUR SHOW, ";CHR$(34);SB$;CHR$(34);",";
58Ø LN=LEN(SB$)
59Ø IF LN>27 THEN LPRINT:LPRINT"WHICH WAS AIRED ";:F=3:GOTO 63Ø
6ØØ IF LN>23 AND LN<28 LPRINT" WHICH":LPRINT"WAS AIRED ";:
F=2:GOTO 63Ø
61Ø IF LN>17 AND LN<24 LPRINT" WHICH WAS":LPRINT"AIRED ";:
F=1:GOTO 63Ø
62Ø LPRINT" WHICH WAS AIRED"
63Ø LPRINT"ON ";DT$;" AT ";TM$;
64Ø IF F=1 LPRINT", I FOUND TO BE":LPRINT"REALLY ";:GOTO 68Ø
65Ø IF F=2 LPRINT", I FOUND TO":LPRINT"BE REALLY ";:GOTO 68Ø
66Ø IF F=3 LPRINT", I":LPRINT"FOUND TO BE REALLY ";:GOTO 68Ø
67Ø LPRINT", I FOUND TO BE REALLY"
68Ø LPRINT OP$(OP);"   "
69Ø IF CM$=""THEN 71Ø
7ØØ LPRINT"I THINK THAT ";CM$;"."
71Ø IF OP>3 THEN LPRINT"BETTER LUCK ON FUTURE SHOWS."
72Ø IF OP<4 THEN LPRINT"KEEP UP THE GOOD WORK."
73Ø LPRINT TAB(14)"SINCERELY,"
74Ø LPRINT TAB(18) MA$
75Ø PRINT:PRINT"TURN CARD OVER AND POSITION FOR ADDRESS."
76Ø INPUT"PRESS <ENTER> WHEN READY";X
77Ø LPRINT AD$(AD)
78Ø PRINT:INPUT"DO ANOTHER CARD (Y/N)";X$
79Ø IF X$="Y" THEN RUN
8ØØ END
81Ø PRINT"THIS IS YOUR NEW AND PERSONAL TV RESPONDER SYSTEM.
WITH IT YOU CAN NOW VERY EASILY COMMENT ON ALL THOSE TV
SHOWS THAT YOU LOVE OR HATE.  RUN IT, AND IT WILL PROMPT YOU"
82Ø PRINT"WITH A NUMBER OF COMMONLY USED PHRASES AND ALL YOU
NEED TO DO IS RESPOND WITH EITHER A NUMBER OR THE REQUIRED
TEXT.  YOU DON'T HAVE TO ADD PUNCTUATION AT THE END, THE PRO-
GRAM DOES IT FOR YOU. WHEN YOU ARE PROMPTED TO INPUT A LONG"
83Ø PRINT"STRING OF CHARACTERS, A LINE WILL APPEAR OVER THE
SPACE WHERE YOU WILL ENTER THE TEXT.  JUST DON'T GO BEYOND THE
END OF THE LINE AND YOUR COMMENTS WON'T AFFECT THE PROGRAM, AND
IT WILL FIT NICELY ON A STANDARD POSTCARD."
84Ø PRINT"WHEN YOU ARE ENTERING YOUR COMMENT THERE WILL BE NO
CURSOR, BUT ENTER TEXT AS USUAL, IT WILL BE PRINTED AND THE
BACKSPACE WORKS.  HAVE FUN."
85Ø INPUT"PRESS <ENTER> TO CONTINUE";X
86Ø RETURN
```

Anthony T. Scarpelli
98 Foxcroft Dr.
Scarborough ME 04074 ■

BEGINNER's CORNER

continued from page 14

Pathways Through the Rom: Guide to Level II Basic and DOS source code
Softside Publications — A compilation of several sources by George Blank, John Hartford, John T. Phillipp and Robert M. Richardson
P= 19.95 V= 25.ØØ

****

EXAMPLE 4. GRAPHICS

TRS-8Ø Graphics for the Model I and Model III
Byte books — R.S. Cat. No. 62-2Ø87 by Kater and Thomas
P= 1Ø.95 V= 13.ØØ

Graphic Software for Microcomputers
Kern — by Korites
P= 21.95 V= 25.ØØ
(Written for Apple but easily converted to TRS-8Ø)

****

EXAMPLE 5. GENERAL TECHNIQUES AND INFORMATION

Programming Design and Construction
Prentis-Hall — by David Higgens
P= 12.95 V= 8.95

The Basic Conversions Handbook for the Apple, TRS-8Ø and Pet Users
Hayden Brain Bank: David Brain, P. Oviate, P. Paquin, C. Stone
P= 7.95 V= 5.ØØ (on sale at a show)

Microcomputer Dictionary
Radio Shack — R.S. Cat. No. 62-2311 by Charles J. Sippl
P= 7.95 V= 1Ø.ØØ

****

I won't go into the advanced texts, except to say that the books in the remaining example are all good. I tend to think of the Barden, Heiserman, Rosenfelder, and Gratzer books as a complete, AND I MEAN COMPLETE, set which contains just about every trick that Basic is capable of, within those covers.

The Pathways book is good too, because it contains reference material that I have not seen in other texts. It also has part of the "Disassembled Handbook" (sort of a disassembled "Disassembled Handbook") which has the easiest and clearest explanation of how to use those ever-present "ROM" routines.

The graphics texts are the best I've come across. Both belong on your shelf. If any better ones come along, I'll be sure to let you know. The Graphics for Micros book is excellent because of its clear explanations of the theory. Most of the programs are easily convertible for the TRS-80, especially if you can write a routine that will connect two points, such as Med Systems' GRBASIC.

The book's General Techniques are all useful. You can find them at your local book shop and decide for yourself which suit your need for the moment.

Well, that about covers them all. Should you come across them at good prices (try college book stores and computer shows), by all means buy them. There is bound to be some repetition of information, but I've found that when one book isn't clear on something, another usually is. That topic, however, is another story. So, until next time, good programming to you.

Spencer Koenig
153-27 73 Avenue
Flushing NY 11367
compuserve 71625,1637 ■

PHONE BILL SORTING PROGRAM
continued from page 48

```
 82Ø IF CO$="" THEN CLOSE : RUN ELSE FI$=LEFT$(CN$,8)+"/FIL"
 83Ø FOR DE=2TOLOF(1) : IF LEFT$(F$(Ø),1)<>"*" THEN NEXT
 84Ø LSET F$(1)=CO$ : LSET F$(2)=FI$ : PUT1,DE : GOTO81Ø
 85Ø REM        TIMER FOR SCREEN
86Ø CLS : FOR X=1 TO 2ØØØ : PRINT @C,RIGHT$(TIME$,8); : IF
INKEY$<>"" THEN 88Ø ELSE NEXT
87Ø C=C+8 : IF C>55 THEN C=Ø : GOTO 86Ø ELSE GOTO 86Ø
88Ø CLS : TM=Ø : C=Ø : A=672 : RETURN
9ØØ REM          INKEY ROUTINE
91Ø IK$=INKEY$ : IF IK$="" THEN 91Ø ELSE RETURN
92Ø PRINT TAB(2Ø)"HIT ENTER TO CONTINUE"; : GOSUB 91Ø : CLS
: X1=Ø : RETURN
1ØØØ REM      SET UP ARRAY OF COMPANY NAMES
1Ø1Ø OPEN"R",1,"COMPANY/FIL" : FIELD 1,3ØASF$(1),12ASF$(2) :
FIELD 1,42ASF$(Ø) : RETURN
1Ø2Ø DL=Ø : IF LOF(1)<2 THEN 8ØØ ELSE FOR DE=2TOLOF(1) : GET
1,DE : IF LEFT$(F$(Ø),1)<>"*"ANDASC(LEFT$(F$(Ø),1))>32 THEN
DL=DL+1 : CO$(DL)=F$(Ø) : CO(DL)=DE : NEXT : NF=DL : RETURN
ELSE NEXT : NF=DL : RETURN
1Ø3Ø CLS : PRINT"THE FOLLOWING FILES ARE ON THIS DISK : " :
PRINT STRING$(64,"="); : FOR DE=2TONF : PRINTDE-1;"
";LEFT$(CO$(DE),3Ø) : NEXT : RETURN
1Ø4Ø PRINT TAB(15)"ENTER COMPANY NUMBER TO WORK ON "; :
GOSUB 91Ø : IK=VAL(IK$) : IF IK<1ORIK>NF-1 THEN 1Ø4Ø ELSE
IK=IK+1 : FI$=RIGHT$(CO$(IK),12) : PRINT" ";LEFT$(CO$(IK),3Ø)
: GOSUB 75Ø : RETURN
1Ø5Ø IK=VAL(IK$) : IF IK<1 OR IK>NF-1 : IK=Ø : RETURN ELSE
IK=IK+1 : FI$=RIGHT$(CO$(IK),12) : PRINT "  ";
LEFT$(CO$(IK),3Ø) : RETURN
1Ø6Ø A$=" WORKING ON "+RIGHT$(CO$(IK),12)+" FILE " :
DL=64-LEN(A$) : A1=INT(DL/2) : A$=STRING$(A1,"=") +
A$+STRING$(32,"=") : A$=LEFT$(A$,64) : PRINT @64,A$; : RETURN
11ØØ REM          LINE PRINTER HEADINGS
111Ø LPRINT "DATE","PHONE #","TIME (MIN)","   AMOUNT" :
LPRINT STRING$(6Ø,"-") : RETURN
112Ø CLS : PRINT : PRINT : PRINT : PRINT LI$;" 1  DATE";
TAB(25)F2$(Ø) : PRINT LI$;" 2  PHONE NUMBER";TAB(25)F2$(1) :
PRINT LI$;" 3  TIME";TAB(25)F2$(2) : PRINT LI$;" 4  COST";
TAB(25)USINGM$;CVS(F2$(3)) : RETURN
12ØØ REM        PRINT SUBROUTINE
121Ø PRINT"ITEM #  DATE","PHONE #","TIME","   COST" : RETURN
122Ø PRINT USING M2$;DE; : PRINT"    "; : PRINTF2$(Ø);"  ";
F2$(1),F2$(2),USINGM$;CVS(F2$(3)) : X1=X1+1 : IF X1=>14 THEN
GOSUB 92Ø : RETURN ELSE RETURN
123Ø REM         LPRINT SUBROUTINE
124Ø LPRINT"ITEM #","DATE","PHONE #","TIME","COST" : RETURN
125Ø LPRINT USINGM2$;DE; : LPRINT,F2$(Ø),F2$(1),F2$(2),USING
M$;CVS(F2$(3)) : RETURN
126Ø LPRINT F2$(Ø),F2$(1),F2$(2),USING M$;F2 : RETURN
127Ø LPRINT" " : LPRINT"THIS TOTAL =";USING M$;GT : LPRINT" "
: LPRINT" " : RETURN
13ØØ CLOSE : CLS : PRINT : PRINT : PRINT : PRINT"ARE YOU SURE
YOU WANT TO KILL OLD PHONE BILL Y/N " : GOSUB 91Ø : IF
IK$="Y" THEN KILL"PHONE/BIL" : GOTO 3Ø ELSE GOTO 3Ø
5ØØØ DATADATE,PHONE #,TIME,COST
51ØØ REM         READ DATA
511Ø FOR X2=1 TO 4 : READ D$(X2) : NEXT : LI$="LINE #" : RETURN
```

CHANGES

As it stands, the program limits the number of company files to 9 and the number of monthly numbers entered to 250. To change the phone number limit, modify the DIM statements on line 20 with the variables PN$(250) and F1(250).

Do not attempt to run a sort unless you have placed at least one phone number under each company file which you have created. Otherwise, when the machine opens a company file it won't find any phone numbers there to compare and will end with an error. However, this is really no problem, because there would never be an occassion in which you would create a file and not have at least one number to put in it.

Dan Keen and Dave Dischert
Soft Horizons
RD 1 Box 432
State Highway 83
Cape May C.H., NJ 08210 ■

CORRECTION

The program that was published with "The Transportation Method of Linear Programming" in the August 1982 issue was unfortunately incorrect. The entire program that appears there should be replaced by the following:

```
10 CLEAR 800: ON ERROR GOTO 720  :REM    "TRANS"
20 A$="###   ####   ####   ###      ###   #####.##     #####.##
    #####.##"
30 CLS: PRINT "TRANSACTION PROGRAM ": INPUT "DATA NAME";F$:
INPUT "DISK";X$
40 FX$=F$+":"+X$: PRINT: INPUT "MAXIMUM NUMBER OF
TRANSACTIONS";D: DIM A%(D,4),A#(D,2)
50 INPUT "SELECT DISK OR KEY INPUT (D OR K)";T$
60 IF T$="K" THEN 110
70 IF T$<>"D" THEN 50
80 I=1:OPEN"I",1,FX$
90 FOR J=1 TO 4: INPUT #1,X%:A%(I,J)=X%: NEXT J:K=I: INPUT
#1,X#,Y#:A#(I,1)=X#:A#(I,2)=Y#: IF A%(I,1)=0 THEN 150
100 I=I+1: GOTO 90
110 PRINT "CHECK NO., MONTH+DAY,ACCOUNT-DEBIT,CREDIT,
AMOUNT-DEBIT,CREDIT":K=1
120 FOR Z=K TO D:K=Z
130 INPUT A%(Z,1),A%(Z,2),A%(Z,3),A%(Z,4),A#(Z,1),A#(Z,2): IF
A%(Z,1)=0 THEN 150
140 NEXT Z
150 CLOSE 1: INPUT "INPUT LINE NUMBERS TO BE LISTED OR 0,0
WHEN COMPLETE";A,B
160 IF A=0 THEN 190
170 CLS: PRINT "NO   TRANS    DATE ACC-DEBIT ACC-CREDIT
AMT-DEBIT  AMT-CREDIT"
180 FOR I=A TO B: PRINT USING A$;I,A%(I,1),A%(I,2),A%(I,3),
A%(I,4),A#(I,1),A#(I,2): NEXT I
190 INPUT "SELECT LINE NUMBER TO BE CHANGED, 0 TO LIST, -1 TO
ADD DATA, -2 TO RECORD, -3 TO INSERT, -4 TO DELETE, -5 TO SUM
DEBITS/CREDITS, -6 FOR PRINTING OR -7 TO RETURN TO GLMENU";P
200 IF P=-7 THEN LOAD"GLMENU",R
210 IF P=0 THEN 150
220 IF P=-1 THEN PRINT "CK NO,MO+DAY,ACCOUNT DEBIT-CREDIT,
AMOUNT DEBIT-CREDIT": GOTO 120
230 IF P=-2 THEN 680
240 IF P=-6 THEN 410
250 IF P<>-3 THEN 320
260 INPUT "NUMBER OF INSERTED LINE 0 TO DEFAULT";P: IF P=0
THEN 190
270 K=K+1:W=0:FOR I=D-1 TO P STEP-1: IF A%(I,1)=0 THEN 300
280 IF W=0 THEN W=1:FOR J=1 TO 4:A%(I+2,J)=A%(I+1,J): NEXT J:
FOR J=1 TO 2 : A#(I+2,J)=A#(I+1,J): NEXT J
290 FOR J=1 TO 4:A%(I+1,J)=A%(I,J): NEXT J: A#(I+1,1)=A#(I,1)
: A#(I+1,2)=A#(I,2)
300 NEXT I
310 GOTO 400
320 IF P<>-4THEN360
330 INPUT"LINE NUMBER TO DELETE 0 TO DEFAULT";P: IF P=0 THEN
190
340 K=K-1: FOR I=P TO D-1: IF A%(I,1)=0 THEN FOR J=1 TO 4:
A%(I,J)=0: NEXT J: FOR J=1 TO 2:A#(I,J)=0: NEXT J: GOTO 180
350 FOR J=1 TO 4:A%(I,J)=A%(I+1,J): NEXTJ : A#(I,1)=A#(I+1,1)
: A#(I,2)=A#(I+1,2): NEXT I: GOTO 180
360 IF P<>-5 THEN 400
370 SD#=0:SC#=0: FOR I=1 TO D: IF A%(I,1)=0 THEN 390
380 SD#=SD#+A#(I,1):SC#=SC#+A#(I,2): NEXT I
390 BA$="###,###,###.##": PRINT"SUM DEBITS ";: PRINT USING
BA$;SD#;: PRINT"  SUM CREDITS ";: PRINT USING BA$;SC#: GOTO
190
400 PRINT "CK NO.,MO+DAY, ACCOUNT-DEBIT,CREDIT, AMOUNT-DEBIT,
CREDIT":INPUT A%(P,1),A%(P,2),A%(P,3),A%(P,4),A#(P,1),A#(P,2)
: GOTO 180
410 INPUT "LINE COUNTER (Y/N)";LC$: IF LC$="Y" THEN CMD
"FORMS(T)"
420 XX$=" NO. CK NO. MO+DAY  DEBIT  CREDIT  $DEBIT
        $CREDIT       $BALANCE"
430 INPUT "INPUT DATE OF RUN XX/XX/XX";DA$
440 Z$="      BALANCE    ##########.##"
450 INPUT "INPUT ORIGINAL BALANCE";BA#:SD#=0:SC#=0
460 LPRINT" ":LPRINT"   DATE    ";DA$:LPRINT USING Z$;BA#
470 LPRINT XX$
480 CLS: PRINT "PRINTER MENU": PRINT "     1     ALL
ACCOUNTS": PRINT"     2     SELECTED TRANSACTIONS": PRINT
"     3     SELECTED DATES": PRINT"     4     SELECTED
ACCOUNT": INPUT GH%
490 IF GH%=1 THEN 560
500 IF GH%<>2 THEN 520
510 INPUT "INPUT MINIMUM AND MAXIMUM TRANSACTION NUMBERS";
SS%,BB%: GOTO 560
520 IF GH%<>3 THEN 540
530 INPUT "INPUT EARLYEST DATE AND LATEST DATE (MODA)";
SS%,BB%: GOTO 560
540 IF GH%<>4 THEN 480
550 INPUT "ACCOUNT YOU WISH TO EXAMINE";SS%
560 FOR I=1 TO D: IF A%(I,3)=1 THEN BA#=BA#+A#(I,1)
570 IF A%(I,4)=1 THEN BA#=BA#-A#(I,2)
580 IF A%(I,1)=0 THEN BA$="###,###,###.##":LPRINT "SUM DEBITS
";: LPRINT USING BA$;SD#;:LPRINT "  SUM CREDITS ";: LPRINT
USING BA$;SC#: GOTO 190
590 IF GH%=1 THEN 650
600 IF GH%=2 AND SS%<=A%(I,1) AND BB%>=A%(I,1) THEN 650
610 IF GH%=3 AND SS%<=A%(I,2) AND BB%>=A%(I,2) THEN 650
620 IF GH%=4 AND A%(I,3)=SS% THEN 650
630 IF GH%=4 AND A%(I,4)=SS% THEN 650
640 GOTO 660
650 LPRINT USING A$;I,A%(I,1),A%(I,2),A%(I,3),A%(I,4),
A#(I,1),A#(I,2),BA#:SD#= SD#+A#(I,1):SC#=SC#+A#(I,2): NEXT I
660 NEXT I: LPRINT "SUM DEBITS ";:LPRINT USING BA$;SD#,"  SUM
CREDITS  ";: LPRINT USING BA$;SC#
670 GOTO 190
680 OPEN"O",1,FX$: FOR IX=1 TO D: FOR IQ=1 TO 2:
DZ#=INT(100D*A#(IX,IQ))/100D:DX#=A#(IX,IQ)-DZ#:
A#(IX,IQ)=DZ #: IF DX#>.005D THEN A#(IX,IQ)=A#(IX,IQ)+.01D
690 NEXT IQ,IX:IX=0
700 IX=IX+1: PRINT #1,A%(IX,1);A%(IX,2);A%(IX,3);A%(IX,4);:
FOR IQ=1 TO 2: PRINT #1,A#(IX,IQ): NEXT IQ: IF IX<D THEN 700
710 CLOSE 1: GOTO 190
720 RESUME 730
730 PRINT "AN ERROR HAS OCCURRED SAVE YOUR FILES--ON NEW DISK
IF NECESSARY": PRINT: GOTO 190  ■
```

PROGRAM CONVERSION (PART 10)

Richard Kaplan

After nine months of writing this series, I have amassed an abundance of reading material pertaining to program conversion between the MODEL I, MODEL II, MODEL III, APPLE, and CP/M computers. No doubt this information may be a bit cumbersome to read completely and utilize; therefore, I will devote a large portion of this month's article to a presentation of charts summarizing most information which I have presented in the last nine months.

CONVERSION TO MODEL I/III FROM MODEL II

MODEL II COMMAND	MODEL I/III EQUIVALENT	ISSUE #
PRINT @ X	PRINT @ (INT(X/8Ø)*64) + (X-INT(X/8Ø)*8Ø))	44
OPEN "R",1,"XXX",128	OPEN "R",1,"XXX" (MOD I ONLY)	44
OPEN "D",1,"XXX"	OPEN "R",1,"XXX"	44
OPEN "E",1,"XXX"	READ AND REWRITE FILE	44
SWAP A,B	T=A:A=B:B=T OR CMD"F=SWAP",A,B (UNDER NEWDOS/8Ø)	47, 5Ø
ERASE A$	CMD"F=ERASE",A$ (NEWDOS/8Ø) CMD"L"A$(Ø) (MULTIDOS)	47, 5Ø
PRINT USING "/ /"	PRINT USING "% %"	47
A MOD B	A-INT(A/B)*B	47
SYSTEM"I"	NOT NECESSARY	47
SYSTEM	CMD"S"	47
SYSTEM"XXX"	CMD"XXX" (ONLY WITH AN ALTERNATIVE DOS) (MOD I)	47
	CMD"I","XXX" (TRSDOS 1.3)	
SYSTEM"SCREEN"	SHFT DWN ARW * (MOD III OR DOSPLUS AND MOD I)	48
	JKL (MOD I OR III WITH NEWDOS/8Ø)	
ERR	ERR/2 + 1 (ALSO CHANGE ERROR CODES)	48,49
PRESSING HOLD KEY	PRESS SHIFT @	48

CONVERSION TO MODEL II FROM MODEL I/III

MODEL I/III COMMAND	MODEL II EQUIVALENT	ISSUE #
PRINT @ X	PRINT @ (INT(X/64)*8Ø) +(X-(INT(X/64)*64))+32Ø	44
OPEN "E",1,"XXX"	READ AND REWRITE FILE	44
SET (X,Y)	DEF FNA(X)=(INT(X/64)*8Ø) +(X-(INT(X/64)*64))+32Ø: PRINT @ FNA(Y*64+X),CHR$(158)	44
PRINT USING "% %"	PRINT USING "/ /"	47
IF XXX YYY	IF XXX THEN YYY	48
PRESSING SHIFT @	PRESS HOLD KEY	48
ERR/2+1	ERR (ALSO CHANGE ERROR CODES)	48

CONVERSION TO CP/M FROM MODEL II

MODEL II COMMAND	CP/M EQUIVALENT	ISSUE #
CLS	PRINT CHR$(X) (DEPENDS ON COMPUTER)	46
PRINT @ X,"ZZZ"	DN-INT(Q/8Ø) AC=Q-INT(Q/8Ø)*8Ø PRINT CHR$(X) IF DN<>0 THEN PRINT STRING$(DN,Y); IF AC<>0 THEN PRINT STRING$(AC,Z); PRINT "ZZZ" (X=HOME CURSOR Y=CURSOR DOWN Z=CURSOR ACCROSS)	46
OPEN "R",1,"ZZZ"	CP/M DOES NOT SEARCH DRIVES	46
LOF	MAINTAIN MANUALLY OR COMPUTE	46
FORI=1TO1Ø	FOR I = 1 TO 1Ø (LEAVE SPACES BETWEEN KEYWORDS)	46
REL=RE+1	RE=RE+1 (CP/M RECOGNIZES 5-LETTER VARIABLES)	46

CONVERSION TO HP MBASIC FROM MICROSOFT BASIC

MICROSOFT COMMAND	HP EQUIVALENT	ISSUE #
LEFT$(A$,Y)	A$(1,Y)	49
RIGHT$(A$,Y)	A$(LEN(A$)-Y+1,LEN(A$))	49
MID$(A$,X,Y)	A$(X,X+Y-1)	49
C$=A$+B$	C$=A$:C$(LEN(C$)+1)=B$	49
ARRAY HANDLING	SEE TEXT OF ARTICLE	49

CONVERSION TO APPLE FROM TRS-8Ø

TRS-8Ø COMMAND	APPLE EQUIVALENT	ISSUE #
CLS	HOME	45
DISK ACCESS	SEE TEXT OF ARTICLE	45

A PRACTICAL EXAMPLE

I recently received a letter from Stuart Reiner, 222 West 15th Street, New York, NY 10011. I believe his letter is appropriate for this column, because it is an excellent example of the need for program conversion techniques, and it can also help to explain the process one must utilize when attempting to convert BASIC programs.

Dear Mr. Kaplan:

I own a TRS-80 MODEL III (48K) with one disk drive. I have been encountering difficulties in translating a program written for the Apple II in Integer BASIC. I use TRSDOS.

The programs I have been trying to convert are part of a Pro Football predicting system included in the book "BASIC Betting — The Micro-Computer Edge" by James Jasper (a copy of the program as written by Mr. Jasper is enclosed).

I have enclosed a copy of the program as translated by me. The program runs through all right, except for the loop contained in the program at lines 7150 and 10235. I have tried entering the FOR and NEXT statements in several different ways, but I have been unsuccessful at all attempts to get the program to excute the loop. At this point the program is supposed to return so that information for a second team may be entered and hn the third team, etc. until the information for all 28 teams for the current week has been entered.

I would appreciate any help you can give me in overcoming this problem.

Thank you very much.

Very truly yours,
Stuart Reiner, CPA

For legal reasons, it is not possible for me to reproduce the entire program listing here which Mr. Reiner sent to me. Instead, I will include only those portions of the program which required conversion.

VS. <>

In the original (Integer BASIC) program, there were a number of lines which were similar to the following:

```
300 IF A$#A1$ THEN 310:A=1:RETURN
```

This line contains a "#", which represents "not equal to" on the APPLE in Integer BASIC. On the MODEL-III (or any other TRS-80, for that matter), the "not equal to" operator is "<>". As a result, our corrected line should read as follows:

```
300 IF A$<>A1$ THEN 310: A=1: RETURN
```

The subsitution of <> for # is a snap. This should be an easy conversion. Right? *Wrong!* There is one more element of this line which still must be converted.

IF ... THEN

If you are at all experienced with TRS-80 programming and you examine the above line, you will see that the end of the line, i.e., "A=1:RETURN", would never be executed on a TRS-80. If an "IF" condition is true on the TRS-80, each command on the line is then executed. If the condition is false, execution of the program jumps to the next line number. On a TRS-80, then, if A$ were not equal to A1$, program execution would jump to line 310, since there is a GOTO statement there. The portion of the line which reads "A=1:RETURN" would never be executed.

On the APPLE with Integer BASIC (NOT with APPLESOFT), the end of the line would be executed if the "IF" clause were not true. An equivalent set of program lines for a TRS-80, then, would be the following:

```
300 IF A$<>A1$ THEN 310
305 IF A$=A1$ THEN A=1:RETURN
```

IF ... THEN ... ELSE

There is a programming concept on the TRS-80's which could reduce the two lines above into one line. The sequence "IF ... THEN ... ELSE" is that concept. In the two lines above, IF A$ is equal to A1$, THEN program execution branches to line 310, or ELSE A is set to 1 and the subroutine is terminated. In computerese, we could state this as follows:

```
300 IF A$<>A1$ THEN 310 ELSE A=1:RETURN
```

DISK ACCESS AND CHR$(4)

In May, 1982 (issue # 45), I discussed in-depth the handling of APPLESOFT disk files, which are virtually identical to Integer disk files. However, there is particular point, which was needed in Mr. Reiner's progam, which may very well cause quite a bit of confusion. Examine the folowing set of lines, which appeared in the program Mr. Reiner sent in:

```
10 D$=""
20 PRINT D$;"OPEN WEEKLY";GNO
7000 PRINT D$;"WRITEWEEKLY";GNO

7003 PRINT #1,A$
7160 PRINTD$;"CLOSEWEEKLY";GNO
```

A knowledgable but not completely experienced TRS-80 programmer might see this APPLE program and translate it as follows:

```
10 D$=""
20 PRINT D$:OPEN "O",1,"WEEKLY":PRINT GNO
7000 PRINT D$:PRINT #1,"WRITEWEEKLY";GNO
7003 PRINT #1,A$
7160 CLOSE #1
```

Let's now examine this attempted conversion and see what might have been easily overlooked:

```
10 D$=""
```

Line 10 in the APPLE program appears innocent enough. Variable A$ is simply being set to a length of 0. Right? *Wrong!* As I mentioned in my May column, D$ is frequently used to denote CHR$(4), which is needed on the APPLE to specify a disk command. However, CHR$(4), which is CTRL-D, is not actually printed on the screen, even though it is part of the program. The original program author should have documented this in his program. However, since he didn't, we must realize this from context. Since CHR$(4) is not needed for TRS-80 disk access, we can delete this line.

```
20 PRINT D$;"OPEN WEEKLY";GNO
20 PRINT D$:OPEN"O",1,"WEEKLY":PRINT GNO
```

This conversion again seems to be quite obvious. The original statement prints D$, opens the file, and prints the value of GNO. Right? Wrong again.

As with our previous example, D$, which is equal to CHR$(4), is not needed on the TRS-80's. The OPEN command was not converted correctly either. On the APPLE, if GNO is equal to 1, the command PRINT D$;"OPEN WEEKLY";GNO will open file WEEKLY1. On the TRS-80, the name of the file must be a string; it cannot be composed of a string portion

continued on page 63

SOFTWARE REVIEW

Defense Command from Big Five Software

John Wilson

Are you looking for a fast-action, truly challenging game? Look no further. Defense Command from Big Five Software is the game for you. While the game is modelled after the popular arcade game, there are quite a few changes that have been made.

You start out with 3 space ships and 4 anti-matter bombs. You are alone, trying to protect 10 KROTNIUM fuel cells from the KROMORFKROM galaxy. As the enemies fly down, you must position your ship directly below them, then fire. If you miss, you won't be able to fire again until the laser shot leaves the screen. You are awarded 40 to 90 points for each alien shot. If an alien gets down to the bottom, he will grab a canister and fly off with it. Shooting him will award double his value, and the cell will start to fall. Being very fragile (the price you pay for nuclear power), you must catch the cells, or they will explode. A caught cell awards either 300 or 400 points.

As your skills improve, there are more aliens, each moving faster. If the situation gets out of hand, an anti-matter bomb (AMB) can be used. Pressing any of the number keys (0-9), or aiming the joystick up will detonate one of the bombs. All aliens on the screen are destroyed, and any cells the aliens were carrying will fall.

There are two special alien ships that should be mentioned. These are the Flagship and he Slicer. The flagship is the same on as in Robot Attack and Attack Force, but is no danger here. It flies along at different levels, carrying a cargo of ships. The flagship is worth 500 points, but any aliens inside will be released. The slicer has been programmed to destroy fuel cells. The explosion from the cell will destroy the slicer. A slicer appears only when your supply of cells is low, or when released from the flagship.

The game ends when you run out of ships or all of the fuel cells have been destroyed. As an aid in playing the game, an extra ship will be sent to you every 10,000 points. At every 5,040 points, you will receive another AMB.

This is one of the better games available today. It has clear, fast graphics, along with excellent sound. One or two people can play the game, with the 10 highest scores saved. It is an excellent program, and is very addictive.

For those of you who like to get a higher score than the program gives, I have found out the way to get a higher cannister bonus. Load the program normally, but do not execute it. Go into the basic Ready mode, and type: POKE 24616,62:POKE 24617,6:POKE 24619,0:POKE 24620,0
After this, type "SYSTEM", then "/18750". You should receive 900 points for each cell caught. To get a larger bonus, increase the value that is poked into location 24617.

John Wilson
204-79 Russell St. W.
Lindsay, Ontario
Canada K9V 2X3 ■

MACSMAP

W. G. MacSwain

"MACSMAP" is a program that creates a map of the entire granule usage of a diskette, showing what is assigned to what files. It is written for a 48K TRS-80 Model I with a printer and will work with TRSDOS, NEWDOS, NEWDOS80 Version 1, and LDOS. Since being written, it has been subjected to much use.

The program works by poking an assembly language routine into high memory (A000H) that reads the target diskette's complete directory into memory starting at B000H.

```
********************* M A C S   D I S K   M A P **********************

DISKETTE -    NEWDOS            Ø7/Ø1/8Ø
TRACKS   -  AVAILABLE  35     LOCKED OUT    5
GRANULES -   ASSIGNED  54          FREE  16          FXDE'S -  Ø
SECTORS  -   ASSIGNED 27Ø          FREE  8Ø           FPDE'S - 19

TRACK>     <1ST GRANULE>           <2ND GRANULE>        <FILE LENGTH>
DC HX>----<SECTORS Ø-4>----------<SECTORS 5-9>----<SECTORS/BYTES-EXTENTS
*************************************************************************
 Ø ØØ < 1  BOOT    /SYS  1>  < 1  SYSØ    /SYS  3> WORDP/NEW 25/147-2
 1 Ø1 < 2  SYSØ    /SYS  3>  < 3  SYSØ    /SYS  3> DATABASE 24/175-2
 2 Ø2 < 1  FORMAT  /CMD  3>  < 2  FORMAT  /CMD  3> FILES 55/121-3
 3 Ø3 < 3  FORMAT  /CMD  3>  < 1  BASIC   /CMD  4> WORDP/OLD 25/32-1
 4 Ø4 < 2  BASIC   /CMD  4>  < 3  BASIC   /CMD  4>
 5 Ø5 < 1    DATABASE    5>  < 2    DATABASE    5>
 6 Ø6 < 3    DATABASE    5>  < 1  COPY    /CMD  1>
 7 Ø7 < 4    DATABASE    5>  < 5    DATABASE    5>
 8 Ø8 < 1  WORDP   /NEW  6>  < 2  WORDP   /NEW  6>
 9 Ø9 < 3  WORDP   /NEW  6>  < 4  BASIC   /CMD  4>
1Ø ØA < 4  WORDP   /NEW  6>  < 5  WORDP   /NEW  6>
11 ØB < 6  WORDP   /NEW  6>  < 1    FILES      12>
12 ØC < 2    FILES      12>  < 1  SYS13   /SYS  1>
13 ØD < 3    FILES      12>  < 4    FILES      12>
14 ØE < 5    FILES      12>  < 6    FILES      12>
15 ØF < 7    FILES      12>  < 8    FILES      12>
16 1Ø < 1  SYS1   /SYS  1>  < 1  SYS2    /SYS  1>
17, 11 < 1  DIR     /SYS  2>  < 2  DIR     /SYS  2>
18 12 < 1  SYS3   /SYS  1>  < 1  SYS4    /SYS  1>
19 13 < 1  SYS5   /SYS  1>  < 1  SYS6    /SYS  3>
2Ø 14 < 2  SYS6   /SYS  3>  < 3  SYS6    /SYS  3>
21 15 < 1  WORDP  /OLD  6>  < 2  WORDP   /OLD  6>
22 16 < 3  WORDP  /OLD  6>  < 4  WORDP   /OLD  6>
23 17 < 5  WORDP  /OLD  6>  < 6  WORDP   /OLD  6>
24 18 < 9    FILES      12>  <1Ø    FILES      12>
25 19 <11    FILES      12>  <12    FILES      12>
26 1A < Ø   -- FREE --   Ø>  < Ø   -- FREE --   Ø>
27 1B < Ø   -- FREE --   Ø>  < Ø   -- FREE --   Ø>
28 1C < Ø   -- FREE --   Ø>  < Ø   -- FREE --   Ø>
29 1D < Ø   -- FREE --   Ø>  < Ø   -- FREE --   Ø>
3Ø 1E < Ø   -- FREE --   Ø>  < Ø   -- FREE --   Ø>
31 1F < Ø   -- FREE --   Ø>  < Ø   -- FREE --   Ø>
32 2Ø < 1  SYS11  /SYS  1>  < Ø   -- FREE --   Ø>
33 21 < Ø   -- FREE --   Ø>  < Ø   -- FREE --   Ø>
34 22 < Ø   -- FREE --   Ø>  < 1  SYS12  /SYS  1>
35 23 < Ø    LOCKED OUT   Ø>  < Ø    LOCKED OUT   Ø>
36 24 < Ø    LOCKED OUT   Ø>  < Ø    LOCKED OUT   Ø>
37 25 < Ø    LOCKED OUT   Ø>  < Ø    LOCKED OUT   Ø>
38 26 < Ø    LOCKED OUT   Ø>  < Ø    LOCKED OUT   Ø>
39 27 < Ø    LOCKED OUT   Ø>  < Ø    LOCKED OUT   Ø>
          COPYRIGHT (C) 1981 BY MACSWAIN ENTERPRISES
```

Example 1: "Macs Disk Map" for a sample diskette.

```
DRIVE  1  NEWDOS   Ø7/Ø1/8Ø   35 TRKS   45 FDES   16 GRANS

WORDP/NEW          256=LRECL        26 RECS        6 GRANS
DATABASE           256=LRECL        25 RECS        5 GRANS
FILES              256=LRECL        56 RECS       12 GRANS
WORDP/OLD          256=LRECL        26 RECS        6 GRANS
```

Example 2: A disk directory (with attributes) under NEWDOS80 Version 1 for the same diskette as in Example 1.

When told to "Load Target Diskette in Drive 0 and Push <Enter>", you can, as an alternative, put the diskette in drive 1, key in a 1, and then <Enter>. Both of these produce a "file length" listing (far right column) that suppresses invisible and system files. If all files are required, put the target diskette into drive 0, key in a 2, and then <Enter>.

The program builds several arrays as it goes through the "GAT" table to see how many granules are used, free or locked out. Next it goes through all the FXDEs and FPDEs, displaying a countdown on the screen and again building arrays. Finally, printing begins as the arrays are checked for the appropriate codes on each track. Simultaneously the file names are printed down the right column with the number of sectors/bytes, and extents for each.

The file name for each granule on the map is followed by a number indicating the total number of granules that file has. The number preceding the file name indicates what number that particular granule is.

Example 1 shows a sample map performed on a 40-track diskette under NEWDOS80 Version 1. BOOT/SYS occupies track 0, granule 1 (sectors 0-4), SYS0/SYS track 0, granule 2 through track 1, granule 2, etc. The file named "FILES" has three allocations or extents: the first is track 11 granule 2 through track 12 granule 1, the second track 13 granule 1 through track 15 granule 2, and the third track 18 granule 1 through track 19 granule 2. (The last five tracks are shown as being locked out because the diskette is formatted only for 35 tracks.) Example 2 shows a disk directory with attributes performed for the same diskette.

To save time while typing in the program, omit any comment line (line that begins with an apostrophe).

```
1ØØ        'MACSMAP' - 2/1Ø/81 VER 1.2 - 48K
1Ø1 '  COPYRIGHT (C) 1981 BY MACSWAIN ENTERPRISES
1Ø2 '  151 CASS AVENUE, AGINCOURT, ONTARIO M1T 2B5
1Ø3 '             ALL RIGHTS RESERVED
1Ø5 '   LOAD 'DISK READING' PROGRAM INTO MEMORY AT AØØØH.
```

```
11Ø FORX=-24576TO-24441:READY:POKEX,Y:NEXT
115 '   'CLEAR' REQUIRED TO RESET VARIABLES FOR SECOND TIME
THROUGH.
116 '   Y(2,3,4Ø): GRAN 1 OR 2: HOW USED, TOT GRANS, GRANS SO
FAR: TRACK
117 '   YN$(2,4Ø): FILE NAME-GRAN 1 OR 2: TRACK
118 '   SN$: FILE NAME: - SN(3,64): NO OF SECTORS, NO OF
BYTES, NO OF EXTENTS
12Ø CLEAR6ØØØ:DIMY(2,3,4Ø),YN$(2,4Ø),SN$(64),SN(3,64)
125 '   PRINT SCREEN TITLE AND INSTRUCTIONS:
126 '           <ENTER>    = READ DRIVE Ø, SKIP SYS & INVIS
FILES
127 '           1 <ENTER>  = READ DRIVE 1, SKIP SYS & INVIS
FILES
128 '           2 <ENTER>  = READ DRIVE Ø, PRINT ALL FILES
13Ø CLS:PRINTCHR$(23);:PRINT@334,"M A C S   M A P":PRINT
14Ø PRINT:PRINT"LOAD TARGET DISKETTE IN DRIVE Ø"
15Ø Y$="":LINEINPUT"         PUSH <ENTER>";Y$
155 '   SELECT DRIVE 1
16Ø IFY$="1"THENPOKE-24574,2:POKE-245Ø7,2:Y$=""
165 '   GO READ DISK AND RETURN HERE - RESTORE DRIVE Ø
SELECTION.
17Ø DEFUSR1=&HAØØØ:B=USR1(B):POKE-24574,1:POKE-245Ø7,1
175 '   A=BYTE BEFORE FIRST BYTE OF SECTOR Ø DIRECTORY
(AFFFH).
176 '   H$=DECIMAL TO HEX CONVERSION STRING.
18Ø A=-2Ø481:MA$="###":H$="Ø123456789ABCDEF"
185 '   PRINT PAGE HEADER.
19Ø PRINT@4ØØ,"T U R N   O N   P R I N T E R !"
2ØØ LPRINTSTRING$(22,"*");" M A C S   D I S K   M A P
";STRING$(23,"*"):CLS:LPRINT
2Ø5 '   PRINT DISKETTE NAME AND CREATION DATE.
21Ø LPRINT"DISKETTE -   ";
22Ø FORX=2Ø9TO216: LPRINTCHR$(PEEK(A+X));: NEXTX: LPRINT
STRING$(1Ø," ");
23Ø FORX=217TO224:LPRINTCHR$(PEEK(A+X));:NEXTX:LPRINT
235 '   COUNT NUMBER OF TRACKS AVAILABLE FROM GAT TABLE. IF
TRACK LOCKED OUT, MOVE CODE 2 INTO Y ARRAY.  PRINT BOTH
TOTALS.
24Ø T=Ø:R=Ø:FORX=97TO136
25Ø IFPEEK(A+X)=252THENT=T+1
26Ø IFPEEK(A+X)=255THENY(1,1,X-96)=2:Y(2,1,X-96)=2
27Ø NEXT
28Ø LPRINT"TRACKS   -  AVAILABLE ";
29Ø LPRINTUSINGMA$;T;:LPRINT"    LOCKED OUT ";:LPRINT USING
MA$;4Ø-T
294 '   BUILD Y ARRAY ACCORDING TO GAT TABLE ASSIGNMENTS:
295 '           255(FF)= BOTH GRANS USED
296 '           254(FE)= #1 FREE, #2 USED
297 '           253(FD)= #1 USED, #2 FREE
298 '           252(FC)= BOTH GRANS FREE
299 '   PRINT 'ASSIGNED' AND 'FREE' TOTALS.
3ØØ FORX=1TO4Ø
31Ø Z=PEEK(A+X)
32Ø IFZ=255THEN37Ø
33Ø IFZ=254THENY(1,1,X)=1:R=R+1:GOTO37Ø
34Ø IFZ=253THENY(2,1,X)=1:R=R+1:GOTO37Ø
35Ø IFZ=252THENY(1,1,X)=1:Y(2,1,X)=1:R=R+2:GOTO37Ø
36Ø PRINT"DIRECTORY ERROR - USE DIRCHECK/CMD":END
37Ø NEXT
38Ø LPRINT"GRANULES -   ASSIGNED ";
39Ø LPRINTUSINGMA$;(T*2)-R;:LPRINTSTRING$(1Ø," ");"FREE
";USINGMA$;R;
395 '   A=FIRST BYTE OF SECTOR 2 (B2ØØH).  CHECK ALL FPDE'S
FOR FXDE'S.  BYTE Ø MUST INDICATE IT IS NOT ITSELF AN FXDE,
BUT THE SECOND LAST BYTE MUST SHOW IT HAS ONE (FE).  IF YES,
HANDLE IN SUBROUTINE.    2ND BYTE ZEROED TO FACILITATE
NEWDOS8Ø-V2
4ØØ A=-19968:PRINT@4Ø8,"- FXDE CHECK -"
41Ø FORX=ØTO63:POKEA+(X*32)+1,Ø:POKEA+(X*32)+2,Ø
42Ø IF PEEK(A+(X*32))>0 AND PEEK(A+(X*32))<128 AND
PEEK(A+(X*32)+3Ø)=254 THEN GOSUB 1Ø6Ø
43Ø NEXTX
435 '   PRINT NUMBER OF FXDE'S.
44Ø LPRINTSTRING$(1Ø," ");"FXDE'S -";USINGMA$;XD
45Ø LPRINT"SECTORS -   ASSIGNED ";:LPRINTUSINGMA$;((T*2)-R)*5;:
LPRINTSTRING$(1Ø," ");"FREE
";USINGMA$;R*5;
455 '   CHECK ALL FPDE'S.  IF ACTIVE, GO AND HANDLE IN
SUBROUTINE.
46Ø FORX=ØTO63
47Ø IFPEEK(A+(X*32))=ØTHEN49Ø
48Ø GOSUB73Ø
49Ø PRINT@4Ø8,"COUNTDOWN ";64-X:NEXTX:CLS
495 '   PRINT NUMBER OF FPDE'S.
5ØØ LPRINTSTRING$(1Ø," ");"FPDE'S -";USINGMA$;FP
5Ø5 '   PRINT PAGE TITLE.
51Ø LPRINT:LPRINT"TRACK>    <1ST GRANULE>          <2ND
GRANULE>        <FILE LENGTH>":LPRINT"DC HX>";"----<SECTORS
Ø-4>";STRING$(1Ø,"-");"<SECTORS
5-9>";STRING$(4,"-");"<SECTORS/BYTES-EXTENTS": LPRINT
STRING$(7 2,"*")
513 '
514 '********** PRINTING ROUTINE **********
515 '
516 '   START OF 4Ø TRACK PRINTOUT.
52Ø FORX=1TO4Ø:C$=CHR$(X-1)
525 '   CONVERT TRACK NUMBER TO HEX.  PRINT BOTH.
53Ø LPRINTUSING"##";X-1;:LPRINT" ";:LPRINTMID$(H$,
((ASC(C$)AND24Ø)/16)+1,1);MID$(H$,(ASC(C$)AN D15)+1,1);
535 '   2-UP PRINTING OF GRANS 1 AND 2.
536 '   CHECK CODES- Ø=USED, 1=FREE, 2=LOCKED OUT.
54Ø FORW=1 TO 2
55Ø IFY(W,1,X)=1THENYN$(W,X)=" -- FREE -- "
56Ø IFY(W,1,X)=2THENYN$(W,X)=" LOCKED OUT "
565 '   NUMBER THIS GRAN.
57Ø LPRINT" <";USING"##";Y(W,2,X);
575 '   NAME OF FILE AND TOTAL GRANS IT HAS.
58Ø LPRINT"  ";YN$(W,X);" ";USING"##";Y(W,3,X);
59Ø LPRINT"> ";
6ØØ NEXTW
6Ø5 '   IS THERE A FILE TO LIST FOR 'LENGTH' COLUMN?
61Ø NS=NS+1
62Ø IFSN$(NS)=""THENLPRINT:GOTO7ØØ
63Ø LPRINT;
635 '   PRINT NAME WITHOUT BLANKS.
64Ø FORJ=1TO12
65Ø IFMID$(SN$(NS),J,1)=" "THEN67Ø
66Ø LPRINTMID$(SN$(NS),J,1);
67Ø NEXTJ
675 '  PRINT NUMBER OF COMPLETE SECTORS, BYTES LEFT OVER, AND
EXTENTS.
```

```
68Ø IFSN(1,NS)<>ØTHENSN(2,NS)=SN(2,NS)-1
69Ø LPRINT STR$(SN(2,NS));"/"; RIGHT$(STR$(SN(1,NS)),
LEN(STR$(SN(1,NS)))-1);"-";RIGHT$(STR$(SN(3,NS)),
LEN(STR$(SN(3,NS)))-1)
7ØØ NEXTX
7Ø5 '   PROVISION FOR THERE BEING MORE THAN 4Ø FPDE'S.
71Ø IFNF>NSTHENLPRINT"DISK CONTAINS > 4Ø ENTRIES - RUN
DIRCHECK FOR COMPLETE SECTOR/BYTE COUNT LIST"
715 '   FINISHED - RETURN TO START.
72Ø LPRINTSTRING$(16," ");"COPYRIGHT (C) 1981 BY MACSWAIN
ENTERPRISES":GOTO12Ø
721 '
722 '
723 '
724 '********** FILE PRIMARY DIRECTORY ENTRY (FPDE)
**********
725 '
727 '   (UPON ENTRY A=BYTE Ø, SECTOR 2, AND X=REL NO OF FPDE
(Ø-63).
728 '   GET NAME OF FILE IN N2$.  ADD ONE TO FPDE COUNT.
73Ø N2$="":FORS=5TO12
74Ø N1$=CHR$(PEEK(A+(X*32)+S))
75Ø N2$=N2$+N1$
76Ø NEXTS:IFPEEK(A+(X*32))<128THENFP=FP+1
765 '   IS THERE A FILE NAME EXTENSION?
77Ø IFPEEK(A+(X*32)+13)>32THENN2$=N2$+CHR$(47)ELSEN2$="
 "+N2$+"  ":GOTO81Ø
775 '   ADD FILE EXTENSION TO NAME.
78Ø FORS=13TO15
79Ø N2$=N2$+CHR$(PEEK(A+(X*32)+S))
8ØØ NEXTS
8Ø5 '   GET NO OF SECTORS IN SEC.
81Ø SEC=(PEEK(A+(X*32)+21)*256)+PEEK(A+(X*32)+2Ø)
815 '   BYPASS ALL SYSTEM AND INVISIBLE FILES ON 'LENGTH'
LISTING?
82Ø IFY$="*"ANDPEEK(A+(X*32))>16THEN85Ø
825 '   BUILD 'FILE LENGTH' ARRAY SN$(NAME), SN(SECTORS,
BYTES, EXTENTS).
83Ø NF=NF+1:SN$(NF)=N2$:SN(1,NF)=PEEK(A+(X*32)+3):SN(2,NF)=SEC
835 '   IF AN FXDE FOR 'LENGTH' LISTING, SHOW AS SUCH.
84Ø IFPEEK(A+(X*32))>127THENSN$(NF)=" -- FXDE -- ": POKE
A+(X*32)+1,Ø
845 '   CALCULATE TOTAL GRANS AND SECTORS ASSIGNED TO FILE.
85Ø GR=INT((SEC/5)+.8):SEC=5*GR
855 '   GRANUAL COUNT 'SO FAR' IS PICKED OFF THIS NORMALLY
UNUSED (AND ZERO) BYTE.  ONLY FXDE ENTRIES WE HAVE PREVIOUSLY
HANDLED WILL CONTAIN SIGNIFICANT AMOUNTS.  Z1=NUMBER OF
EXTENTS FROM FXDE'S (IF ANY).
86Ø GC=PEEK(A+(X*32)+2)+1:Z1=PEEK(A+(X*32)+1)
865 '   START OF CHECKING POTENTIAL 5 EXTENTS.
866 '   LOAD SN ARRAY - SEE LINE 815.
87Ø FORS=22TO3ØSTEP2:IFY$=""ANDPEEK(A+(X*32))>16 THEN 88Ø
ELSE SN(3,N F)=Z1
875 '   TR=TRACK NUMBER.
88Ø TR=PEEK(A+(X*32)+S)
885 '   GN=NUMBER OF CONTIGUOUS GRANS THIS EXTENT (MINUS
ONE).
89Ø GN=PEEK(A+(X*32)+S+1)
895 '   IS THIS EXTENT AN FXDE POINTER?
9ØØ IFTR=254THENRETURN
9Ø5 '   THIS INDICATES PREVIOUS ONE WAS THE LAST?
91Ø IFTR=255ANDGN=255THENRETURN
914 '   ADD 1 TO EXTENT COUNT Z1.
915 '   LL=1ST OR 2ND GRAN ON TRACK.  IS THIS ONE THE 1ST?
92Ø Z1=Z1+1:LL=1:IFGN<32THEN94Ø
925 '   2ND GRAN.  STRIP HIGH ORDER BITS.
93Ø GN=GN-32:LL=2
935 '   ADJUST GRAN COUNT TO ACTUAL NUMBER.
94Ø GN=GN+1
945 '   GP=GRAN CONTROL.
95Ø GP=1
955 '   MOVE NAME TO ARRAY.
96Ø YN$(LL,TR+1)=N2$
965 '   MOVE TOTAL GRANS TOO.
97Ø Y(LL,3,TR+1)=GR
975 '   IF THIS AN FXDE, PICK UP CORRECT TOTAL GRANS FROM
NORMALLY UNUSED BYTE 4.
98Ø IFGR=ØTHENY(LL,3,TR+1)=PEEK(A+(X*32)+4)
985 '   MOVE 'SO FAR' GRAN COUNT TO ARRAY.
99Ø Y(LL,2,TR+1)=GC
995 '   INCREMENT 'SO FAR' GRAN COUNT.
1ØØØ GC=GC+1
1ØØ5 '  TOTAL GRANS THIS EXTENT AND NUMBER ASSIGNED TO ARRAY
EQUAL?
1Ø1Ø IFGN=GPTHENNEXTS
1Ø15 '  INCREMENT GRAN CONTROL.
1Ø2Ø GP=GP+1
1Ø25 '  GRAN 1ST OR 2ND TO ARRAY?
1Ø3Ø IFLL=1THENLL=2:GOTO96Ø
1Ø35 '  IT WAS 2.  RESET LL AND INCREMENT TRACK NUMBER.
1Ø4Ø LL=1:TR=TR+1
1Ø5Ø GOTO96Ø
1Ø53 '
1Ø54 '********** FILE EXTENSION DIRECTORY ENTRY (FXDE)
**********
1Ø55 '
1Ø56 '  (UPON ENTRY, A AND X ARE SAME AS IN LINE 727).
1Ø57 '  MOVE FILE NAME (COMPLETE) TO N2$.  Z3=ADDRESS OF FPDE
(BYTE 1) EXTENT COUNTER.
1Ø6Ø Z3=A+(X*32)+1:N2$="":FORS=5TO15
1Ø7Ø N1$=CHR$(PEEK(A+(X*32)+S))
1Ø8Ø N2$=N2$+N1$:NEXTS
1Ø85 '  CALCULATE TOTAL GRANS ASSIGNED TO FILE.
1Ø9Ø SEC=(PEEK(A+(X*32)+21)*256)+PEEK(A+(X*32)+2Ø)
11ØØ GR=INT((SEC/5)+.8)
1105 '  Q=FIRST BYTE OF FPDE AND SUBSEQUENT FXDE(S).
111Ø Q=A+(X*32)
112Ø GOTO122Ø
1125 '  AN FPDE (OR FXDE) WAS JUST HANDLED THAT HAD AN FXDE
POINTER. N2$ HAS FILE NAME, TC CONTAINS TOTAL GRANS 'SO FAR',
AND GR IS TOTAL GRANS FOR FILE.
1126 '  TR CONTAINS '254' AND GN HAS THE POINTER TO NEXT FXDE.
CALCULATE THE NEW ADDRESS OF THE FIRST BYTE AND PUT IT IN Q.
```

```
1130 G1=INT(GN/16)
1140 GN=GN-(G1*16)
1150 Q=A+(GN*256)+(G1*16)
1155 '  PUT TOTAL GRANS 'SO FAR' IN REL BYTE 2.
1160 POKEQ+2,TC
1165 '  PUT TOTAL GRANS IN REL BYTE 4.
1170 POKEQ+4,GR
1175 '  PUT FILE NAME IN REL BYTES 5 TO 15 FROM N2$.
1180 FORN3=0TO10
1190 POKEQ+5+N3,ASC(MID$(N2$,N3+1,1))
1200 NEXTN3
1205 '  XD=TOTAL NUMBER OF FXDE'S ON DISKETTE.
1210 XD=XD+1
1215 '  LOOK AT THE EXTENTS- GET NO OF GRANS 'SO FAR'.
1220 TC=PEEK(Q+2)
1225 '  POTENTIAL 4 EXTENTS + A POINTER (OR FILE END).
1230 FORS=22TO30STEP2
1235 '  GET TRACK AND NO OF CONTIGUOUS GRANS.
1236 '  PUT NUMBER OF EXTENTS (LESS 4 FOR FPDE ITSELF) IN
FPDE COUNTER (BYTE 1).
1240 TR=PEEK(Q+S):GN=PEEK(Q+S+1):IFZ2>3THENPOKEZ3,Z2-4
1245 '  IS THIS EXTENT AN FXDE POINTER?
1250 IFTR=254THEN1130
1255 '  WAS LAST EXTENT THE FINAL ONE THIS FILE?
1260 IFTR=255ANDGN=255THENRETURN
1265 '  STRIP HIGH ORDER BITS, GET ACTUAL GRAN COUNT & ADD
TO TOTAL 'SO FAR' COUNT. GO TO NEXT EXTENT.  INCREMENT EXTENT
COUNT Z2.
1270 Z2=Z2+1:IFGN>31THENGN=GN-32
1280 TC=TC+GN+1:NEXTS
1285 '  MACHINE LANGUAGE PROGRAM READS DIRECTORY (TRACK 11H,
SECTORS 0-9) INTO MEMORY, STARTING AT B000H.
1286 '  IF A DISK ERROR OCCURS, A MESSAGE WILL BE PRINTED &
THE PROGRAM HALTS.  PRESSING ANY KEY WILL CAUSE ANOTHER READ
ATTEMPT.
1290 DATA 243,62,1,50,225,55,33,236,55,54,3,1,0,0,205,96,0,
203,70,32,252,205,201,1,14,10,33,0,176,17,0,17
1300 DATA 205,61,160,40,6,13,200,28,36,24,245,33,124,160,6,
12,126,44,205,51,0,16,249,205,73,0,195,0,160,205,65
1310 DATA 160,200,197,213,229,62,1,50,225,55,237,83,238,55,
33,236,55,54,19,197,193,197,193,203,70,32,252,54,136
1320 DATA 193,197,193,197,17,239,55,24,13,230,149,61,32,10,
126,203,79,40,246,26,2,3,24,246,175,182,54,208
1330 DATA 225,209,193,201,13,68,73,83,75,32,69,82,82,79,82,0
```

W. G. MacSwain
MacSwain Enterprises
151 Cass Avenue
Agincourt, Ontario
Canada M1T 2B5 ■

PROGRAM CONVERSION

continued from page 56

and a numeric portion. Therefore, this line could be properly converted as follows:

```
20 OPEN "O",1,"WEEKLY"+STR$(GNO)

PRINT D$;"WRITE WEEKLY";GNO
PRINT D$:PRINT #1;"WRITE WEEKLY";GNO
```

By now we know to ignore D$'s and to treat numerics as strings when converting APPLE disk statements to the TRS-80. However, here we have an exception. The WRITE command on the TRS-80 is part of the OPEN command, since a file must always be defined as an input file or as an output file. Therefore, WRITE statements on APPLE programs may be deleted on TRS-80 programs, as long as the READ or WRITE is incorporated into an OPEN statement. In some cases, it may be necessary to CLOSE and re-OPEN a file on the TRS-80, whereas on the APPLE a new OPEN statement would be necessary.

CONCLUSION — LOOKING TOWARDS THE FUTURE

This concludes this month's tips on program conversion. As always, I am open to feedback of any type concerning the MODELS I/II/III, APPLE, PET/CBM, IBM, ATARI, and CP/M. In future months, I will expand upon the topics covered over the last 9 issues. At present, I expect to concentrate on the APPLE, PET, and IBM computers, but this may change depending upon reader interest.

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

Form **1040A** Department of the Treasury—Internal Revenue Service **U.S. Individual Income Tax Return** **1982** OMB No. 1545-0085

Use IRS label. Otherwise, please print or type.	Your first name and initial (if joint return, also give spouse's name and initial)	Last name	Spouse's social security no.
	Present home address (Number and street, including apartment number, or rural route)		
	City, town or post office, State and ZIP code	Your occupation ▶ / Spouse's occupation ▶	

Presidential Election Campaign ▶ Do you want $1 to go to this fund? Yes No / If joint return, does your spouse want $1 to go to this fund? Yes No — **Note:** *Checking "Yes" will not increase your tax or reduce your refund.*

H&E COMPUTRONICS INC.

● ● EVERYTHING FOR YOUR TRS-80™ ● ATARI™ ● APPLE™ ● PET™ ● CP/M™ ● XEROX™ ● IBM™ ● ●

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. - * ATARI is a trademark of Atari Inc. - * APPLE is a trademark of Apple Corp. - * PET is a trademark of Commodore
* CP/M is a trademark of Digital Research - *XEROX is a trademark of Xerox Corp. - * IBM is a trademark of IBM Corp.

BUSINESS PAC 100

100 Ready-To-Run Business Programs

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software

(ON CASSETTE OR DISKETTE).....Includes 128 Page Users Manual.....
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

	NAME	DESCRIPTION
1	RULE78	Interest Apportionment by Rule of the 78's
2	ANNUI	Annuity computation program
3	DATE	Time between dates
4	DAYYEAR	Day of year a particular date falls on
5	LEASEINT	Interest rate on lease
6	BREAKEVN	Breakeven analysis
7	DEPRSL	Straightline depreciation
8	DEPRSY	Sum of the digits depreciation
9	DEPRDB	Declining balance depreciation
10	DEPRDDB	Double declining balance depreciation
11	TAXDEP	Cash flow vs. depreciation tables
12	CHECK2	Prints NEBS checks along with daily register
13	CHECKBK1	Checkbook maintenance program
14	MORTGAGE/A	Mortgage amortization table
15	MULTMON	Computes time needed for money to double, triple, etc.
16	SALVAGE	Determines salvage value of an investment
17	RRVARIN	Rate of return on investment with variable inflows
18	RRCONST	Rate of return on investment with constant inflows
19	EFFECT	Effective interest rate of a loan
20	FVAL	Future value of an investment (compound interest)
21	PVAL	Present value of a future amount
22	LOANPAY	Amount of payment on a loan
23	REGWITH	Equal withdrawals from investment to leave 0 over
24	SIMPDISK	Simple discount analysis
25	DATEVAL	Equivalent & nonequivalent dated values for oblig.
26	ANNUDEF	Present value of deferred annuities
27	MARKUP	% Markup analysis for items
28	SINKFUND	Sinking fund amortization program
29	BONDVAL	Value of a bond
30	DEPLETE	Depletion analysis
31	BLACKSH	Black Scholes options analysis
32	STOCVAL1	Expected return on stock via discounts dividends
33	WARVAL	Value of a warrant
34	BONDVAL2	Value of a bond
35	EPSEST	Estimate of future earnings per share for company
36	BETAALPH	Computes alpha and beta variables for stock
37	SHARPE1	Portfolio selection model-i.e. what stocks to hold
38	OPTWRITE	Option writing computations
39	RTVAL	Value of a right
40	EXPVAL	Expected value analysis
41	BAYES	Bayesian decisions
42	VALPRINF	Value of perfect information
43	VALADINF	Value of additional information
44	UTILITY	Derives utility function
45	SIMPLEX	Linear programming solution by simplex method
46	TRANS	Transportation method for linear programming
47	EOQ	Economic order quantity inventory model
48	QUEUE1	Single server queueing (waiting line) model
49	CVP	Cost-volume-profit analysis
50	CONDPROF	Conditional profit tables
51	OPTLOSS	Opportunity loss tables
52	FQUOQ	Fixed quantity economic order quantity model
53	FQEOWSH	As above but with shortages permitted
54	FQEOQPB	As above but with quantity price breaks
55	QUEUECB	Cost-benefit waiting line analysis
56	NCFANAL	Net cash-flow analysis for simple investment
57	PROFIND	Profitability index of a project
58	CAP1	Cap. Asset Pr. Model analysis of project
59	WACC	Weighted average cost of capital
60	COMPBAL	True rate on loan with compensating bal. required
61	DISCBAL	True rate on discounted loan
62	MERGANAL	Merger analysis computations
63	FINRAT	Financial ratios for a firm
64	NPV	Net present value of project
65	PRINDLAS	Laspeyres price index
66	PRINDPA	Paasche price index
67	SEASIND	Constructs seasonal quantity indices for company
68	TIMETR	Time series analysis linear trend
69	TIMEMOV	Time series analysis moving average trend
70	FUPRINF	Future price estimation with inflation
71	MAILPAC	Mailing list system
72	LETWRT	Letter writing system-links with MAILPAC
73	SORT3	Sorts list of names
74	LABEL1	Shipping label maker
75	LABEL2	Name label maker
76	BUSBUD	DOME business bookkeeping system
77	TIMECLCK	Computes weeks total hours from timeclock info.
78	ACCTPAY	In memory accounts payable system-storage permitted
79	INVOICE	Generate invoice on screen and print on printer
80	INVENT2	In memory inventory control system
81	TELDIR	Computerized telephone directory
82	TIMUSAN	Time use analysis
83	ASSIGN	Use of assignment algorithm for optimal job assign.
84	ACCTREC	In memory accounts receivable system-storage ok
85	TERMSPAY	Compares 3 methods of repayment of loans
86	PAYNET	Computes gross pay required for given net
87	SELLPR	Computes selling price for given after tax amount
88	ARBCOMP	Arbitrage computations
89	DEPRSF	Sinking fund depreciation
90	UPSZONE	Finds UPS zones from zip code
91	ENVELOPE	Types envelope including return address
92	AUTOEXP	Automobile expense analysis
93	INSFILE	Insurance policy file
94	PAYROLL2	In memory payroll system
95	DILANAL	Dilution analysis
96	LOANAFFD	Loan amount a borrower can afford
97	RENTPRCH	Purchase price for rental property
98	SALELEAS	Sale-leaseback analysis
99	RRCONVBD	Investor's rate of return on convertable bond
100	PORTVAL9	Stock market portfolio storage-valuation program

☐ **CASSETTE VERSION** **$99.95**
☐ **DISKETTE VERSION** **$99.95**
☐ **TRS-80* MODEL II VERSION** **$149.95**

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

AMERICAN EXPRESS · VISA · MASTER CHARGE

24 HOUR ORDER LINE
(914) 425-1535

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

ASK FOR OUR 64-PAGE CATALOG

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

H&E COMPUTRONICS INC.

WABASH WARRANTY FLEXIBLE DISKS

SPECIAL INTRODUCTORY OFFER

5

2

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

800-431-2818 24-HOUR HOTLINE **914-425-1535**

H&E 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 ☐ ☐

Credit Card No. _____ Exp. _____

Name _____

Address _____

City/State/Zip _____

Phone (____) _____

The Original Magazine for Owners of the TRS-80™ MicroComputer

MODEL I • MODEL II/16 • 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/16 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

VISA

☐ One Year Magazine Subscription $24 ☐ New ☐ Renewal
☐ Two Year Magazine Subscription $48.00 ☐ New ☐ Renewal
☐ Sample Issue $4. ☐ Mod II/16 Newsletter Subscription $18

Your Choice: TRS-80* at Your Fingertips ☐ or Word Processor/Data Management ☐
Model ☐ Model II/16 ☐ Model III ☐ Color Computer ☐ Pocket Computer ☐

NEW! NEW! **MOD-II/16 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.

ADVERTISING DIRECTORY

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • XEROX™

The VERSABUSINESS™ Series

Each VERSABUSINESS module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every VERSABUSINESS™ module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any VERSABUSINESS™ module, you may return it within 30 days for a refund. Manuals for any VERSABUSINESS™ module may be purchased for $25 each, credited toward a later purchase of that module.

To Order:

Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

- add $3 for shipping in U.P.S. areas
- add $4 for C.O.D. or non-U.P.S. areas
- add $5 to CANADA or MEXICO
- add proper postage elsewhere

DEALER INQUIRIES WELCOME

All prices and specifications subject to change / Delivery subject to availability.

COMPUTRONICS, INC.

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. *APPLE is a trademark of Apple Corp. *IBM is a trademark of IBM Corp. *OSBORNE is a trademark of Osborne Corp. *CP/M is a trademark of Digital Research *XEROX is a trademark of Xerox Corp.

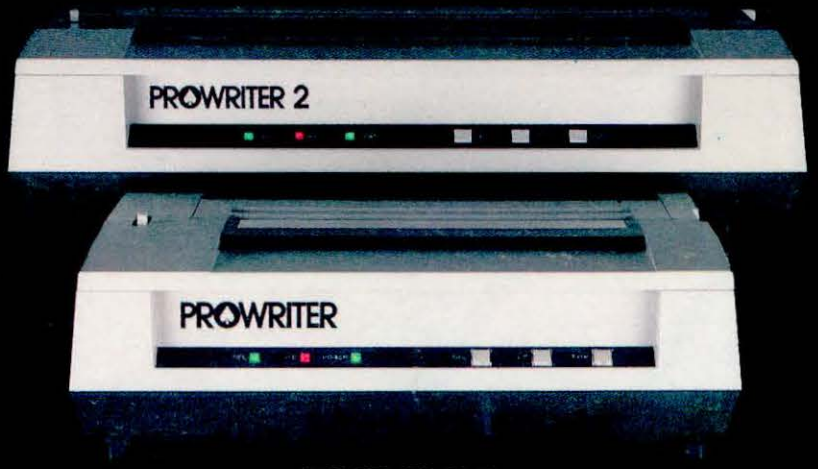
THE LEADING EDGE IN PRINTERS

ONE GREAT LINE. ONE GREAT WARRANTY.

Finally, there's one full family of printers that covers every business or word processing application—all from C. Itoh, a company known for packing more product into less price; and all distributed exclusively by Leading Edge, a company known for searching out and providing that very thing. Which means that one call to one source can get you any printer, any time you need it, for any purpose. All backed by a full years' warranty from Leading Edge. (Try *that* on any other line of printers.)

THE PRO'S.

The Prowriters: business printers—and more. The "more" is a dot-matrix process with more dots. It gives you denser, correspondence quality copy (as opposed to business quality copy, which looks like a bad job of spray-painting).
Prowriter: 120 cps. 80 columns dot matrix compressable to 136. 10" carriage. Parallel or serial interface.
Prowriter 2: Same as Prowriter, except 15" carriage allows full 136 columns in normal print mode. Parallel or serial interface.



THE STAR.

The Starwriter F-10. In short (or more precisely, in a sleek 6" high, 30-pound unit), it gives you more of just about everything—except bulk and noise—than any other printer in its price range. It's a 40 cps letter-quality daisy-wheel with a bunch of built-in functions to simplify and speed up word processing. It plugs into almost any micro on the market, serial or parallel.

THE MASTER.

The Printmaster F-10. Does all the same good stuff as the Starwriter except, at 55 cps, the Master does it faster.

Distributed Exclusively by Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021. Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.