MARCH 1983
ISSUE NUMBER 55

PRICE U.S. $2.95
U.K. £1.25

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 & RETURN POSTAGE GUARANTEED

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. ****** $1000 ******

More than $1000 worth of valuable coupons for free merchandise, discount software, hardware at a fraction over cost and computer supplies.

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

master charge VISA AMERICAN EXPRESS Card

☐ 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: $1000 offer(see box A) ☐ or Word Processor/Data Management ☐
Model I ☐ 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

PUBLISHER
Howard Y. Gosman

BUSINESS MANAGER
Steven M. Kahan

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

BUSINESS EDITOR
Peter Shenkin

MANAGING EDITOR
Martin Leffler

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

ADVERTISING DIRECTOR
Kevin Rushalko

SALES MANAGERS
Valerie Furci
Sheryl Prevot

DEALER SALES MANAGER
Janet Lasher

ART DIRECTOR
Edmund Khaleel

OFFICE MANAGER
Beatrice Kahn

SOFTWARE MANAGER
Darlene Bell

CUSTOMER SERVICE
Robert Williams

INVENTORY CONTROL
Michael Wiseltier

SHIPPING
Joan Gentry
Al Pizzo

PRODUCT DEVELOPMENT
Steven Kaplan
David Staub

PRODUCTION
Adele Damiano
Louis Wetstein

MARKETING MANAGER
Andrew Hofer

PROGRAMMING MANAGER
Roy Flynt
Nancy Rhodes

MARCH 1983 **ISSUE NUMBER 55**

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

HOME ROBOTICS

Heath/Zenith has introduced the first completely self-contained mobile robot for students and hobbyists. The HERO 1 looks something like R2D2, and is equipped with a surprisingly wide array of senses to help it interact with the world. The HERO 1 can detect sound, light, and motion, and has an ultrasonic obstacle detector — a ranging device that has a resolution from 1/4 inch to 8 feet. You can probably program the HERO 1 to answer the door for you — it can speak in complete sentences with its own fully programmable speech synthesizer.

Using an arm capable of seven axes of motion, the robot can be programmed to pick up and manipulate small objects (up to 1 lb.). The entire head of the robot (with arm) rotates 360 degrees, like a tank turret. The arm has a shoulder rotation of 150 degrees, arm extension of 5 inches, wrist pivot of 180 degrees and wrist rotation of 360 degrees. The gripper opens to 3 1/2 inches and rotates 90 degrees.

Heathkit's intentions lie beyond providing a great new toy for people with money to burn. The HERO 1 comes with a 1200-page companion course, "Robotics and Industrial Electronics," which provides students with a practical grasp of robot technologies, including industrial electronics, mechanics, computer theory and robotics programming. Robots will play an ever-increasing role in industry, and those who start working with robots now will be getting a big head start on what is becoming one of the really important industries for the future.

The Heathkit robot is in production now, and is available for $1500 in kit form, or $2500 fully assembled.

PUBLISHED ABSTRACTS

Two new publications have started recently, both of which publish abstracts (brief summaries) of many recent articles related to microcomputing.

continued on page 7

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

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

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

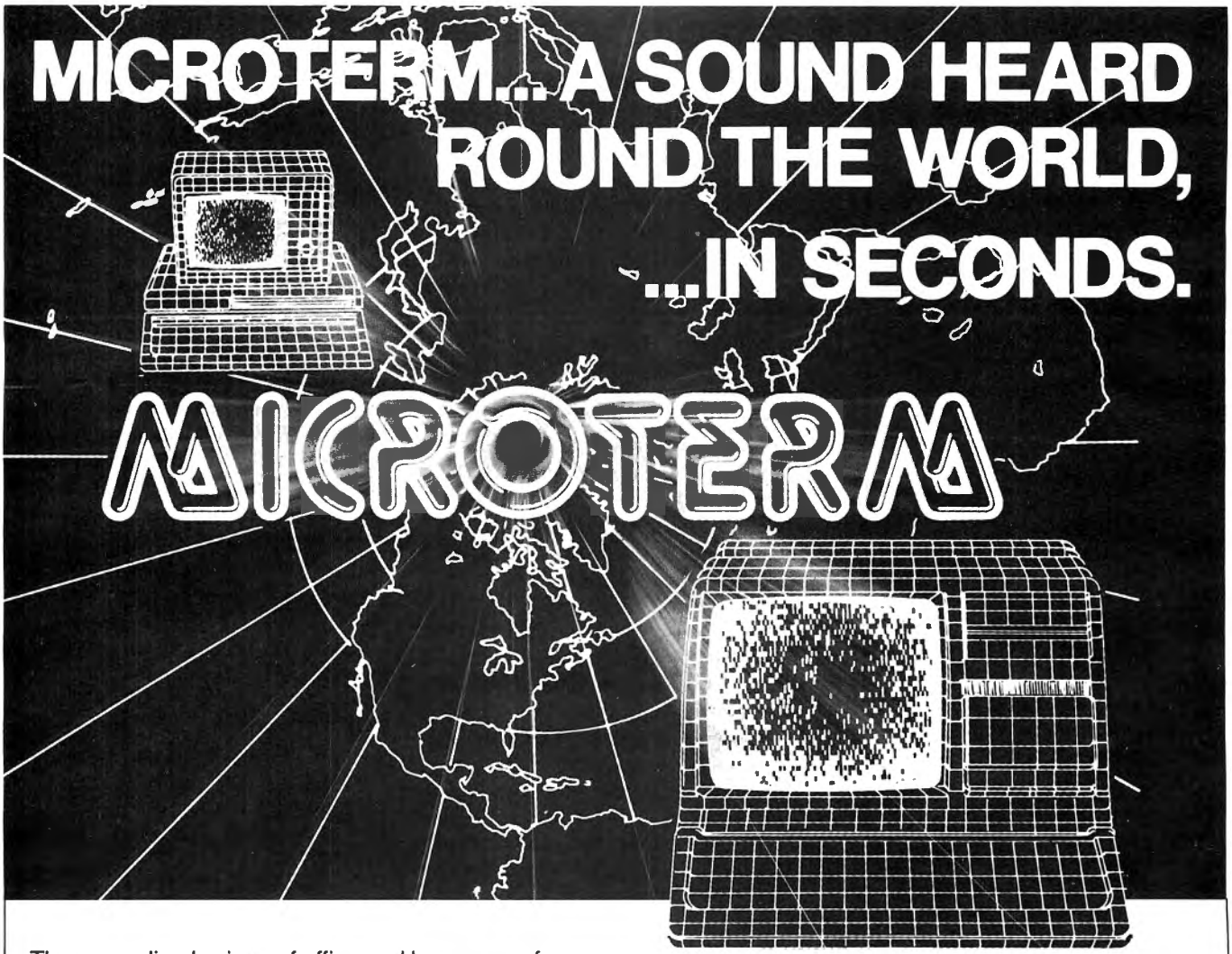
The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE is typeset by Photonics, Ltd., 188 Highwood Ave., Tenafly, NJ 07670, and is printed by Kay Offset Printing Service, Inc., 154 Grand Street, New York, NY 10013.

SUBSCRIPTION RATES

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

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

YOUR SUBSCRIPTION HAS EXPIRED IF . . . THE NUMBER ABOVE YOUR NAME AFTER THE DASH ON YOUR MAILING LABEL IS 55 (OR LESS). THE NUMBER FOLLOWING THE DASH TELLS YOU THE LAST ISSUE THAT YOU WILL RECEIVE. For example, if your subscription number is 16429-55, your subscription expires with this issue (issue #55).



The expanding horizon of office and home use of new communications and data services tying smart terminals into networks through telecommunications links makes the world need a high speed terminal program.

Busy computers shouldn't have to wait for data simply because an old fashioned link can't handle today and tomorrow's telecommunications. Slow terminal telecommunications can stand the smart operator's world on its ear.

Into today's fast paced world, Micro Systems Software presents MicroTerm, the first truly high speed terminal for this inquisitive world. Some computers can run at up to 4800 baud null-free (9600 in some direct-connect applications).

MicroTerm enables you to do more in less time, in both ASCII and the new "error-free" direct file mode. Its unique Macro-Key function allows you to have 10 user-defined keys that transmit up to 64 characters at a single stroke. You can even dial a phone number and transmit the buffer at a specified time completely unattended by the operator.

And while MicroTerm improves your computer's "ears" by outperforming any other telecommunications terminal program, its low price won't take a bite from your bit budget. It's only $79.95 retail.

You can't lose with MicroTerm's features, performance, price, documentation, or support.

It's the only terminal program enabling you to continue operations in the command mode while receiving additional data through the RS232 cable. And you can adjust video width, turn on the printer, open the buffer and do many other things and then return to the terminal mode without missing a thing.

Available for the TRS-80 Models I, II, III, 16, IBM PC, Zenith Z-100, and Apple II computers.

If these advantages are what you want in your world, communicate with your nearest MicroTerm dealer. For information contact: Micro Systems Software, Inc., 4301-18 Oak Circle, Boca Raton, Florida 33431, Telephone: 1-800-327-8724.

MICRO-SYSTEMS SOFTWARE, INC.

4301-18 Oak Circle, Boca Raton, Florida 33431, Telephone: 1-800-327-8724.

THE CRYSTAL BALL

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

BIG BLUE VS. BABY BELL

A potential new giant in the computer industry is about to awaken. As AT&T reorganizes its operations to accomodate its new independent offspring, arrangements are being made to bring their vast knowledge and experience in computer and networking technology to the computer market. A new corporation, nicknamed "Baby Bell" has just begun operations, with the objective of selling telephone and *computer* equipment to the public. Drawing on the expertise of AT&T's research and development, this could be the one company that can *really* give IBM a run for their money. AT&T has been using computers for a *long* time. In the United States, we have the largest and most efficient telephone system in the world. The kind of network-switching technology needed to keep a system like this up and running staggers the imagination. And the system *does* run well.

In case there's any question in your mind about the abilities of AT&T and associates, consider the history of their research. Ever heard of Bell Labs? This famous research laboratory has been the location of more world-shaking innovations and inventions than any other laboratory. Transistors and Lasers are just two examples of the kind of things they come up with at Bell Labs.

Baby Bell is bound to be successful when they start marketing small computers (or terminals) and network services — they have all the right connections. The kind of networking services that they will be able to offer will usher in the next big revolution in electronic banking, communications and information services. In a very short amount of time, say, within 2-3 years, you can expect to see this company take a *big* share of the market.

THE USER INTERFACE

A subject of much concern in computer design today is the "user interface," or how well a computer or program interacts with the user. But there may be a new meaning for "user interface" soon.

A microprocessor was recently used by a team of engineers and doctors to perform a fantastic new task: to help a paraplegic walk again — under computer control.

A young girl suffered a spinal column injury in a car accident. The result was permanent paralysis from the waist down — no movement and no sensation at all. The consequences of this type of injury are long lasting and very difficult for the victim to bear. The patient's mobility is permanently and severely limited. The patient's leg muscles will eventually atrophy away, causing terrible disfigurement as well as loss of mobility. Some patients never really recover from the depression resulting from their condition. Others can't bear to live with it at all.

Quite some time ago (about 13 years or more) some bright people realized that computers would eventually become far smaller than the giant mainframe machines in use back then. So they started working on their new technology right away, and now their work has become truly practical.

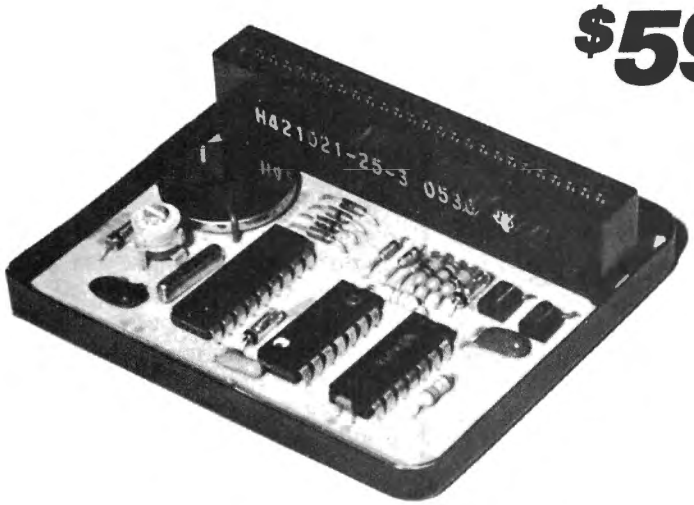
The result of their work is amazing. On nationwide TV, this girl, *totally paralyzed* below the waist, stood up and walked as if she were uninjured. She was wearing the prototype system designed to give mobility back to paralytics. Strapped to her legs and torso were some 28 sensors that feed data about the position and tension of many muscles in her legs and body. In addition, 14 electrical "stimulators" were held in place up and down both legs, capable of sending electrical impulses directly to her leg muscles.

When you lift your leg, many muscles in your back and abdomen must work along with your leg muscles. Moving your legs in different ways causes different kinds of tension and movement in these upper body muscles — *patterns* of tension and movement that can be sensed and *recognized* by a computer. Although this girl had no movement below the waist, the muscles in her upper body are unaffected. Thus, when she consciously tries to move a leg,

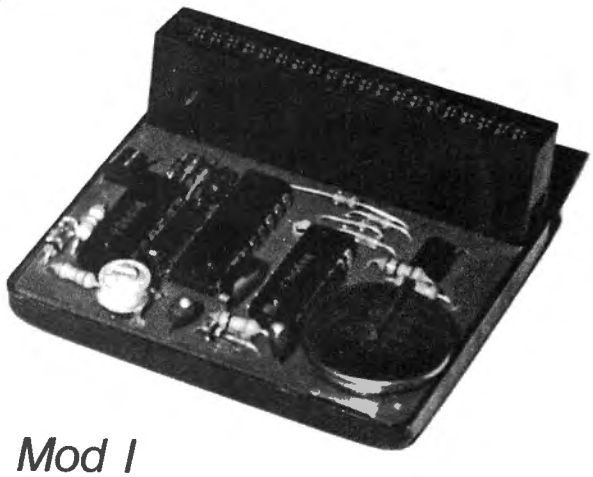
continued on page 6

NEWCLOCK·80

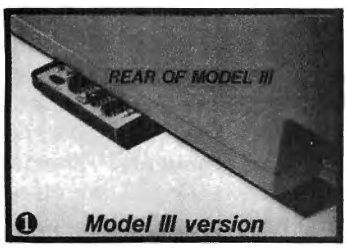
$59.95



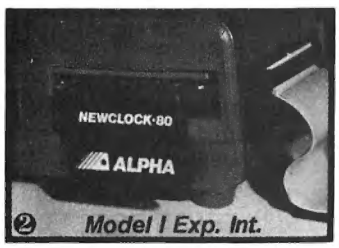
Mod III



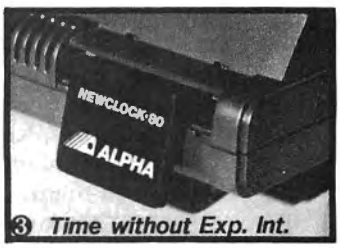
Mod I



❶ Model III version



❷ Model I Exp. Int.



❸ Time without Exp. Int.

❹ The complete package.

Wouldn't it be nice if your computer could always boot up with the right time and date and then stay accurate. Newclock-80 will enhance your Model I or III system with powerful clock/calendar/timer functions.

Using LSI (large scale integration) and custom circuits, Newclock-80 provides MO/DATE/YR, HR:MN:SEC plus AM/PM and day of week and even takes care of leap years! It continues to keep time and date with quartz accuracy when the computer is turned off or experiences a power failure. A single battery lasts over 2 years.

Compatibility: Newclock-80 is compatible with any operating system, including DOSPLUS, NEWDOS, LDOS. With its fully decoded circuitry it will work with any other hardware you may own. Bus expanders are available.

Installation is very simple, no tools, no disassembly, no soldering. Just plug it in, that's all. There is no power supply or messy cable. Newclock-80 plugs into the rear of the keyboard ❸ or side of the Exp. Int. ❷. Model III Newclock fits the 50 pin card edge (underneath) ❶

The Software: Newclock-80 is as easy to use as it is to install. -"SET", a Basic program, is used only once to set the time and date and select 12 or 24 hour format. -"TIMESTR", also in Basic, patches your computer "TIME$" function to read Newclock-80. It also adds "TIME$" to keyboard-only systems, a short routine is simply "poked" into low memory.

Newclock-80 uses 12 ports (176 to 188): 6 for the time, 6 for the date. The data is conveniently stored in decimal form, no conversion is needed. You can read or modify any digit using simple Basic "INP" and "OUT" statements.

No risk trial. Order your Newclock-80 today, see how easy it is to install and operate then decide within 30 days if you want to keep it. If for any reason you are not delighted with its quality and performance, you may return it for a prompt and courteous refund.

Your unit will come complete ❹ with software on tape, detailed instructions, handy reference card, and a 90 day warranty. Specify Model I or III. Software is also available on disk: add $5.
Lithium battery (not included) available from RADIO-SHACK (#23-162) or add $1.50 to your order.

Thanks to outstanding engineering and efficient manufacturing, ALPHA Products is once again able to offer a great product at a surprising price. Order your Newclock-80 at no obligation today.

Toll Free Order Line

800·221·0916

Orders Only. NY & Info call (212) 296-5916. Hours: 9-5 E.S.T.

ALPHA Products

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

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

MasterCard VISA

THE CRYSTAL BALL

continued from page 4

muscles in her upper body move in a particular pattern, the pattern is deciphered by the computer, and the computer sends a cooresponding pattern of impulses to the stimulators attached to her legs. The leg muscles then respond by contracting or relaxing — under computer control.

A dramatic videotape of the first attempt to use the system was shown on TV. After being attached to a network of cables, the girl approached a set of parallel bars in her wheelchair and was helped into a standing position, supporting herself with her arms. Then, with great difficulty, she tried to move her legs. Not much happened at first. The team of scientists had hoped for instant success, with the microcomputer already "knowing" how to walk. But the system was not quite that perfect, and besides, in several years since the accident, the girl had actually *forgotten* how to walk. But the girl was determined, and after some effort and experimentation, she was able to slide one leg forward, shift her weight onto it, and then slide the other leg forward — her first computer-assisted steps. Never underestimate the human mind's adaptability. She practiced hard, learning to move her legs, balance and stand, and finally, to walk unsupported. This is no easy task for the computer or a person. To simply stand still, balanced, the brain (or computer) must be constantly sending thousands of signals for imperceptible contractions and relaxations of hundreds of muscles in the legs and body.

By the time she was on TV, this girl could move as if she were uninjured. To see her lift her paralyzed leg to show the sensors attached was an inspiring sight. Just as important, the leg she lifted showed *no signs of atrophy,* — it was a well exercised leg, in good condition. In fact, a byproduct of this system will be a computer-assisted exercise clinic for paraplegics. A patient can sit down at a nautilus-type exercise machine, hook up to the computer, then sit back and relax as the computer makes the paralyzed legs exercise themselves.

The inventor of this system explained to the talk show audience that, 13 years ago, you would have to have a van drive around connected to you by a cable, with a large computer inside. One spectator asked about the size of the computer now used. The inventor took off his tie clip and held it forward for the camera. Mounted on the tie clip was one of his computers.

There has been much other speculation about the human-computer interface. At the present time, computers can communicate with humans in many different ways. You can talk with a computer via keyboard and video screen. You can talk directly to a computer with your voice, and it can talk directly back. Various sensors can be attached to your body to allow the computer to monitor all kinds of bodily functions.

But the ultimate user interface is still science fiction: the direct connection of computer circuits and the human nervous system. It's one thing to have a computer recognize a pattern of muscular movement and send motor impulses to the muscles, and another thing altogether to have the computer read and recognize nervous impulses directly. But as computer and neurological research both advance, someone will eventually be able to read patterns of information, and perhaps even thoughts, directly from patterns of nervous impulses. The scenario is this: someday, not too far away, technology will allow the installation of a microcomputer *within the human body.* A person would be able to simply *imagine* speaking to the computer (unconsciously "subvocalizing" the words), and the computer will recognize and interpret the patterns caused. Then the computer could speak back to the person, using speech synthesis and bone conduction, direct connection to auditory nerves, or even direct connections into the brain itself. As far back as the early 1960s, scientists forsaw the possibilities of this type of "implant computer". The results of a system like this are both fantastic and ominous. An implant computer would be able to transmit and receive data from large external computers. Imagine a worldwide telephone network where you need only to think of a person you wish to talk to, and the system puts you in direct contact with that person —

a sort of "electronic telepathy". Imagine an unlimited memory, a worldwide database, instantly accessible from inside your own head! Want to turn the lights on and off, or open and shut the garage door? Just *think* of it and it's done!

But what would happen to humanity? Would a worldwide system monitor our actions, or even our thoughts? If we can use a computer *now* to control leg muscles, how far are we from being able to control a human being? A recent article on this subject had the amusing title of "Man: The Ultimate Peripheral."

Getting even deeper into science fiction: how many people operate machinery, or drive a car, and have the feeling that the machine is an extension of their own body? If a few generations pass with people living with computers in their heads, they might feel that the computer network is an extension of their own minds — and in a sense it *will* be. What about artificial intelligence? Will our computers actually become minds in a symbiotic relationship with our own? Again there has been much speculation on this subject: humanity will become something other than human if biological and electronic intelligences eventually merge to become one mind.

Back down to earth: this kind of scenario may be a dream come true to some people and a nightmare to others. The hardware won't be available for a while, but you can see we may be headed in that direction. How many people can do fast arithmetic in their heads, or quickly do long division on a piece of paper? Not many can, not since handheld calculators became available. As computers can perform more and more of the tasks that we used to perform ourselves, we tend to become dependent on the computer and to lose the ability to do the task ourselves. We're not going to say that this is bad or good, but it does seem to be inevitable. In any case, we are definitely developing a more and more intimate relationship with our computers. How close will the relationship ultimately become? That remains to be seen. ■

BITS AND PIECES

continued from page 2

PEEK — The Journal of Micro Abstracts, is a monthly publication which allows the user to target his reading and organize his reference library by furnishing abbreviated abstracts of feature articles, hardware, software and book reviews, instructional material, etc., appearing in the current microcomputer literature. *PEEK* addresses the full range of subjects of interest to readers including business, utilities, equipment innovations, personal use and, occasionally, games. The combination of categorized abstracts and and a simple search system will allow convenient retrieval and allow readers to devote their attention to their micro systems. The sources abstracted are selected to be valuable to users of all popular micro systems. A 12-month subscription to *PEEK* costs $30 and is available from Herbert Skovronek, ed., *PEEK — the JMA,* Moraine Road, Morris Plains, NJ 07950.

The *Personal Computer Weekly News Abstracts* began publication on January 10th. This "Monday morning" eight-page newsletter reports on more than a hundred computer-related periodicals, 80 major daily newspapers and consumer publications, and news releases from over 3000 vendors. The *Abstracts* cover major metropolitan and national newspapers and periodicals such as *Time, Business Week, The New York Times, The Wall Street Journal, Money,* and many others who cover the personal computer field. The *Abstracts* will be mailed out Air Mail every Friday night in order to be on reader's desks on Monday morning if possible.

David E. Phillips, the editor, states that "for less than $2 a week, decision makers can now, in five to 15 minutes, decide where to look in which publications to find news relevant to their field, without sifting through the complete contents of all of these publications, both major and minor." According to Phillips, it takes over 200 hours of reading to compile each issue. The contents of each issue are cross-indexed for quick referral, and the newsletter is three-hole punched for filing in a binder supplied with each annual subscription.

The *Personal Computer Weekly News Abstracts* has a subscription

continued on page 8

THE CRYSTAL BALL
continued from page 7

price of $85 per year (including binder), and a special introductory subscription of $35 for 13 weeks (no binder included) will be available until March 31st, 1983. Contact: David E. Phillips, Editor, *Personal Computer Weekly News Abstracts*, 306 West 46th Street, New York, NY 10036; (212) 986-3333.

A PORTABLE MODEL III

A Fort Worth, Texas company has done what their neighbor, Radio Shack, would have been wise to do themselves. They have repackaged a Model III and turned it into a completely portable computer that looks like a cross between a Model III and an Osborne 1. If they market it successfully, the "Model III in a Briefcase" may come to be recognized as one of the most powerful portable computers in existence. It is, of course, completely software-compatible and disk-compatible with a standard Model III, which makes this a perfect "second computer" for present Model III owners — you can leave the desktop Model III at the office and take the portable home for the weekend, or out-of-town on a business trip.

The basic design of the "Model III in a Briefcase" is very similar to portable computers like the Osborne, Kaypro, and other "sewing machine" portables. The entire unit weighs 26 lbs., and is housed in an attractive and rugged brushed aluminum case (which will fit under a standard airline seat). A standard Model III keyboard is installed in the lid of the unit, which folds down to reveal the screen and disk drives. The disk drives are installed vertically, allowing room for a 9-inch CRT screen between them. The CRT is a slow-decay green phosphor screen. (An ordinary monitor screen flashes, or "strobes" at you 60 times each second, and this may be a significant cause of eyestrain if you use your computer for extended periods. The slow-fading green screen eliminates this problem.)

A few very important accessories are expected to be available soon, to be built right into the unit's housing. A built-in battery pack will make the computer *really* portable, so you'll be able to use it anywhere (an important feature for businesses such as contractors, surveyors, and others who may have job sites where power is not readily available). Another good option planned is a built-in auto-answer modem (a feature that will probably appear in many computers in the next few years). The most impressive add-on will be a built-in hard disk drive with a *removable* hard disk.

The "Model III in a Briefcase" currently sells for $2895, and you can get further information from Adcock & Johnson, P.O. Drawer 8778, Fort Worth, Texas 76112, (817) 429-5131.

ELECTRONIC ENCYCLOPEDIA

A computerized version of the *World Book Encyclopedia* is now available through the CompuServe Information Service. This project is intended to test the feasibility of this type of service, and enables computer users to search through a listing of more than 31,000 subject entries and retrive any part of the ten-million word text of the encyclopedia. Subscribers can select articles from a menu display that shows all articles whose titles begin with a key word you specify. A News Flashback feature covers current events and steers the reader to related encyclopedia articles, and a challenge feature tests the user's awareness of facts covered in various articles. The encyclopedia will be updated quarterly to insure that all information is current.

AUTOMATED ELECTRONIC MAIL

A new communications utility is available that has some of the most advanced features ever seen in a microcomputer terminal program. The ACEMAIL system was designed to support the auto-dial/auto-answer features of the Hayes Stack Smartmodem, and is capable of receiving or making calls, and sending or receiving data *completely unattended!* With the ACEMAIL system, it is possible to program your computer to place a call at any time and date (redialing as many times as necessary to get on line), and once logged on, it can send or receive whatever files you have programmed it to transfer — all *automatically.*

A disk file can be created to store commands for a long list of outgoing calls, including the date, time, phone number, transmission speed, prompt delay and names of files to

be transmitted. When the proper time comes, the program refers to the disk file, places the call, and makes the transmission.

The ACEMAIL system is available for 300 or 1200 baud Smartmodems, and the Smartmodems are also available from the same source. The ACEMAIL 1200 Software costs $119; the ACEMAIL 300 Software is priced at $79. The Hayes Stack Smartmodem, 300 and 1200 baud models, are available for $239 and $619 respectively. The ACEMAIL system was created by ACE Computer Products of Florida, Inc., 1640 N.W. 3rd Street, Deerfield Beach, FL 33441; voice phone: (305) 427-1257; data phone: (305) 427-6300.

MAKING MONEY WITH YOUR COMPUTER

Anyone who's ever wanted to start their own part-time business will appreciate this encouraging and realistic look at how a personal computer can become the basis of a successful home money-making enterprise. The authors give 33 innovative computer ideas ranging from lost pet services to roommate referrals, from direct mail advertising to personal investment analysis, from a unit comparison shopping service to novelty publications, from a computerized babysitting listing to church record-keeping, from bioshythm charts to computer music. And that's just the beginning for an imaginative self-starter.

We have seen at least three or four of these "using a computer to make money books" over the past several years. This seems to be the best of them and it is worthwhile reading if that's what you are setting out to do.

There are plenty of helpful hints on advertising services — including tips on how to write response-pulling ad copy, how to handle newspaper advertising, and tips on developing and keeping customers.

Making Money with Your Microcomputer was written by Robert J. Traister and Rich Ingram and published by TAB BOOKS INC., Blue Ridge Summit PA 17214 (717) 794-2191. The price is $7.95. (It's not available through H & E Computronics, Inc. but it is available through local computer book stores or by calling the telephone number above)

INFOWORLD ANNOUNCES ANNUAL HARDWARE AND SOFTWARE WINNERS

InfoWorld has announced the winners of its annual award for the most significant hard and software of 1982. The winner in the hardware department is the IBM PC. The winner in the software department is Microsoft's Multiplan (picked over VisiCalc).

Speaking of *InfoWorld*, if you don't subscribe, then you are missing the best of all of the computer publications. *InfoWorld*, is truly written for computer users (both new and experienced). It contains the most up-to-date computer information available anywhere to micro computer owners, and it comes out *weekly*. A subscription is $25 (and that's for 52 issues on a weekly basis). To subscribe, call toll-free 800-343-6464 or write to *InfoWorld*, Circulation Department, 375 Cochituate Road, Framingham MA 01701-9987 (VISA and MASTERCARD can be used).

THE SMARTBUFFER

Multiport SMARTBUFFER enhances systems with several computers and printers. Multichannel SMARTBUFFERS can support up to 4 inputs (computers, modems, etc.) and 4 outputs (printers, plotters, modems, etc.) at the same time. Several computers can rapidly transmit data to shared outputs, usually printers, simultaneously! The SMARTBUFFER'S expandable internal memory (up to 256K) keeps you and your computer(s) from having to wait. The computer(s) quickly dump their files to the SMARTBUFFER and are left free while the SMARTBUFFER drives the printer(s). The appropriate printer is selected at a flick of a switch or by software steering.

Word processors connect a high speed dot matrix printer and a correspondence quality printer to one or more computers. The operator can select a rough draft or final copy with front panel switches.

Accountants load different forms into different printers and the appropriate printer is selected by the program, eliminating the need to constantly change forms. You can cut checks on one printer and run a ledger on the other.

Corporate users can share one

continued pn page 17

PROGRAM PREVIEWS

A. A. Wicks

This Month: TRASHMAN

You are working some problems on your computer, using a program that requires various inputs from you. Suddenly, "nothing" happens. You press a key, then the space bar, and still there is no reaction from the computer. As you continue to press <Enter>, as suddenly as the computer stopped operating it begins to process again. You muse about the possibility that something is wrong with the computer or the program, and continue with your work. But a short while later, the same thing happens.

If this has happened to you and you don't know what causes it to happen, you are about to find out. If you already know the problem, perhaps you don't know that there is a 98% cure. If the problem is sufficient to annoy you or cause work processing problems, then the program to be reviewed offers a great amount of help.

Suppose you are using a program that utilizes a large number of strings in its makeup. In the processing the string will be changed. As an example, let us assume A$ is the string designated for a household inventory item. One moment, it is a "TV set." The next, it is a "Chair" that you input. When this change occurs, BASIC moves the string (be it words, names, numbers, etc.), to a new location in memory, and the old place is left vacant (for the moment).

Now, it will be easy to understand that in a short while (depending upon the amount of memory available), all memory will be used up, or, more specifically, "allocated," because there are all those empty holes now. BASIC now busies itself telling the computer to straighten out this mess, and to get these strings "compressed" in order to make more room. Using BASIC to do this task, it doesn't happen in a flash. In fact, depending upon how many strings are involved, the computer can "lock up" for anywhere between a few seconds to many minutes at a time.

Some Model I/III users will tolerate these delays, especially if they recognize what is happening. Others may be irritated by the delay, and it might be interesting for business users to study the cumulative delay and costs of such intermittent waiting.

If this string compression ("garbage collection," as it is aptly expressed), were speeded up, the delays could be negligible and tolerable. Indeed, string compression time can be improved considerably by using a program produced by PROSOFT, and called, appropriately, "TRASHMAN."

TRASHMAN is a machine language utility (hence the speed of operation), that can reduce the time for string compression by 95% or more. It needs a reserve of only 578 bytes of memory, plus just two bytes for each active string. It can be used with all of the major operating systems, and according to PROSOFT, most other machine language programs.

Obviously, it would be pointless to have TRASHMAN running with a program that had only a few strings. Although there would be a speed up of compression, the apparent speed difference would be unnoticeable. However, with several hundred strings active, including large string arrays, the speed up is not only observable, it is, well—amazing! I used it with a program that has nearly 500 strings involved. Without TRASHMAN, compression delay was clocked at 38 seconds. When I used TRASHMAN, the delay was approximately three seconds—I could not click the stopwatch off and on quickly enough to record it accurately. This is a speedup of about 93%—which is close to the tabulated claims for TRASHMAN. PROSOFT shows that 250 strings will compress in 0.7 seconds from a normal speed of 11.8 seconds (94% improvement); 1000 strings at 3.5 seconds compared to a normal compression time of 179.6 seconds (98%); and, 2000 strings (this is where the speedups are particularly noticeable), will compress in 7.8 seconds—compared to 713.2 seconds normally—a 98.9% improvement. A 98.9% improvement does not seem

to sound like much, when we so casually hear that something is "100% better than the sudsy cleaner," for example. Nevertheless, when you compare nearly 12 minutes (time for a leisurely coffee and doughnut break), and eight seconds—then you may be properly impressed.

INSTALLATION

Installing and using TRASHMAN is an easy task. It is distributed on a Model I single density disk in a format that can be read by any Model I or Model III, and any operating system. Two-drive users merely copy it from Drive 1 to the Drive 0 operating system. There are no restrictions to making a backup copy either. PROSOFT is equally solicitous of the one-drive user's needs—by placing the distribution disk in Drive 0 and pressing Reset, you are prompted through the process of getting TRASHMAN on to a disk of your own. (Not applicable to TRSDOS 2.7DD, but instructions for this are provided also.) Model III operators are provided with complete guidance for installing TRASHMAN and a special note for NEWDOS/80 users is provided.

OPERATING WITH TRASHMAN

Operating with TRASHMAN installed is not complicated, the fact that it is there is transparent to the user. Three steps are required in order to initiate its action. The first step briefly involves setting memory for TRASHMAN from DOS. Secondly, you go to BASIC from your DOS. And thirdly, you tell TRASHMAN how much space you wish to reserve for string compression. In most cases, this means that you only need to add one line to the beginning of your BASIC program, and you can leave this line in permanently, if you plan to be using TRASHMAN with the program at all times. In this event, TRASHMAN will always automatically request the space allocation. Otherwise, you may reserve space by keyboard entry each time, if you wish.

As was mentioned, TRASHMAN must know the amount of space as determined by the user. For example, two bytes are needed for each active string; hence, 200 strings will require 400 bytes. If there are doubts as to how much space to reserve, you can over-estimate, but if your estimate is too high, the line of code that you enter will catch it. You can then reduce your estimate and try again. The call to TRASHMAN is made via a USR call, included in the same initial line. This line, except for the variable factors, is included in the instructions.

That is the extent of installing and operating TRASHMAN. You just continue from this point by running your BASIC program, and, except for results, you are not aware of TRASHMAN. If there were previous delays for string compression while running the program, you should notice a significant change—indicating that TRASHMAN is busy doing "his" cleaning up in the background. The previous pauses for compression should be completely eliminated, or reduced to a few seconds. In the case of no apparent change being noted, then it is likely that insufficient string compression space was allocated. TRASHMAN then permits the computer to do normal compression.

CRITICAL FACTORS

This emphasis on string compres sion space points up a critical facto about TRASHMAN—to be mos efficient, just the right amount o space must be allocated. Allocate too much space and it may affec the BASIC program to the exten you may receive an "OUT OF STRING SPACE" or "OUT OF MEMORY" message. Or, if this does not occur, TRASHMAN will "clean up" too frequently—not a problem as severe, for the program will probably still be operating faster than without TRASHMAN, but not as fast as it could.

Another consideration, but not one that may be considered "critical," is that in allocating new space, TRASHMAN "Clears 50"—thus destroying all variables active at the time. Therefore, it is important to initiate TRASHMAN at the beginning of the program. If other BASIC programs are run, the space allocated by TRASHMAN is maintained. If this is not desired, then a reset or return to DOS must be made. It is

continued on page 17

SOFTWARE REVIEW

ACCEL3 BASIC COMPILER from SOUTHERN SOFTWARE

George Kwascha

Southern Software has finally improved their ACCEL compiler to the point where if you need an inexpensive BASIC compiler to speed up execution time and/or protect your source code, ACCEL3 is well worth the $99.95 price tag. If you are a previous purchaser of ACCEL2, like myself, you can send back your original tape or disk and get ACCEL3 for $52.00. Programs that previously could not be successfuly compiled by ACCEL2 can now be compiled by ACCEL3 with very little difficulty. ACCEL3 is available for tape, disk and stringy floppy systems.

ACCEL3 is a complete rewrite of all previous versions of the ACCEL compiler. It is much quicker in compiling code and creates a smaller subset than ACCEL2. Other improvements include the ability to compile: multi-dimensional arrays, variable dimensioned arrays and incomplete FOR-NEXT loops. ACCEL3 now supports the following commands: CSAVE, CLOAD, CLOAD?, SAVE, LOAD, and RUN. The IN and OUT functions are also optimized. ACCEL3 is a three pass compiler, where ACCEL2 made five passes.

What You Get

My copy of ACCEL3 came on a flippy diskette with the Model I version on one side and the Model III version on the reverse. The Model I version can be run on a Model III by using the CONVERT utility. The diskette is self-booting with an automatic loading routine to transfer a copy of ACCEL3 onto a system diskette. ACCEL3 is purported to run on any available operating system, since it is not system- or location-dependent. The loader allows you the option of locating ACCEL3 at any address of your choice or of letting the loader locate it at the highest available memory location. ACCEL2 users should note that the memory location is 32 bytes lower for ACCEL3. Locating it at previous locations will cause a re-boot during the third pass of compilation. I recommend that you let the loader locate ACCEL3 at the address of its choice, unless you are going to be using the compiler to compile programs for small size machines. Also included on the disk version are four other programs: EXEC, MUSIC, OVERRUN and MICE. EXEC is a useful utility that will allow you to execute a sequence of TRSDOS and/or BASIC commands. This can be extremely useful when you wish to insert your disk and watch it as it progresses through the loading routines of TRSDOS commands, setting file size and memory size and loading or running a BASIC program. The latter three programs are BASIC programs that are included on the disk to illustrate the marked improvement in execution speed that can be accomplished by compiling a BASIC program.

The Manual

The ACCEL3 manual is terrible! I do not enjoy making such a statement, since I love ACCEL3. If you need an excellent compiler, do not let my criticism of the manual keep you from purchasing this fine piece of software. The ACCEL3 manual is very much like the ACCEL2 manual. The content has been changed to comply with ACCEL3, a few pages have been added, but generally it is the same. There is not enough detail or explanation on the topics covered. The manual will make a statement, give you a short example with a few notes in parentheses, and proceed. A lot of what-ifs are left unanswered. Sentence wording and structure is awkward in places. It is difficult to follow the Model I and III procedures throughout the manual. Separate sections for Model I and III would be helpful. A table of contents would be useful. It is very difficult to find something in the manual that you remember reading. A few pages of documentation have been added to the manual on how to operate EXEC. I have not been very successful in understanding the information contained on these pages. I think that if we all complained together that maybe Southern Software would re-write the manual (at no additional cost?). Or, even better, maybe we could entice some enterprising free-lance writer to do it. (Hmm . . . not a bad idea.) First-time ACCEL3 users will have to wade through the documentation. Previous ACCEL2 users shouldn't have too much trouble.

Operation

Operation of ACCEL3 is similar to ACCEL2, and I don't see the need to cover material that has already been reviewed in past publications. The result of compilation is a program that has a mixture of BASIC statements and Z80 machine code instructions. Below is a listing of the functions and operations compiled by ACCEL3.

GOTO, GOSUB, RETURN
VARPTR, POINT, INP
ASC, CHR$, LEN
RIGHT$, LEFT$, MID$
CVI, MKI$, CVS, MKS$, CVD, MKD$
PRINT, OUT
ON...GOSUB, ON...GOTO
IF...THEN...ELSE
SET, RESET
PEEK, POKE
FOR...NEXT
AND, OR, LET

Addition, subtraction, multiplication, division, string concatenation, comparisons, one and two dimensional arrays.

I have also noted that some of my LPRINTs get compiled. The best feature by far of the ACCEL3 compiler is that it allows compilation of improper FOR...NEXT loops. "Sloppy" FOR...NEXT loops are created when you exit the loop defore completing it. For example:

```
10 FOR I = 1 TO 5
20 IF X = 8 THEN 50
30 NEXT I
50 END
```

The above example could not be compiled by ACCEL2 because of the statement in line 20. Line 20 will cause program execution to exit the loop before I reaches the value 5 if the value of X is equal to 8. This is a simple example of the problem, but I found that 95% of the time this problem existed when ACCEL2 would not successfully compile a program. ACCEL3 has eliminahed this problem altogether! Listing the compiled program shows that

ACCEL3 does not leave blank lines for REMARK statements or REM lines for compiled statements as ACCEL2 did. All but uncN0iled lines are removed.

Some of the limitations of ACCEL3:

1. ACCEL3 will not allow you to redefine variables within the program, i.e., you cannot use the CLEAR statement after any definition statements or declare a variable an integer and then change it to single precision.

2. TRON will not work properly since line numbers have been adulterated during compilation. BASIC diagnostic messages will also be in error when referring to specific line numbers.

3. Out of memory cannot be diagnosed at run time.

4. Any diagnostic information provided by ON ERROR may not be correct due to the manner in which the compiler handles code.

5. A single colon (:) on the first line is not allowed.

6. The following commands cannot be used on a compiled program: EDIT, DELETE, MERGE, NAME and AUTO.

7. Do not use GOSUBxxx, GOTOxxx or RUNxxx (where xxx is a line number) as a direct keyboard command, since the line may not exist.

8. ACCEL3 cannot handle some long strings. If a string error occurs during compilation, the string must be broken up.

9. SAVE and CSAVE cannot be used within a program. They are strictly keyboard commands.

10. ACCEL3 compiled code will poll the keyboard if a colon is present at the front of the line.

Conclusion

In conclusion, I must recommend the ACCEL3 BASIC Compiler. For compiling most BASIC programs it cannot be beat. It is quick and simple to use. Compiled programs that are sold do not require royalties. The manual can be improved upon, but does serve the purpose of teaching you to operate the software. ACCEL3 is a must for all ACCEL2 users!

ACCEL3 is distributed by Algorix, Box 11721, San Francisco, California 94101.

George Kwascha
2023 Milton Road
Durham, NC 27712 ■

CAN THE COMPUTER SAVE THE COUNTRY?

Mike Shadick

I guess the logical question would be: is America in NEED of saving? And, if our nation IS in some sort of jeopardy, from whence cometh the threat?

To say that America is headed for trouble in the twenty-first century is — according to some gloom-and-doomsayers — an unarguable fact of our nation's existence. Yet the source of the trouble is not so much outside of our national borders as it is from WITHIN. Problems which have seethed and boiled and eaten away at out national inards for decades are today making themselves painfully heard, seen, and felt, in virtually every aspect of our lives.

Not the least of the problems which we — both as individuals and as a national entity — face, is FRICTION, the sort of friction which can grind huge factories to a screeching halt, overnight. I am speaking, of course, of the traditional friction between labor and management. It appears, indeed, to be a distinctly AMERICAN tradition, does it not? Why, it could even be said that America achieved its zenith at the very time when management's formerly-omnipotent power was finally matched by, and eventually throttled by, organized labor. The unions' hard-fought and frequently-victorious battles for more of just about everything were once heralded as a prime factor in America's preeminent greatness and world leadership.

Today, however, all of the above is open to question! Being questioned most of all, perhaps is whether both labor and management (and, by implication, America itself) might not benefit more from a posture of mutually-beneficial cooperation, rather than from the workers' and the bosses' traditional positions: namely, at each other's throats.

Enter the computer.

Dare I imply that computery is a solution for labor/management friction? I do, indeed! For IT IS A FACT WHICH HAS ALREADY BEEN CONCLUSIVELY DEMONSTRATED.

Computers, you see, portend to be the great EQUALIZERS of society. A man or woman interacting with computery is neither white collar nor blue collar; neither executive nor underling; NEITHER MANAGEMENT NOR WORKER. In short, he or she is nothing more — and nothing LESS — than a fully individual and EQUAL human being, increasing his or her own proactivity through computer interaction.

Those of you who have ready access to your own computery — which includes the vast majority of this readership — have undoubtedly observed already the phenomenon of which I speak. Distinctions between people — especially the socially-created distinctions of rank and class and status — tend to dissolve and disappear, when one is interacting with a computer. Indeed, it can validly be stated that, for the duration of such interaction, distinctions between persons tend to disappear ALL TOGETHER.

This is, of course, exactly as it should be! For the computer itself, after all, knows no such distinctions. So, for its operators to RETAIN them would be highly counterproductive to their own computer interaction — to say the least!

Thus we thereby have an effective scenario for the resolution of labor/management disputes and differences. Not overnight, of course! Nor can I, at this point, furnish the programming specifics which would enable the scenario above to be implemented. Suffice it to say, however, with regard to labor/management relations, that if binding arbitration could be accomplished via computery, it would be infinitely more palatable for ALL parties concerned.

Am I some sort of microdreamer? Perhaps! Yet, the ironing out of differences between corporations and the labor unions representing their employees could, in fact, be accomplished in a much more expeditious fashion than is currently the case, if "the great equalizer" were employed (!) between them. Traditional foes would find new friends. Those who once were each other's adversaries would become advocates for one another's causes and interests. Most significantly, labor and management would begin to foster a new spirit of MUTUAL cooperation and support.

Alas, such a spirit has been woefully lacking in American industrial relations — among other places!

But yet, I'm the first to admit that my computerized scenario for labor/management harmony is futuristic — if not fantastic! Yet the fact remains that it COULD work. Moreover, it WILL work — if those traditional foes truly desire to eliminate friction, and to thereby get the wheels of American industry smoothly rolling once again. I'm fully convinced that the only way it will be done is — quite literally — BIT BY BIT. The "great equalizer" is today's (and tomorrow's) only visible vehicle for bringing labor and management close enough together to matter for America — if you catch my microdrift!

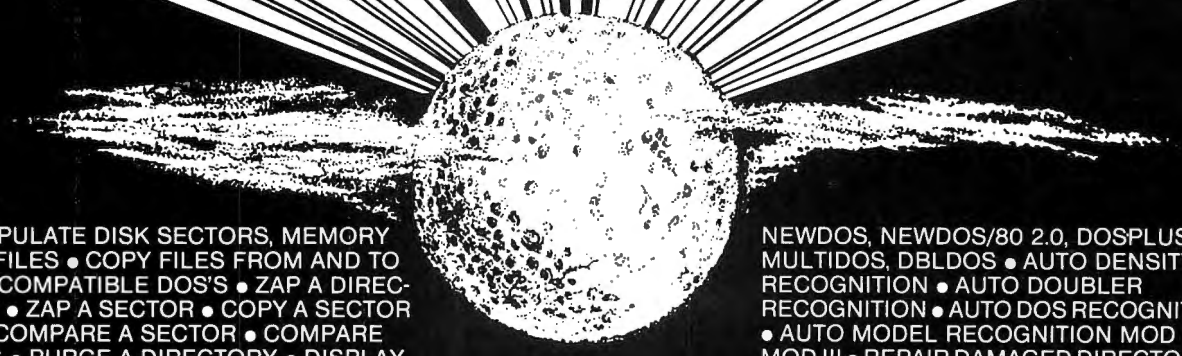
Mike Shadick
Cedar Square West, Apt. E-414
1515 South Fourth Street
Minneapolis, MN 55454 ■

EVERY FEW YEARS A UNIQUE PROGRAM ARRIVES

...SO PRACTICAL...SO USEFUL IT BECOMES HERALDED BY COMPUTER OWNERS INTERNATIONALLY AS A TRUE CLASSIC...AN INDUSTRY STANDARD...HIGHLY ACCLAIMED BY EVERY MAJOR TRS-80 PUBLICATION. YOU'VE SEEN IT WITH ONLY A HANDFUL OF TRS-80 PROGRAMS...NOW EXPERIENCE A LEGEND WITH THE NEW...

VERSION 3.0

SUPER UTILITY PLUS



MANIPULATE DISK SECTORS, MEMORY AND FILES • COPY FILES FROM AND TO NON-COMPATIBLE DOS'S • ZAP A DIRECTORY • ZAP A SECTOR • COPY A SECTOR AND COMPARE A SECTOR • COMPARE FILES • PURGE A DIRECTORY • DISPLAY DISK MAPS • EXAMINE FILE LOCATIONS • READ AN ALTER DATA ADDRESS MARKS • DOUBLE SIDED DRIVE SUPPORT LDOS, DOSPLUS, AND MULTIDOS • MORE • BOOTS ON MOD I OR MOD III 35, 40, OR 80 TRACKS • SINGLE OR DOUBLE DENSITY • TEST MEMORY • READS ALL CURRENT POPULAR DOS'S; TRDOS, LDOS,

80 Readers' Choice 1982 — **1st PLACE 80-MICRO 1982 UTILITY PROGRAM OF THE YEAR**

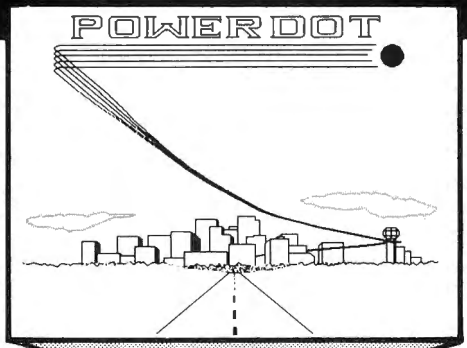
PROTECTED MEDIA
BACK UP COPY INCLUDED

NEWDOS, NEWDOS/80 2.0, DOSPLUS, MULTIDOS, DBLDOS • AUTO DENSITY RECOGNITION • AUTO DOUBLER RECOGNITION • AUTO DOS RECOGNITION • AUTO MODEL RECOGNITION MOD I OR MOD III • REPAIR DAMAGED DIRECTORIES • REPAIR BOOT SECTORS • REPAIR HIT & GAT TABLES • CREATE A SPECIAL OR CUSTOM FORMATTED DISK • REFORMAT WITHOUT ERASE • BUILT IN PRINTER ROUTINES • SOFTWARE BULK ERASE A DISKETTE • PATCHES TRSDOS AT OWNERS OPTION • EASILY CONFIGURED FOR ANY SYSTEM • MORE •

THE PROGRAM W/MANUAL 79.95 • THE TECH MANUAL 14.95 • THE BOOK INSIDE SU+ 19.95

POWERDOT

A GRAPHICS BREAKTHROUGH! Now the ULTIMATE in graphics design and editing. PowerDOT is the state-of-the-art in HI-RES Dot Mappers. Unmatched in quality and features. PowerDOT allows the user to create TRUE HI-RES printouts on the EPSON (Graftrax or Graftrax+ req.), or the C.Itoh PROWRITER from Leading Edge without ANY hardware modifications! The program has the ability to create drawings several times larger than the computer screen. You are only limited by disk storage. Your screen is merely a "picture window" to a much larger drawing area. You define the picture size. You select the print mode: hi-res, lo-res, or ultra-res. A very versatile utility and truly ahead of its time. Only **$49.95.**



POWERDRAW

A full screen graphics editor. PowerDRAW allows the user to draw directly to the screen utilizing cursor controlled movement. Graphics and text may be combined and saved directly to disk or tape, and recalled at a later time. The "SAVE" formats are: CONDENSED TOKENS, EDAS SOURCE, EDTASM SOURCE, BASIC DATA STATEMENTS, BASICS STRINGS, BASIC ARRAYS, and LOAD FILE FORMAT. The files may be run alone or merged into any type of program or application. Animation is also possible when screens are sequenced. Design custom forms, titles, pictures, and graphics for your applications. Received EXCELLENT reviews in INFOWORLD 6/82, 80-US 10/82, and 80-MICRO 10/82. Three printer drivers are included, and the program is JOYSTICK compatible. A GREAT value at only **$39.95.**

POWERMAIL

The most powerful, mailing system for the TRS80. Powermail is a highly sophisticated mass mailing system designed to run under all of the popular DOS's currently available for the Mod I or III. The program is written entirely in machine language for maximum operation speed, and occupies only 4K of the available RAM in your computer. There are no slow periods when Powermail is running. New features have been added to the program that others have always lacked. You now have the ability to keep track of mailings using the 24 flags that are incorporated into the Powermail program. The Powermail system will handle a file up to 8 megabytes, or 65,535 names, whichever is smaller. The program will run in as little as 32K and one disk drive, although 48K and 2 drives are desirable. The program will also sort the entire maximum file size and open up to 168 files simultaneously during the process. Only **$99.95**

PROGRAMS OF TOMORROW HERE TODAY

POWERDRIVER

New printer drivers for the new Radio Shack™ Superscripsit™ This new generation of custom printer drivers allows you to utilize all of SUPERSCRIPSIT'S features with your EPSON, PROWRITER and C.ITOH F-10 STARWRITER printers. You can now utilize all of the printers custom features such as compressed, expanded and proportional print, underlining, boldface, superscripting, and sub-scripting. All drivers can be called from within SUPERSCRIPSIT at "document.open time." Order by printer designation as follows: Only **$29.95** each

POWERDRIVER E	EPSON MX70/80/100
POWERDRIVER P	C.ITOH PROWRITER
POWERDRIVER F	C.ITOH F-10

POWERSOFT
Products from Breeze/QSD, Inc.

Available from Selected Dealers Everywhere

11500 Stemmons Fwy., Suite 125
Dallas, Texas 75229
To order call toll free 1-800-527-7432
For product information (214) 484-2976

LETTERS TO THE EDITOR

Questionable Business Practices

I am writing you in regard to an advertisement on page 45 of your November 1982 issue, by "Data Resources," at 304 Elati Street in Denver, Colorado, purporting to offer a microcomputer called the LNW80 and Verbatim diskettes.

I was solicited by a postcard from this organization in August 1982 and responded with a telephone order for 30 diskettes on August 23, 1982. A charge was immediately made against my Mastercard account for $474.85, but the diskettes were never shipped in spite of frequent phone calls. It was repeatedly claimed that the diskettes had not been available.

On October 15, I requested that my money be refunded. Shortly thereafter, Data Resources ceased answering their telephones, and on about November 11, the lines were disconnected.

My experience again illustrates the extreme danger of responding to advertisements for mail order merchandise from little-known firms.

I would further suggest a charge made to Mastercard before the merchandise is shipped should immediately alert one to the possibility of fraud.

I have reported this to the Denver Better Business Bureau and to the Postal Inspector.

Stanley M. White, M.D.
49 Alpine Way
Asheville, NC 28805

Advice for Computer Widows

Just in the past month since Michael Herbert Shadick got his new computer and has taken in piecemeal typography jobs, I've learned how forelorn "computer widows" feel when left to their own demises, while their husbands "fool around" with their perky computers.

You see, Mike used to work for a service that does typesetting and typography near the apartment complex where we live. Mike would update their labels for mailing or keyboard some manuscripts, and then he would come home as soon as the project was completed. He always disliked the distraction, though, of leaving his writing projects—because he spends hours pouring over articles he writes on computers, pieces for children's publications, and bits and pieces for religious bulletins. But once he'd finished the project that had brought him into work that day, he'd come whistling cheerfully, saying the program he'd run had gone "extra well." Then, after dinner, he'd put his writing projects aside, so we could have a rousing game of scrabble, which we play with great vengenace, or we'd just sit and "rap."

All this until ... about a month ago. Michael came home quite agitated, his eyes fairly sparkling and jubilantly exclaimed, "I've just bought a computer. I don't have to go into work any more." A criticism of his "break habits"—having too many cigarettes on company work time, he said, had prompted him to return his employers' guff by offering to buy their computer and subsequently take the work home.

That's how he came to join the ranks of the stay-at-home computer buffs. "There's a lot less hassle," he maintains, although he has to keyboard more pages of material, since along with his new self-instigated freedom has come piecemeal wages. My man has become an inveritable monster until all his work has been keyboarded—and as of late the projects seem to be a lot more lengthy. I see more of Mike around the house, but he's wrapped up most of the time in his "affair" with the new computer.

What computer skeptics have always maintained, that the computer will change us—internally as well as externally—seems to be ringing true. To quote some self-styled skeptic, "Computers are beginning to effect every facet of our lives ... just like fire!"

What's the best way for a "computer widow" to fight it? Well, I'm a pretty good typist, or should I say keyboarder, and I got Mike to show me how to give the commands to his computer. Don't think I'm going to let that computer push me around. I think that what we need is another computer in the family—and a printer we can both use. Then we can *both* get down to business.

Sallie Stephenson
Cedar Square West, Apt. E-414
1515 South Fourth Street
Minneapolis, MN 55454

Updates to DATA-WRITER

Computronics readers may be interested in the further enhancement of DATA-WRITER ("Program Previews," by A.A. Wicks, *Computronics*, January 1983) by the addition of ACCESS, a program module that permits access of any record (up to 10,000) in just one second. DATA-WRITER 2.0, including ACCESS, is available at a price of $145. Additionally, I would like to note that our toll-free order line was listed incorrectly in your Advertising Directory. The correct phone number is 800-221-1624.

Jerry White, President
Software Options, Inc.
19 Rector Street
New York, NY 10006

Changes to Kaleidoscope

I've recently acquired the Oct.'82 edition of your magazine, which I'm still enjoying. I'm a computer hobbyist, and own a compatible Model I Level II computer.

So, fumbling through some of the programs, I've come across "Kaleidoscope." It's simple, neat and very nicely done. The Zeliff's are to be congratulated.

Nevertheless, I've tried some modifications that I think improves it a little bit. The first ones are in lines 350 and 360. Changing them to:

350 IF N=2 THEN S=1
360 IF N=4 THEN N=0:S=0: GOTO120

makes the program run, in a way, faster and without interruption. Then, I've changed lines 160 and 250 to read:

160 FOR Y=INT(B/2)+B+W TO B STEP-1

250 FOR Y=B TO INT(B/2)+B+W

which creates an expanded version of the original kaleidoscope. It'll take just a little bit longer for it to fill up the screen, but I thought it well worthwhile.

I hope it will give you and your readers as many good moments as it's giving me. And I would like to thank Steve and Sandy for the magnificent idea, and beg their pardon for these uncalled-for changes.

My very best regards to all.

Eduardo Hauff
CP. 3699
01000 Sao Paulo, Brazil

PROGRAM PREVIEWS
continued from page 11

possible to deactivate TRASHMAN, however, by requesting zero space in a "Define USR" statement. TRASHMAN also may be removed from memory, by the same method.

INFORMATION SHEET

The extent of the program does not warrant a manual, as such, but a four-page 8 1/2 by 11 inch pamphlet is adequate, well-written, and provides all the information anyone will require.

BACKGROUND

TRASHMAN was written by Glenn Tesler of PROSOFT, son of the author of NEWSCRIPT, Chuck Tesler, and now listed as co-author of the latter program. This astute teen-ager, who was 14 years old when he wrote TRASHMAN, has produced a very useful and technically complex machine language utility program. The functions of TRASHMAN can be seen operating at its ultimate capability during the operation of NEWSCRIPT. The original NEWSCRIPT (by another name), required a boring and frustrating amount of time in performing space compression. The latest NEWSCRIPT does this so rapidly that the time might not be noticeable if it were not for a brief flash of an enlarged letter "C" in the upper right corner of the screen. The "C" indicates "Compression Occurring," but, as mentioned, it could be eliminated and the delay would hardly be noticed. In this connection, when using NEWSCRIPT recently, I returned to DOS and then to BASIC to work on another program (which uses a large number of strings). I was surprised to note that the "C" was flashing from time to time on the screen as I operated the program, just as it did with NEWSCRIPT—indicating that the NEWSCRIPT version of TRASHMAN was still in memory, working away at compression.

SUMMARY

TRASHMAN is one of those programs that could be classified as"desirable but not essential." It performs correctly and satisfactorily, but unless compression delays are bothering a program user, it probably would not be classified as an urgent necessity. However, because of its simplicity, and especially if delays are annoyingly frequent, then TRASHMAN is the answer, and at the price, it does not take many operating delays to justify its purchase.

TRASHMAN. For TRS-80 Model I/III, Disk. PROSOFT, Box 560, North Hollywood, CA 91603, $39.95.

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

BITS AND PIECES
continued from page 9

printer among several terminals or micros. Little-used redundant printers are eliminated, resulting in substantial savings.

SMARTBUFFER allows software and hardware selection of outputs, can perform code conversions, and are user programmable. SMARTBUFFER is available in parallel, serial, or with both in up to 8 channels.

SMARTBUFFER sells for $335 and up and is available from DATAMATCH CORP., 3810 Oakcliff Ind. Ct., Atlanta GA 30340 (404) 441-0408. ■

ARRAY OF HOPE FOR BASIC PROGRAMMERS (PART 7)

Arne Rohde

6.2.5 Quicksort

There are a number of sorting algorithms which are known by the name Quicksort, but they are all variations on a sorting algorithm first described by C. A. R. Hoare in 1962. The method has also been called a partition exchange sort, since the array to be sorted is split into partitions which are then ordered in a particular sequence. Unlike the Shell sort, quicksort does not compare fixed items in the array, but is dependent on the key values being compared. As the name also implies, quicksort in its various guises is one of the fastest sorting algorithms known for sorting large arrays of random data. It does not seem to be very widely used in the microcomputer industry, although the MMSForth package includes a very impressive demonstration of the potential sorting speed of quicksort compared to Shellsort, an insertion sort, and a selection sort.

The quicksort, like the Shellsort, does not leave duplicate key values in their original sequence, but unlike Shellsort it requires a varying amount of extra memory to store array pointers. For this reason the storage of these pointers is often done on a stack in stack oriented languages such as Forth, or in assembly language. However, since there is an upper limit on the number of items which may be stored, we can dimension an array in Basic to serve the same purpose.

In quicksort it is necessary to keep track of the unordered portion of the array which remains. Initially, of course, this is the complete array. One of the remaining portions is then ordered, perhaps creating two smaller unordered portions in the process. The array limits of one of these portions is then stored, and the other portion is then ordered, and so on until each portion of the array is in sequence. If the portion to be ordered next is chosen as the smaller of the two portions just created then we will have the smallest possible upper limit on the number of pointers to be stored.

The sort is not as simple to illustrate as the previous sort methods, and for small amounts of data it can seem confusing. The first phase consists of choosing a particular key value from the array, and storing this temporarily. The array is then searched from the beginning until an element with a higher key value is found. It is then searched from the other end for a lower key value, and these two values are then exchanged, unless the lower key value has a lower position in the array. If this is the case, then the temporary key is exchanged with the last item with a lower key value. All elements in the lower part of the array will then have a lower key value, and all elements in the higher part of the array will have a higher key value. The chosen key will be in its correct final position, and can be ignored for the remainder of the sort. The other two portions of the array are then treated in a similar manner.

The crucial part of the sort consists in choosing a sensible key value as the so-called pivot item. The best possible value results in two portions of equal size, the worst possible gives one portion with only a single element. Several suggestions have been made for choosing the best value. The simplest method is to choose the first item in the portion to be ordered, but this is a very bad choice if the array is almost in sequence. Another suggestion is to choose the central item of the portion, and another takes the median value of the first, middle and last items. The last suggestion is probably one of the best, even if it requires extra calculations.

The first phase of a quicksort using our array, and choosing the first item as pivot, can be illustrated as follows:

Initial array	3	5	1	6	2	7	4
Choose pivot = 3	3						
Find first with higher key		*					
Find last with lower key							*
Try next						*	
Lower value found					*		
Exchange lower and higher		2			5		
Find next with higher			*				
This one is higher				*			
Find next with lower				*			
We have met in the middle							
Exchange pivot with prev.	1		3				
Sequence after phase 1	1	2	3	6	5	7	4

Key value 3 is now in its correct position, but now there are two potentially unordered portions of the array, namely item 1 and 2, and items 4, 5, 6, and 7. Since 1 and 2 is the smallest portion, we store the limits 4 and 7 (array pointers, not key values) and continue with items 1 and 2. Since this portion only contains two elements, it can be ordered with a compare and exchange if necessary. No exchange is required in this case, and no new portions are created. We can now look at the stored limits, and start ordering the portion from 4 to 7. The sequence is the same as for the first phase.

Initial sequence	6	5	7	4
Choose pivot = 6	6			
Find first higher key		*		
Found here - value 7			*	
Now find lower key				*
Found, exchange them			4	7
Find next higher				*
The end, exchange pivot	4		6	
Final sequence	4	5	6	7

The smallest remaining portion, item 7, is only a single item, so it must be in sequence. We do not know if the other portion, items 4 and 5, is in sequence, so they are compared and exchanged if necessary. Since there are now no portions of the array stored for later processing the sort is completed, and the elements are ordered in sequence.

An array of this size should not be sorted with quicksort, and in practice when portions of this size are created during a sort they will usually be sorted by some simpler method, such as a bubble sort or sifting sort. However, to simplify the code in our example, we shall use the quicksort method for all portions with 3 or more items. For timing purposes several methods of choosing the pivot will be tried, namely the first in the portion, the

continued on page 20

continued from page 18

middle item in the portion, the last item in the portion, and the median of the first, middle and last item. For random arrays the method should not be of much significance, but for arrays almost in sequence it can make a noticeable difference. Irrespective of the pivot chosen, it will initially be moved to a temporary data item to reduce the amount of subscripting for comparisons, and the first item in the portion will be left empty during the phase.

For this method we shall define an extra two dimensional array to store the lower and upper limits of the remaining portions to be ordered. The largest number of items in this array should be Log2 of the number of items to be sorted, or 7 for arrays up to 127 elements, 8 for up to 255, etc. The logarithm can be found in BASIC, although we only have natural logarithms available. The formula for changing base from e (or natural logarithm) to base 2 is

```
LOG2(n) = LOGe(n) / LOGe(2)
```

The right hand side of this formula is easily calculated in BASIC, and hence we have the desired value for dimensioning the array. This array will be defined as TLIM(x,1), where x is the above value. The pointer to the last used position in this array will be TP. The other definitions will be as for the other sort algorithms we have examined, but more indexes will be required than previously.

```
10 TM = LOG(AM+1)/LOG(2)+1: DIM TLIM(TM,1)
...
100 TLIM(0,0) = 0: TLIM(0,1) = AE - 1: REM set array limit
110 TP = 1: REM One portion remaining
120 IF TP < 1 THEN 310: REM No more portions - end
130 TP = TP - 1: IL = TLIM(TP,0): IH = TLIM(TP,1)
140 IF IH - IL < 1 THEN 120: REM 0 or 1 element portion
150 IF IH - IL = 1:
        IF A(IH) < A(IL) THEN AW = A(IH): A(IH) = A(IL):
            A(IL) = AW: GOTO 120
        ELSE 120: REM 2 elements in portion, sort them
160 I = INT((IH + IL) / 2): REM Middle item index
170 IF A(I) < A(IL) THEN AW = A(I): A(I) = A(IL):
        A(IL) = AW: REM Sort first, middle, last
180 IF A(I) > A(IH) THEN AW = A(I): A(I) = A(IH):
        A(IH) = AW:
        IF A(I) < A(IL) THEN AW = A(I): A(I) = A(IL):
            A(IL) = AW: REM these 3 now in sequence
190 IF IH - IL = 2 THEN 120: REM Only 3 in portion
200 AP = A(I): A(I) = A(IL): REM Make pivot = A(middle)
210 JL = IL + 1: JH = IH: REM indexes for search
220 IF JL > JH THEN 280: REM Portion finished
230 IF A(JL) <= AP THEN JL = JL + 1: GOTO 220
240 IF JH <= JL THEN 280: REM portion finished
250 IF A(JH) >= AP THEN JH = JH - 1: GOTO 240
260 AW = A(JL): A(JL) = A(JH): A(JH) = AW
270 JL = JL + 1: JH = JH - 1: GOTO 220
280 A(IL) = A(JL - 1): A(JL - 1) = AP: REM Pivot in place
290 IF (JL - 2 - IL) < (IH - JL) THEN
        TLIM(TP, 0) = JL: TLIM(TP, 1) = IH: IH = JL-2
    ELSE TLIM(TP, 0) = IL: TLIM(TP, 1) = JL - 2: IL = JL
300 TP = TP + 1: GOTO 140: REM Use smallest portion
310 : REM Sort completed
```

The outer loop starts in line 120, and the routine returns to this point whenever a new set of limits is to be removed from the temporary array TLIM. If there are no stored limits, the sort is completed; otherwise a check is made for the size of the portion to be ordered. If it is less than two elements, it is already in sequence. If it consists of two elements, these are compared and exchanged if necessary, and a new set of limits found. Line 160 to 190 are used to find the item to be used as the pivot. In this example it is done by placing the first, middle and last items in sequence. This has the added advantage that if the portion only consists of three elements, they will be in sequence after this code, and this is checked for in line 190.

If there are more than three items, the middle item is moved to the pivot AP, and the first item moved to the middle. Because the last item in the portion is known in this case to be larger than the pivot, line 210 could be changed to set JH to IH - 1 instead of IH. This change cannot be made if another method is used to choose the pivot item. Lines 220 and 230 find an element which is larger than the pivot, starting at the second item of the portion being ordered. If index JH is bypassed, control passes to line 280; otherwise lines 240 and 250 are used to find an element with a lower key value than the pivot, starting at the end of the portion. If an item is found before the index JL is bypassed, the two elements are exchanged and the search continues at line 220.

When control passes to line 280, the portion has been ordered, and the pivot is moved to the correct position in the array, after moving the element at that point to the beginning of the portion. Lines 290 and 300 then find the smallest of the two portions just created, and stores the limits of the largest portion in the temporary array. The sort process then continues with the new limits at line 140.

If another method of finding the pivot item is desired, the code can be simplified, since the lines used to find this can be removed. Lines 160 to 200 should be replaced by the new code which places the pivot item in AP and leaves the first position in the portion free.

Another possible change which can be used to speed up the sort would be to check for the size of the portion to be ordered. If it is lower than a certain value, then another sort method, such as the sifting sort or the Shellsort, could be used to arrange the items in that portion in sequence. In the timing examples, this technique will be used to test the possible savings. This is done by replacing line 150 with the following line:

```
150 IF IH - IL < x GOSUB 1000: GOTO 120
```

where x is the number of items used as limit, and line 1000 and following contains a routine to sort the items from index IL to IH into sequence. If the first, middle and last items were still ordered to find the median, then it would be logical in this case to order them in the sequence middle, first, last to simplify the movement of the pivot item. Line 190 should then also be removed. In the timing examples, the sifting sort was used in some cases if the number of items in a portion was 10 or less.

The times for various array sizes, using the median value of the first, middle and last items were as follows:

(see top of next column)

Significant savings are apparently possible by using

continued on page 22

BeaLin Corp. *OFFERS YOU...*

HARDWARE & SOFTWARE

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 ..... ($69.95)
TRAX-SW WITH GRAFTRAX-PLUS ..... ($129.95)
GRAFTRAX-PLUS ..... ($69.95)
GRAFTRAX 80 ..... ($65.00)

(TRAX-SW and TRAX-SW with GRAFTRAX-PLUS available until Feb. 1, 1983 at the introductory prices of $49.95 and $109.95 respectively.)

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

TEAC DISK DRIVES 40 TK/SS/DD .. ($260.00)
(one year warranty/price includes case and power supply)

HAYES SMARTMODEM $229.00
1200 BAUD VERSION $599.00

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.

(MODEL II & XVI requires cable modification)

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.)

LNW COMPUTERS

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

LNW MODEL I $1495.00
48K/4MHZ OPERATION 5-8 Double density controller supports Hi-Res B&W and Color TRS-80 Model I compatible; supplied with Dos + 3.4 and LNW Basic.

LNW MODEL II $1995.00
125K & all Model I features, supports Model I systems and CPM. Supplied with CPM.

(Special prices for package purchases - call for quotation)

LNW EXPANSION SYSTEM..... $369.95
32 K memory with RS232 for use with *TRS80 MODEL I

VERBATIM DATA LIFE DISKETTS (Box of 10) $29.95

Epson Dust Covers for MX80, MX80FT and MX100. ***(call for prices)***

INVOICE-PLUS $69.95
Compatible with most operating systems, runs on MODEL I or III, store all transactions on disk, print on entry or spool for printing later.

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) 490-2744 MODEM (301) 730-2229

**Trademark of Tandy Corp.*
***Trademark of Epson*

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

VISA OR MASTERCARD ACCEPTED — DEALERS INQUIRIES WELCOME

continued from page 20

Array size	Quicksort only		Quicksort + sifting	
	Random	Almost seq	Random	Almost seq
2Ø	4.3	3.8	3.4	2.9
5Ø	11.1	9.4	9.8	7.5
1ØØ	25.9	21.8	23.7	15.5
25Ø	73.Ø	61.5	64.8	48.Ø
5ØØ	169.2	162.4	152.5	1Ø6.8
1ØØØ	356.7	299.9	336.6	245.9

another sort method for small portions of the array, especially if there is some sequence in the elements previously. It also seems that, for arrays in random sequence, quicksort is faster than Shellsort for arrays with more than about 50 to 100 elements. However, for keys almost in sequence, Shellsort is considerably faster for all sizes. Both of them, though, are beaten by the sifting sort for arrays almost in sequence. If large random arrays are sorted, quicksort can be much faster than any of the other sorting methods we have examined, although at the expense of slightly more complicated code.

Times were also measured for other methods of choosing the pivot item. In this case the times for each array size varied more than for the method used above. In almost all cases, averages of several runs are used to generate the times given. The times measured were as follows:

Array size	Last item Alm seq.	Middle item Almost seq.	First item Quicksort Almost seq	First item Quicksort Random	First item Quick+sift Random
2Ø	8.9	4.1	7.3	4.1	3.5
5Ø	37.8	11.Ø	26.8	12.7	9.6
1ØØ	112.8	22.8	1Ø2.Ø	28.2	22.5
25Ø		65.Ø		82.9	78.8
5ØØ		146.2		193.3	162.4
1ØØØ					366.7

Here the significance of choosing a good value for the pivot can be seen clearly. A bad choice can give significantly longer sort times, especially if there is some sort of order in the key values beforehand. The last column should be compared with the times in the previous table. It shows that even for random keys it is better to spend some processing time to find a suitable pivot value, especially for large arrays.

The quicksort and sifting sort combined was also used to test two different methods of sorting a string array. If sufficient memory is available for pointers to the string array, it can be significantly faster to sort the pointer array rather than the string array. This is especially true when the string area only has limited free space, thus requiring a number of string reorganizations. The sort was run with differing string lengths, but with the same amount of string space reserved. For smaller arrays, no reorganization was needed, but for larger arrays, more time was spent on reorganizing than on actually sorting.

In all cases, 20000 bytes of storage were reserved for strings, and for the pointer array two bytes were required for each string item. The times were measured for strings with an average length of 8.5 bytes, and for strings with an average length of 70 bytes. With this length only the smaller arrays could be sorted with the available string space. In all cases the string area was reorganized before the sorting commenced. The measured times were as follows:

Array size	String sort 8.5 byte	String sort 7Ø byte	Pointer sort 8.5 byte	Pointer sort 7Ø byte
2Ø	3.5		3.7	
5Ø	9.6		9.9	
1ØØ	24.6	3Ø.2	26.3	25.2
25Ø	72.2	875.8	74.7	78.5
5ØØ	2Ø5.7		168.2	
1ØØØ	1264.2		378.3	

It can be seen that, for 8.5 byte strings, reorganization was required for the 500 and 1000 element arrays; and for 70 byte strings, most of the time for the 250 element sort was spent on reorganizing strings. For this case the pointer sort is more than 10 times faster than the string sort, and would still have been faster if the memory required by the pointer array had been reserved for strings instead.

If the pointer array is dimensioned as P(AM), then it should be initialised so that the value of P(I) is set to I. Instead of referring to A(I) for the compares, reference should be made to A(P(I)) which is the string pointed to by pointer item P(I). When an exchange occurs, the exchange should be between P(I) and P(J) and not A(P(I)) and A(P(J)), since only the pointers and not the strings themselves are exchanged.

In summary, quicksort can be recommended for sorting large arrays in memory, especially if it is combined with the sifting sort or Shellsort for sorting small portions of the array. However if it is known that the array is almost in sequence beforehand, the sifting sort on its own can still be recommended. Perhaps the first phase of the quicksort, which examines every element, could be used to count how many elements were out of sequence, and use the sifting sort for the complete array if only a few were found to be unsequenced. The number of exchanges made on this pass is a partial measure, but probably not sufficiently accurate.

This brings us to the end of sorting methods for the time being. There are many other methods available, and also a large area we have not covered, namely the sorting of data resident on external storage media. The techniques available for this type of sorting are also many and varied, and for anyone interested, a good coverage is given in some of the books mentioned in the literature list.

6.3 Sorting Literature

The books mentioned in the section on searching also contain algorithms for sorting arrays. Besides these, there are a large number of books and articles covering the topic sorting.

Sorting and Sort Systems by Harold Lorin, published by Addison Wesley as part of the Systems Programming Series, has one of the best coverages of sorting methods, both for internal and external sorting. The algorithms are coded in PL/I, but one of the appendices reproduces the algorithms from the journal *Communications of the ACM*,

continued on page 31

ONLY $69.95

The Magnificent WORD PROCESSING SYSTEM

For the TRS-80 Model I and III

- Written in fast Z80 machine language.
- Single key control of all editing functions for ease of use.
- Dynamic display of word count, line count, and free memory count.
- Superscripts, subscripts, underlined, bolded, expanded and condensed type styles - combine and intermix within a line.
- Automatically justifies and word-wraps on the screen as you type.
- Search, Replace, and Global Search and Replace.
- Odd and even page user-definable headers, footers, and page number lines.
- User-definable linespacing, sheet size, top, left, and bottom margins.
- Move blocks of text from disk, to disk, and within the text.
- Examine disk directory on any disk and kill files while editing.
- Full screen editing of EDTASM and BASIC text files.
- Automatic renumbering for EDTASM and BASIC files.
- Print contents of screen function.
- Print previewing formats text, inserts headers, numbers pages, etc. on the screen without printing it on paper.
- Page by page pausing capability for sheet fed printers.
- Supports both parallel and serial printers.
- Printer control code access.
- Supports proportional space justifying on Centronics 737, 739, Line Printer IV, Daisy Wheel II, Diablo, Qume, Starwriter, Spinwriter, C. Itoh Prowriter, NEC PC-8023A-C, TEC 8500R.
- Also supports special functions of Microline, MX-80, MX-100, and Graftrax Plus.
- Any character or symbol your printer can print is accessable with the Special Character feature.
- Works with NEWDOS, NEWDOS80, TRSDOS, LDOS, and DOSPLUS - Single or Double Density.

GUARANTEE

Many word processing systems claim theirs are the best, but few would dare to guarantee them. Not us! We are confident that **ZORLOF** is the most useful word processing system on the market for under $200.00. If you don't agree, return it within 30 days for a full refund.

CALL (305) 259-9397

Add $2.00 shipping & handling. Florida residents add 5% sales tax. Checks require 3 weeks to clear banks.

ANITEK

MasterCard VISA

ANITEK SOFTWARE PRODUCTS □ P.O. BOX 1136 □ MELBOURNE, FL. 32935 □ (305)259-9397

SOFTWARE REVIEW

REMASSEM-1

REMsoft, Inc.

Frank D. Gunseor

I've been programming in BASIC now for five years and I have become familiar with more dialects of this high level language than I care to think of. Assembly language programming holds out the promise of speed and increased flexibility and is necessary for many real-time control situations, thus making assembly language programming a goal many people would like to achieve.

If most of you are like I am you've purchased every book published on assembly language programming, and yet have not been able to sit down and start writing an assembly language program of your own.

To a large extent one must think differently to begin programing in assembly language, and REMASSEM-1 might just be the product you're looking for to help you achieve this elusive goal.

REMASSEM-1 offers an introduction to TRS-80 assembly language programming for the beginner using the CRT as both a blackboard and a filmstrip projector. While probably not utilizing the computer system to it's fullest, it offers a novel way to teach yourself assembly language programming in a painless and even enjoyable way. The tape-based system comes with eight tapes for three different uses. Five tapes are audio, and contain ten 40 minute lectures on assembly language programming by Joe Willis.

One side of the remaining three tapes is a system tape called BLKBRD, which is used to load the remaining five sides of the display tapes which provide the video display for the ten lessons.

Also included in the package is the third printing of the first edition of *TRS-80 Assembly Language Programming* by William Barden, a reprint from the Zilog *Z80 CPU Technical Manual,* ordered by operation group rather than alphabetically, and finally a small manual in a three-ring binder which contains flowcharts and listings of the two assembly language programs which are the main subjects of the courses' latter lessons.

The manual additionally presents a study plan, including recommended reading from the included book, geared to each lesson, and pre-addressed forms for asking REMsoft programmers questions about the course.

The lessons are well thought out and the display tapes' 32-character per line format makes for easy, strainless reading in the majority of the lessons. Mr. Willis makes excellent use of a flashing display to emphasize important points when necessary, but don't overdo it. The student is audibly prompted to press the down arrow to move to the next display.

The lessons are loaded in pairs in both the tape and disk versions, and you can step through each lesson for review or use the <M> Menu option to review the subject matter in each lesson.

Mr. Willis's easy-going manner and humor keeps what may at times become rather dry and seemingly repetitious material on a light plane, making learning seem almost painless.

The following is a short summary of the lessons. Lessons 1 and 2 cover binary, octal and hexadecimal numbering systems. Binary addition, two's complement, arithmetic flags, Boolean operators, Z80 registers and assembly language program format, i.e., EDTASM or T-BUG.

Lessons 3 and 4 include 8- and 16-bit load instructions and arithmetic.

In lessons 5 and 6, jumps, calls, returns, rotate and shift, bit manipulation, input, output, exchange, block transfer and search, pseudo ops and equates are examined. The first of two assembly language programs is introduced —QDMTST (Quick and Dirty Memory Test) — which writes zeros to each memory location and reads them back again.

Lessons 7 and 8 finish up QDMTST and introduce KBDTVMOD, a routine which modifies keyboard and CRT routines to allow selection of sending a character to a printer as well as the CRT.

Lessons 9 and 10 present a video modification routine, printer routines, and a message output routine. In addition, software development principles are discussed, including the importance of defining the problem and outlining the solution before coding begins. ALL programmers should post these principles near their desks.

Almost all of the programs loaded perfectly the first time on either my CTR-41 or CTR-80. Those that didn't needed a certain amount of volume adjustment, which didn't seem to be too critical. Even so, it would be nice to load all six sides of the system tapes without fiddling with the volume setting. Ssome of the audio lecture tapes had minor but annoying background noise, which should have been eliminated.

All in all, I was very satisfied with the course. It demonstrates what can be achieved in the way of educational uses for the microcomputer and what is a unique presentation even though it did not even begin to make use of the full power of the micro in this setting. Some may even say that it's an expensive method of presenting a film strip. I say it's an excellent start.

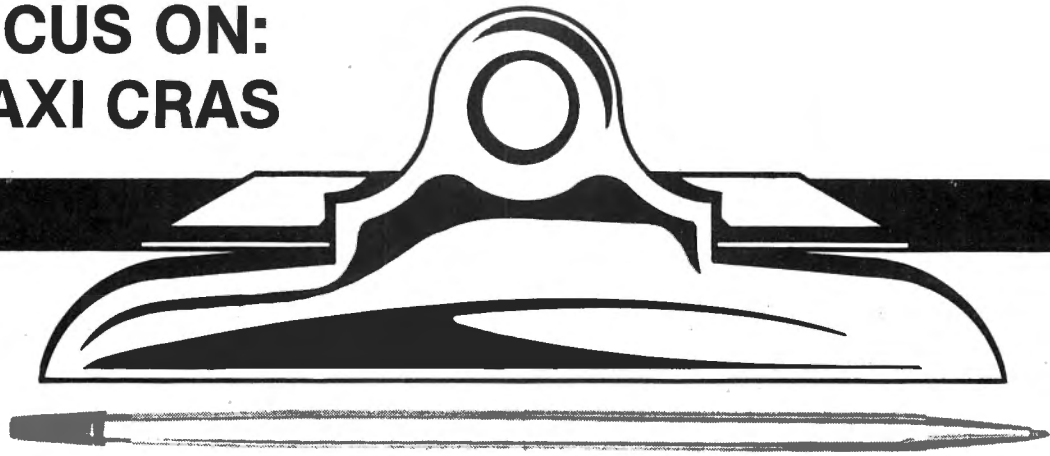
I hope Mr. Willis is contemplating a series including an intermediate and advanced course which could include computer testing and maybe some interactive programming examples.

I can strongly recommend this course to the beginning assembly language programmer who has little or no knowledge of what assembly language programming is all about.

Joseph E. Willis, REMASSEM-1. REMsoft Inc., Euclid, Ohio 44119. $74.95 (tape), $79.95 (disk). Available for the MOD I & III through H & E Computronics toll free: 800-431-2818 and other dealers.

Frank D. Gunseor
312 Fifth Avenue
Lewiston, ID 83501 ■

FOCUS ON: MAXI CRAS



What is Maxi CRAS?

Maxi CRAS (for Check Register Accounting System) takes the work out of printing checks, balancing your account, and reconciling it with bank statements. And that's only the beginning!

Who can use it?

Maxi CRAS is powerful enough for small businesses, and easy enough to handle for family or personal use. Best of all, it won't make you change the way you do business. Write checks by hand and enter them into the computer later, or let Maxi CRAS do all the work, and print checks automatically.

But what makes it different from all the rest?

We're glad you asked! Disk-based Maxi CRAS handles a virtually unlimited number of checks and deposits each month. And to keep track of all those financial transactions, Maxi CRAS supports up to 223 separate income and expense accounts, and your transactions can be assigned to one or even all 223 accounts.

Why is this so important? For some transactions it's not. Suppose you write a check for $250.00 to Jolly John's Jalopy Jumpers for repairs to your automobile. You would simply assign the check to the AUTO REPAIRS account. But suppose you had Jolly John fill the tank, and you picked up a gallon of milk while you were there. With Maxi CRAS you can simply add the $50.00 worth of gas and the $2.00 worth of milk to the $250.00 you owed him for the new windshield wiper, and write a check for $302.00. Many systems would force you to assign this check to a single account. But Maxi CRAS lets you charge $250.00 to AUTO REPAIRS, $50.00 to your GAS & OIL account, and $2.00 to your FOOD account. Six months, or even six years from now, you'll still know exactly what that $302.00 went for. That's power!

I'm interested. Tell me more!

Organizing the data is just half the picture. Maxi CRAS provides six essential reports and statements in a fraction of the time required to do them by hand. At tax time, you'll have all the data you need at your fingertips. No more sifting through shoeboxes full of cancelled checks!

THE BUSINESS DIVISION
A Subsidiary of Scott Adams, Inc

MAXI CRAS
By Dale Kubler
For your TRS-80 Model I or III with 48K, two disk drives, and an 80-column printer.
$99.95
Soon to be released for IBM/PC
THE BUSINESS DIVISION
BOX 3435 • LONGWOOD, FLORIDA 32750
(305) 862-6917

For detailed forecasting and budgeting, Maxi CRAS now interfaces EASILY with VISICALC(TM), the electronic spreadsheet program sold by Radio Shack. A Maxi CRAS exclusive!

The Last Check Register Accounting System You'll Ever Need!

Strong statement? Check out these features!

- Write checks by hand, or print automatically on single or continuous form checks (NEBS 9020).
- Data Entry routine second to none — saves time AND eliminates errors. You don't need to be a computer expert to use Maxi CRAS.
- The best selection of printed reports available — complete check register, income and expense subtotals, bank statement reconciliation, list of check register notes, and an account distribution statement. Compare!
- For even more detailed analysis, Maxi CRAS data is readable by VISICALAC(TM).
- Checkbook balance is constantly updated and instantly accessible. No more embarrassing overdrafts!
- A notes option can be used to flag tax-deductible transactions.
- Handles up to 223 income AND expense accounts. Many other systems only allow ONE income account.
- Assign transactions to a single account, or distribute over multiple accounts. Assign specific amounts, or pro-rate by fraction or percentage. Cash transactions can be recorded in any account without affecting check-book balance.
- Check Address data base stores up to 40 addresses and automatically prints them on your checks.
- Fast and easy bank statement reconciliation. Compare!
- Maxi CRAS is supplied with a complete User's Manual, sample printouts, and TDOS, a special version of the DOSPLUS operating system.

WE INVITE YOU TO COMPARE.

Compare Maxi CRAS with any other system available. We did, and we're sure you'll find Maxi CRAS to be the most versatile, easy to use Check Register system available — at any price!

ASK RICHARD

Richard Kaplan

There are quite a number of stories I have heard about computers which appear to me to be complete contradictions. For starters, how can I get information from a damaged disk if the information is not there?

When a diskette "goes bad," very often the actual information (the program or the data for the program) is still on the disk perfectly intact. It is only the disk's *directory* that is actually damaged.

What is a disk's "directory?"

The directory is a certain part of the disk which tells the computer *what* information is on the disk, as well as *where* on the disk the information is. Every set of data stored on a disk, such as a BASIC program or data for an accounting program, is called a *file* and is given a name. Whenever you wish to get information from the disk, you must tell the computer the name of the file you wish to load. This filename is listed in the directory, so the computer knows precisely where to find the information.

Is any other information stored in the disk's directory?

Yes. A directory usually includes each file's *length* (how many sectors or granules are allocated to the file), the *type* of file it is (a BASIC program or a data file, for example), the file's *password,* if any, and sometimes the *date* on which the file was created.

Why did you say that a directory "usually" includes this information?

Every different computer has a different format for its directory, which is determined by the operating system in use. (See January's *Ask Richard* for an explanation of operating systems.) The Model II's disk directory, for example, includes the month and day on which every file was first created and last updated, while the Model III directory includes only the month in which the program was created. The Model I TRSDOS does not store any dates, but both the DOSPLUS and NEWDOS/80 operating systems include this feature on the Model I.

Why would a program have a *password,* as you mentioned a while back?

Most computers allow you to specify a password when placing a file onto a disk. To use a protected program or data file, you must first give the correct password. When someone asks the computer for a display of the directory, all filenames are displayed, but the passwords for protected files are *not* displayed. As a result, the information in these files (supposedly) cannot be obtained without the password. The primary use of a file password is to prevent use of the program or data file by unauthorized persons. In addition to passwords for protection of individual files, there is also a password that protects the entire disk (under TRSDOS, if you don't specify a disk password, it will be set as "PASSWORD"). When you attempt to backup a disk, you must enter the correct disk password first. If you don't know the password, you are prevented from copying the disk.

How effective are passwords?

In actual practice, passwords are usually no more than a nuisance. They do little to thwart software piracy.

What do you mean by "piracy"?

A software *pirate* is a person who illegally copies software made by other people and trades or sells these programs.

Why do you say that passwords are not very effective?

Passwords can be *very* easily bypassed with only minimal knowledge of computers. The technical details of software protection are beyond the scope of this feature, but suffice it to say that numerous publications have printed specific instructions for identifying a disk's password.

Then are there *any* effective means of software protection?

No protection method is 100 percent safe. If a pirate is determined to copy a particular program, there is no way to stop him. However, there are many protection schemes (again, beyond the non-technical nature of this series) which are effective against all but the most devoted pirate. My argument is simply that Tandy's standard TRSDOS passwords are, in my opinion, an ineffective nuisance.

Getting back to my first question, how can I "recover" information from a defective disk if the information is not there?

When a disk "goes bad", usually only the directory is defective. This means that the information really is on the disk, but the computer doesn't know where to find it. Usually the directory is the first part of a disk to become unreadable, since by its very nature it is accessed more frequently than any other part of the disk.

In order to recover a defective disk, a "utility program" is used which can look at or change any of the thousands of characters on a disk. A skilled (the term *skilled* is emphasized here) programmer can either replace the defective directory with a new one, or he can tediously find the needed information on the disk (without the aid of the directory), tell the computer where the information is, and have the computer copy the information to a new disk.

I have heard it mentioned that the Model II has an "alternate directory." What does this mean?

All Model II disks created with the TRSDOS operating system have two distinct directories — the *primary directory* and the *alternate directory.* Whenever new information is placed onto a Model II disk, both directories are updated.

Then since the Model II has two different directories, shouldn't bad disks on this computer be very rare?

In theory, yes. In actual practice, *no.* For some reason (unknown to me) TRSDOS does not ever actualy *use* the alternate directory, even if the primary directory goes bad. It seems almost pointless to even have this second directory.

Then does this alternate directory serve any purpose at all?

Yes, it does. Independent software developers have

recognized the need to utilize this second directory. Racet Computes, for example, manufactures a utility program (Mod II Utility Package) which can be used to use copy the alternate directory of a bad disk onto a new disk. To perform this task requires a good knowledge of disk locations, but it is feasible.

Getting on to a different topic — which seems to defy logic — I recently purchased an accounting program for two disk drives which could not store as much information as I need to store. I figured that four disk drives would enable me to store twice as much data as two drives would, but I found this is not true. Why?

This is beyond doubt the most common misconception among persons whom I have spoken to. Generally, adding extra drives will not increase the capacity of commercial software packages.

A detailed explanation of the programming involved in a situation such as this would be inappropriate in this series. The bottom line, however, is that to write a program which can look at data from more than one disk drive can be very difficult. If a program is not specifically designed for this function, the computer will not know to look on the extra drives.

Then what if I purchased a double-density, quad-density, or double-sided disk drive?

Assuming you purchase a suitable operating system (see the first two installments of *Ask Richard)* these extended capacity disk drives normally *will* function with commercial software packages. There are exceptions, such as programs which must store information in memory to sort data. To be absolutely certain that a given program will operate in a non-standard configuration, you must call the product's manufacturer.

I know that you mentioned double-density and double-sided disk drives last month. But what are *quad-density* drives?

As the name would imply, quad-density drives can store approximately four times as much information as can standard Model I single-density drives, In practice, they are used more often on the Model III than on the Model I, though someone who requires a quad-density drive normally can absorb the extra expense and purchase a hard disk drive.

What is a *hard disk drive?*

A hard disk drive can store millions of characters of information, as compared to a floppy drive, whose capacity is measured in thousands of characters of storage. A typical floppy disk drive might store 150,000 characters, while a hard disk drive might store 12 million characters, or *bytes.*

Hard disks are usually permanently sealed. That is, hard disk drives do not have media which can be freely inserted and removed, as floppy diskettes are. Some hard drives have replacable "disk packs," but most hard disk drives do not.

Are there any other advantages to a hard disk drive other than storage capacity?

Yes. Hard drives usually operate at much faster speeds than floppy drives.

If I purchase a hard drive, should I sell my existing floppy drives?

Unquestionably NO!!

Why should I have *both* a hard drive and a floppy drive?

First of all, software obviously cannot be purchased on a hard disk. This means that you must always have a floppy drive to load a commercial software package. The program can then be "downloaded" (transferred) to the hard disk drive (as long as the floppy disk program is not protected or passworded!).

Another reason for a hard disk drive owner to retain his floppy drives is for backing up data files. If you run your business on a hard disk drive, you must make periodic copies of your data in case it is destroyed, or in case the hard drive has to be sent in for repair. It typically takes 30 to 100 floppy diskettes to back up a full hard disk drive, but this inconvenience far outweighs the consequences of losing a company's complete set of financial information.

Speaking of hard disk drives, I have heard the term "Winchester" used quite often to describe hard disk drives. What does this mean?

Winchester is a code name IBM used for its research project in developing a specific type of disk drive. A hard drive which utilizes Winchester technology has a "read/write head" (like the tape heads in a tape deck) which rides only a few thousands of an inch above the surface of the rapidly spinning disk. Today, the term "Winchester" is used generically by virtually every manufacturer except (can you figure this one out?) IBM.

Moving on to a different topic, I received a diskette in the mail which was damaged when it arrived. I can't figure how this could have happened, since it was completely insulated with aluminum foil when it arrived.

The answer to this lies right in the question. The insulation" of the aluminum foil is exactly what damaged the disk. Unfortunately, I have seen many customers mail disk to Computronics for technical assistance with this alleged protection.

Aluminum foil wrapped around a disk acts as a terrific conductor of electricity. The result is that the slightest bit of static electricity or other current will often destroy all data on the diskette. The solution is simply never to wrap disks in aluminum foil.

A harmless little spark of static electricity can destroy data?

Static electricity is *not* harmless to computers. Static electricity can not only destroy data on a diskette, it can also destroy computer circuits. In general, if a computer disk or an integrated circuit receives a jolt of static electricity, it will be completely destroyed. Integrated circuits (or "chips") subjected to this effect usually must be replaced. Disks are usually re-usuable, but all data is usually lost.

Is there any way to avoid static electricity?

There is one technique in particular which is quite effective in combatting static electricity. You should always touch a piece of metal, such as a filing cabinet, before touching a disk or, most importantly, a computer circuit or "board." If this is done, you will almost always avoid the static discharge. Another method (more expensive) is to purchase a grounded anti-static floor mat and place it on the floor in front of the computer. These mats are available from most computer supply companies.

What is a "board", which you mentioned before?

A "board" (sometimes called a "card") is a collection of many integrated circuits mounted together on a printed circuit board, which can then be plugged as a single unit into a "card slot" in the computer. Many computers, including the TRS-80, have provisions for installing boards to add on new functions. An "RS-232" board can, for example, be installed in the Model I to allow for communication with other computers. A board can be purchased for the Model III which will enable CP/M to be used.

What is CP/M?

CP/M is a special operating system (see the first two installments of *Ask Richard*) which permits the owner of a certain computer to write programs which can also run on thousands of other computers (for more information, see "Program Conversion" in issue number 46, June, 1982).

You mentioned *RS-232*. What is this?

RS-232 refers to a standard adopted long ago for communications between computers. (232 is simply the number assigned by a board of standards to this particular communications standard.) An RS-232 "port" is a connector which contains 25 pins. When a matching cable (called a "DB-25" cable) is hooked up to these pins, information can be transferred between computers.

When you mentioned static electricity and boards before, you mentioned that boards can easily be damaged by static electicity. Are they fragile in other ways, as well?

Yes, they are. Whenever you install a board in a computer, you must always turn the computer's power off first. Failure to do this will almost always result in a completely ruined circuit board.

I have heard that it is advisable to keep my computer on all the time. Is this really true?

Although I cannot confirm this information completely, I have been told this by a number of sources, including a Tandy repairman who fixes the computers at Computronics. According to this repairman, most of the wear and tear on a computer comes not from usage, but from voltage surges caused by turning the machine on and off. It certainly is conceivable that this is a good idea, but the Model I and III can be damaged from excessive heat from the power supplies unless a fan is used, and even though the Model II has its own built-in fan, that fan might burn out from continuous use, and the noise level would be annoying.

Speaking of electricity, what are "voltage surges"?

A voltage surge is a burst of electricity which occurs frequently in power lines. Very often, these surges can cause information to be lost or scrambled in computer memory (RAM), or they can cause disks to be damaged if the disk drive is in use when the voltage surge occurs.

Is there anything I can do to prevent voltage surges?

Voltage surges cannot be prevented, but you can avoid their harmful effects on your computer. The simplest solution is to purchase a *line filter*, which can significantly lessen the interference from electrical lines. The best solution, however, is the installation of a *dedicated line*.

What is a "dedicated line"?

A dedicated line is a power line within your home or business which is specifically and solely devoted you

continued on page 59

THREE PROGRAMS AND MORE BRAIN SURGERY

Gordon Speer

MORE BRAIN SURGERY
Lower-case Letters

When I operated successfully on the memory of my Model I (December 1982, page 28) by simply exchanging memory chips until the faulty one was located (like Christmas tree lights), I began to wonder if I might also add lower-case letters to make word processing easier. I actually have another keyboard which has the "factory mod" lower-case, so I haven't been doing without, but I wanted both keyboards to behave the same way. There was a combination of courage, confidence, and experience at work here. I suspect that having had the courage to attempt the "surgery" gave me the experience, and now I had the confidence to try some more. I studied some old magazines and references, and phoned some people to find out what was necessary to do the conversion. I even took the keyboard with the factory-modified lower-case apart to see if the Tandy method was the same as the one I was about to attempt (it wasn't).

Model-I computers were originally sold without lower-case capability. To add this feature, one inexpensive chip must be added to the keyboard to provide the last bit to the character generator memory. (The wizards of Fort Worth finally saw fit to include it when they changed the keyboard styles later). The eight bits held by the character memory chips (one each) supply the byte which determines which character is produced by the character generator. Seven bits can produce 128 characters, but 8 bits can handle up to 256. When they included graphics in the original 128 characters, they had to leave out the lower-case.

The old character generating chip actually has the lower case letters on it, but they do not have "descenders" on the g, j, p, q, and y, so they don't look very nice if you are used to the later style. Converting your old keyboard to modern lower-case, then, will involve removing the old character generator chip, adding a low-profile socket, cutting one circuit trace on the board and soldering a few small wires, at a total cost of about $25. If your soldering technique is not up to par you have several options. You might forget the whole thing, or find a friend who solders, or get an old circuit board from someplace and practice (your soldering technique) until you are expert enough to do the job.

I ordered a CG-1 character generator from KSG in Alabama and followed Steve Gilbert's excellent directions, and, just as he promised, the new lower-case letters look exactly like the ones in the factory mod. The biggest problem was removing the old character generator and installing the low-profile 18 pin socket to hold the new one. You might cut off the pins and remove them one at a time. I invested in a big DIP removing heater that goes in my soldering iron and melts all 18 connections at the

same time. You still need about four hands to do the job, or a spring-loaded extractor. I bought the extractor from TRS, but had to grind the hooks to get them to fit under the ends of the DIP and add some rubber bands to keep the hooks in place.

MICHIGAN and GRAPHICS for EPSON

Those of you who have EPSON printers with the GRAFTRAX option will like these two programs. If you don't have a graphics printer, you may as well skip these, although other printers may be similar. (Write for changes if you have an NEC.)

MICHIGAN is an example of a six-line Old English style capital M, followed by the letters "ichigan" in double wide print to illustrate how graphics and normal printing may be combined for stationery printing, trademarks, etc. I have always felt that one good example is better than a whole textbook of theory. The data statements in the MICHIGAN program contain the binary codes of the dot locations used to make the "M". The NEC lineprinter works similarly, but the binary codes are reversed.

GRAPHICS is the utility program that figured out all the date for MICHIGAN. You don't really want to do that by hand! This is a simple ETCH-A-SKETCH™ (Fisher-Price) which allows you to draw any diagram on your screen, and then use your TRS-80 to calculate and print out the data statements necessary for a graphics program, such as MONOGRAM. It uses the screen to represent a 42 wide by 48 high dot graphics tablet, roughly 12 times wider and 10 times higher than the actual printed output will be. To use it you either play with it freehand, or sketch your pattern on the screen with a washable felt pen. Then use the four arrows to move the flashing cursor around, pressing S to set, and C to clear the pattern you are designing, until you like what you have. Then press P and all the necessary data statements will be printed on your EPSON printer.

Send me any original designs you come up with, and we'll publish the best ones (and maybe even send you a little prize!).

```
10 'MICHIGAN - EPSON GRAFTRAX example
20 CLEAR 1000
30 LPRINT CHR$(27)"A"CHR$(8);   '8/72 INCH LINEFEEDS
40 FOR L=1 TO 6                 'LINES
50 LPRINT CHR$(27)"K"CHR$(42)CHR$(8);  'GRAPHICS MODE
60 REM 42 = columns, 8 = not beyond 256 columns (9+ = beyond)
70 FOR C=1 TO 42                'COLUMNS
80 READ X                       'CODES FOR DOT COLUMNS
90 IF PEEK(14312)<>63 THEN 90   'PRINTER NOT READY
100 POKE 14312,X                'SEND CODE TO PRINTER
110 NEXT C
120 IF L=5 THEN LPRINT CHR$(14)"ichigan";
130 LPRINT                      'NEW LINE
140 NEXT L
150 LPRINT CHR$(27)"2"          'NORMAL LINE SPACING
160 DATA 0,0,0,1,3,7,15,31,63,124,254,127,63,31,15,7,3,1,1,3,
7,14,31,60,126,255,127,63,31,15,7,3,1,0,0,0,0,0,0,0,0,0
170 DATA 0,3,255,255,255,255,255,255,255,0,0,0,128,192,255,
255,255,255,255,255,0,0,255,0,0,0,128,192,255,255,255,255,
255,255,120,48,0,0,0,0,0,0
180 DATA 124,248,240,255,255,255,255,255,255,132,132,132,132,
132,255,255,255,255,255,255,132,132,255,0,0,0,0,0,255,255,
255,255,255,255,0,0,0,0,0,0,0,0
190 DATA 0,0,0,255,255,255,255,255,255,0,0,0,0,0,255,255,255,
255,255,255,0,0,255,0,0,0,0,0,255,255,255,255,255,255,0,0,0,
0,0,0,0,0
200 DATA 0,1,7,255,255,255,255,254,248,0,0,0,0,1,255,255,255,
255,255,255,1,0,255,0,0,0,0,0,255,255,255,255,255,255,
0,0,0,0,0,0,0,0
210 DATA 127,249,240,224,192,128,0,0,0,0,0,0,0,0,128,192,224,
240,248,252,254,255,254,60,24,0,0,64,224,240,248,252,254,255,
126,60,24,0,0,0,0,0
```

```
10 'GRAPHICS
20 'PREPARES DATA FOR EPSON GRAFTRAX
30 'Use ARROWS to change POSITION of the CURSOR on the screen
40 'Screen is approximately 10 TIMES the size of printer
output
50 'Type S and C to SET and CLEAR the screen pattern
60 'When pattern is complete type P to output data to printer
70 CLS:CLEAR 1000
80 A$=INKEY$                    'STROBE THE KEYBOARD
90 RESET(X+1,Y)                 'CURSOR OFF
100 IF A$=CHR$(91) THEN IF Y>0  LET Y=Y-1:IF POINT(X,Y+1)=-1
THEN SET(X+1,Y+1)
110 IF A$=CHR$(10) THEN IF Y<47 LET Y=Y+1:IF POINT(X,Y-1)=-1
THEN SET(X+1,Y-1)
120 IF A$=CHR$(9) THEN IF X<123 LET X=X+3:IF POINT(X-3,Y)=-1
THEN SET(X-2,Y)
130 IF A$=CHR$(8) THEN IF X>0   LET X=X-3:IF POINT(X+3,Y)=-1
THEN SET(X+4,Y)
140 SET(X+1,Y)                  'CURSOR ON
150 IF A$="S" THEN SET(X,Y)  :SET(X+2,Y)
160 IF A$="C" THEN RESET(X,Y):RESET(X+2,Y)
170 IF A$="P" THEN 200
180 GOTO 80
190 '
200 'PRINTING ROUTINE
210 LPRINT CHR$(15);            'COMPRESSED PRINTING MODE
220 FOR R=0 TO 40 STEP 8        'STARTING ROW
230 LPRINT "DATA ";
240 FOR X=0 TO 123 STEP 3       'HORIZONTAL POSITION
250 FOR Y=0 TO 7                'VERTICAL POSITION
260 IF POINT(X,Y+R)=-1 THEN T=T+2>(7-Y)       'EPSON GRAPHICS
270 NEXT Y
280 LPRINT USING"###";T;
290 IF X<123 THEN LPRINT",";
300 T=0
310 NEXT X
320 LPRINT:LPRINT
330 NEXT R
340 A$=""
350 GOTO 90
```

TYPE

TYPE is a touch typing practice program. It displays a letter in the center of the screen. When you type the correct letter it changes to a different letter. Every twentieth one you get an estimated speed to tell you how well you are doing. It is a tough way to type, because you can't look ahead, but it is a good way to learn the location of the letters on the keyboard.

```
10 'TYPE
20 CLS
30 PRINT CHR$(23);              'WIDE LETTERS
40 R$=CHR$(RND(26)+64)          'RANDOM LETTERS
50 IF R$=A$ THEN 40             'NO REPEATS
60 PRINT @ 476,R$;              'MIDDLE OF THE SCREEN
70 A$=INKEY$                    'STROBE THE KEYBOARD
80 N=N+1                        'TIMER COUNTING ALL THE TIME
90 IF A$<>R$ THEN 70            'CORRECT KEY PRESSED YET?
```

```
100 L=L+1                          'COUNT CORRECT RESPONSES
110 IF L<20 THEN 150               '20 CORRECT YET?
120 PRINT @ 24,INT(30000/N)        'PRINT WORDS PER MINUTE
130 N=0                            'RESET THE CLOCK
140 L=0                            'RESET CORRECT KEY COUNTER
150 GOTO 40
```

Gordon Speer
3304 Woodlawn Road
Sterling, IL 61081 ■

ARRAY OF HOPE FOR BASIC PROGRAMMERS
continued from page 22

where most of them originally appeared. In the journal they were coded in Algol or Fortran. Timings are also given for different data for a number of the sort algorithms.

Sorting by William A. Martin, published in the ACM journal *Computing Surveys,* Vol. 3 No. 4, gives an overview of different sorting algorithms.

Table Lookup Techniques by C. E. Price, also from *Computing Surveys,* Vol. 3 No. 2, gives a good overview of searching methods.

A Comparison of Sorts by John P Grillo in *The Best of Creative Computing Vol. 2* compares the times for 3 sorting algoritms. These are Bubble Sort, Delayed Replacement Sort, and the Shell-Metzner Sort. The conclusion is that Shellsort should be used for all sorts. Quicksort is neither mentioned nor timed.

Typical of the "discovery" of old and well known sorting methods are two articles in the April 1981 issue of *Kilobaud Microcomputing.* Both of them describe versions of the sifting sort or shuttle sort, one of them claiming to make the bubble sort an efficient sorting method. Before people give in to the urge to write an article on a "new" sorting method, they would be well advised to look through the book by Knuth or the one by Lorin. If it is not found in either of these books, then it is possibly new, but it would also be an idea to compare the sorting speed with the Shellsort or quicksort before claiming it as the fastest sort available. This is not intended to discourage people from experimenting with sorting methods, but rather a warning that many others have made similar attempts.

Especially for external sorts, there are large gains to be made in sorting speed by reducing the number of external references, and it is perhaps in this area that the least amount of experimenting has been done. Anyone wanting to write a faster sort is advised to try writing a routine to sort a disk file with the least number of disk accesses, and with the least amount of extra external storage. I hope this series has given the reader some insight into the methods available for sorting and searching, and the relative merits of each of the methods.

Arne Rohde
Pilevej 31
7600 Struer, Denmark ■

BEGINNER'S CORNER

Spencer Koenig

This Month: Your Most Friendly BASIC Editor (continued)

Last time, I was discussing the BASIC editor for the Model I and III. The only aspects I didn't cover then, but will cover now, are the improvements Disk BASIC made available by some of the various disk operating systems around.

The operating systems that I can cover for you start with TRSDOS for the Model III. (Model I TRSDOS Disk BASIC didn't add any features, according to the manual I have.) Also covered in depth are DOSPLUS 3.3, NEWDOS80 Version 2.0, and MULTIDOS. Well, now that I've bitten off a large chunk of information to explain, it's time to get to work.

My favorite operating system is NEWDOS80 Version 2.0. I find it to be a friendly environment in which to work, but at the same time it's also very flexible and powerful. It also has a great deal in common with the other operating systems I'll cover, so let's begin with the NEWDOS80 system. We'll begin by looking at EXAMPLE 1.

****

EXAMPLE 1

NEWDOS8Ø 2.Ø and 1.Ø Disk Basic Commands

COMMAND	RESULTS
, (comma)	This is used to edit a line where the most recent error has occurred. This sets the current line pointer to that line.
Delete or D	Command used to delete a line or lines. Lines cannot be deleted simply by typing in the line number.
. (period)	Used to display the current line.
Down-Arrow	Lists the next line.
Up-Arrow	Lists the previous line.
; (semi-colon) or Shift-Up-Arrow	Goes to the beginning of the program.
/ (slash) or Shift-Down-Arrow	Puts you at the end of the file. If you find that the Shift-Down-Arrow doesn't work it's because you have the new ROM.
: (colon)	This command allows you to scroll up one page (16 lines).
@ (at sign)	Allows you to scroll one page toward the end of the file.

****

As you can see (by omission), your basic BASIC editor is still intact. These additional commands allow you to move around the program for easier editing. This kind of mobility usually comes with screen editors of various types that you see advertised. I've used some of them, and I like them, but if you're not careful you can get yourself into some nasty trouble.

The first command that I have listed is the " , ". When you run the program and an error occurs, the current line pointer within the BASIC interpreter is set to the line with the error. Using the comma command automatically puts you into edit mode on that line. This is a bit different from Level II BASIC as well as NEWDOS80 Version 1.0. In an error situation, Level II and NEWDOS80 1.0 automatically set you in edit mode on that line ready to edit.

The problem with this is that, under certain conditions, the error could have occurred due to a mistake in one of the variables associated with that line. You might have even gotten a syntax error when no syntax error really existed. If you happen to do something wrong and inadvertently edited the line, you will find that the variables will have been reset to zero, leaving you out in the cold.

In response to this situation, the people at Apparat came up with this command, which forced you to go into edit mode only if you deliberately wanted to.

The next command is " . ", and it is almost self-explanatory. If you want to list the line, think of what's at the end of a sentence or line, namely the " . ". Now didn't that work out nicely? (That's not a question, thank you.)

The next two commands, the UP-ARROW and the DOWN-ARROW, display the next line or the preceding line, respectively. Just remember, to go to the next line down, hit the down arrow. If that doesn't work hit the " / " key (if that doesn't work, I'll be very disappointed, and you will be too). The reason the up or down arrows don't work is due to the newer ROMs installed by the big R.S. The NEWDOS people couldn't go around customizing their operating system for everyone, so they simply included an alternative command. I wish everyone could be so clever.

The UP-ARROW command is in the same boat, so to speak, as the DOWN-ARROW command. The alternative is to use the " ; " command. If you look at your keyboard, you'll notice that these commands seem to be centering around one area. With the exception of the regular commands while in edit mode, (A, H, Q etc.). you also have these or their abbreviations. See EXAMPLE 2.

****

EXAMPLE 2

We're still in NEWDOS8Ø 2.Ø or 1.Ø.

COMMAND	RESULT
AUTO or A	Automatic numbering by specified step
EDIT or E	I think that's clear. Don't you??
LIST or L	This lists the program.

This shouldn't be confused with " L " while in EDIT mode, which lists the line you're editing.

****

These commands can be used as long as certain conditions are met. In other words, there are strings attached. (Get it? No, huh? You will.) The command must be the first character input to the computer, but it cannot be followed by an " = " sign. If you do input an

" = " sign, then the computer will think you want to initialize a variable to a specific value, i.e. A = 1. The computer scans the input as a string (now you get it!) and tries to figure out what you want to do according to

The third condition (1: first character input, 2: no " = " sign) is that you can follow the first character with a " . " or a specific number. The number can represent a line you wish to edit, or as with the AUTO command, the start of numbering by a specified step (begin with 2 and increment by 3).

The next topic comes under the heading of block moves. By this I mean blocks of code can be renumbered or moved to new locations with the old locations deleted or not deleted (that is the question). Let's look at EXAMPLE 3.

****

EXAMPLE 3

NEWDOS8Ø 2.Ø and 1.Ø commands

COMMAND	RESULT
DI start,new location.	Delete old line and insert at new line location.
DI .,new location.	The same as above but the period defaults to the last line on the screen or the last line edited.
DU start,new location.	Allows you to copy code to new location without deleting the old location.
DU .,new location.	Same as above and again the period defaults to the last line edited or listed on the screen.

****

I suggest that you do as we did last time and experiment with these commands. Try combining them with the edit commands in some convoluted or permeated fashion (or maybe a facinating rhythm). Will the Abort command work with block copies? (Only my hairdresser knows for sure.) Give it a try and see what happens (that's the command, not my hairdresser).

A subset of block copies is line renumbering, except that it isn't really a subset but rather an "it gets a similar-to" results. By this I mean that you can get the same kind of results copying a block of code from one place to another or by renumbering a block of code. However, the actual uses of these commands will be obvious, and their differences apparent, as you'll see.

Lets look at the options in EXAMPLE 4.

****

EXAMPLE 4

NEWDOS8Ø 2.Ø only

COMMAND

RENUM new starting #,increment step, old start #, old ending # (,U) (,X)

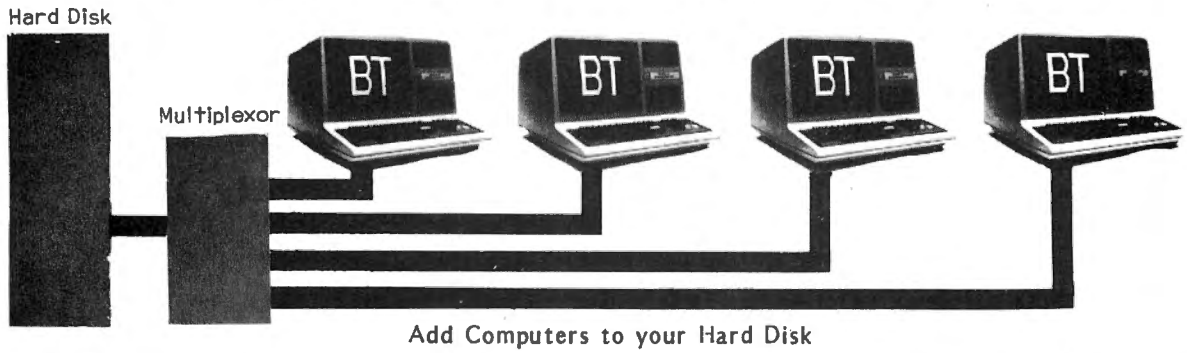
RENUM The whole program is renumbered with a default to an increment step by 1Ø.

continued on page 54

THE BT ENTERPRISES COMPUTER CONNECTION

Multiplexor for your Hard Disk

GET IT ALL TOGETHER



Add Computers to your Hard Disk

Adding a hard disk to your computer is the first step in making your micro a powerful computer. Adding more computers to your hard disk makes it a real system. The Bi-Tech MULTIPLEXOR will do just that. Adding a multiplexor to your hard disk sub-system allows you to share a hard disk among many computers without the need for additional hard disk sub-systems. Adding a Multiplexor will allow users to share programs and data present on the hard disk, making a true DATA BASE available to all users. The Bi-Tech MULTIPLEXOR is capable of operating at a speed that is faster then the hard disk. No waiting as in other systems. Data transfer rates are the same as in a single computer to hard disk sub-system. The Bi-Tech MULTIPLEXOR is totally transparent to the user, software, and hardware. Now available for most computers with a BT Hard Disk sub-system and also for many other hard disk sub-systems.

Multiplex vs. Multiuser

Unlike multiuser systems, which shares the microprocessor, the concept behind multiplexing is to share only the disk storage. This frees each computer to use as much processor time as desired without affecting the other users. The Bi-Tech MULTIPLEXOR links multiple computers to a single hard disk sub-system

Multiplex vs. Network

Unlike networks, which transfer data serially, the parallel transfer rate of the Bi-Tech MULTIPLEXOR is the same as the original hard disk parallel transfer rate. No time consuming serial to parallel conversions necessary. Serial networks require one computer to be the host for the rest of the system. In a Multiplex system any computer can be run totally independent of the other computers. Each user is not affected by a hardware or software failure on another computer.

A network also requires expensive decoders at each station, but a Multiplex station needs only an inexpensive host adaptor to link into the Bi-Tech MULTIPLEXOR.

SPECIFICATIONS

* Single Circuit Board
* No External Power required
* No Speed loss between CPU & Hard Disk
* 4 Users Per Multiplexor
* WD1000/1001 Compatable

5 Meg Sub-system.....$2399.95
10 Meg Sub-system.....$2549.95
15 Meg Sub-System.....$2699.95

PRICES

Bi-Tech MULTIPLEXOR complete.....$795.00
Bi-Tech Model I/III Host adapter.$395.00
Bi-Tech Model II Host Adapter....$575.00
Bi-Tech IBM-PC Host Adapter.....$595.00
Other Computers available soon, CALL

(Sub-Systems include 1 Mod I/III Host Adapter & Dos Plus 4.0 Operating System)

BT ENTERPRISES

B.T. Enterprise Dept 2C
10B Carlough Road
Bohemia, N.Y. 11716
(516) 567-8155 (voice)
(516) 588-5836 (modem)

B.T. Enterprises is a division of Bi-Tech Enterprises Inc.

Dealer Inquires Welcome
Prices Subject to Change
N.Y.S. Residents Add Tax

Orders Only
800 645 1165
N.Y. Call (516) 567-8155

American Express, Carte Blanche, Diners Club, MasterCard and VISA accepted.

THE BT ENTERPRISES COMPUTER CONNECTION

Controller Boards that give you the time of day!

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.

KIT CONSISTS OF:

- The NEW HOLMES Controller Board with real Time Clock & battery included completely assembled & tested
- Switching power supply
- Disk drive mounting brackets
- All cables and hardware
- Instructions with diagrams
- Drives are optional

*Controller Board supports 8 inch external drives and extended density 5-¼ inch drives (dual headed and 80 track)

203104	*Upgrade Kit	$329.50
203114	*Upgrade Kit w/2 40tk	$729.50
203115	*Upgrade Kit w/2 40tk DH	$949.50
203116	*Upgrade Kit w/2 80tk	$949.50
203117	*Upgrade Kit w/2 80tk DH	$1149.50

***INTRODUCTORY SPECIAL — NOW AVAILABLE WITH DOS PLUS 3.4 FREE**

203004	Controller Bd. w/Real Time Clock & Battery	$229.50
200500	Switch Power Supply (3 voltages)	$75.00
203010	Controller Assembly Kit	$50.00
203510	Controller Assembly Kit w/Power Supply	$110.00
213001	16K Memory Kit	$19.95
213002	32K Memory Kit	$34.95

DH — Dual Headed

Add $2.00 Shipping and Handling

B.T. Hard Disk Model III

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 Holmes Engineering Disk Controller System with Real Time Clock 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. Then have installed a Hard Disk Drive (Winchester Technology). When the system is configured, the Hard Disk Drive contains the operating system and becomes the master (primary) drive. The Hard Disk has 40 times the speed of data transfer as compared to a normal floppy disk as well as containing the storage capacity of more than 30 standard floppy disk drives.

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

TRS-80 is trademark of Tandy Corp.

AVAILABLE CONFIGURATIONS

4 x 5	5 Megabyte Formated Storage	$3995.00
4 x 10	10 Megabyte Formated Storage	$4145.00
4 x 15	15 Megabyte Formated Storage	$4295.00

Add $20.00 Shipping & Handling

B.T. Hard Disks

We want you to have all the power that your TRS-80 is capable. These Five, Ten and Fifteen 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!

SYSTEM PRICING

5 Megabyte Winchester Fixed Disk

201505	Model I Version	$2399.95
203505	Model III Version	$2399.95

10 Megabyte Winchester Fixed Disk

201510	Model I Version	$2549.95
203510	Model III Version	$2549.95

15 Megabyte Winchester Fixed Disk

201520	Model I Version	$2699.95
203520	Model III Version	$2699.95

Add $10.00 shipping and handling

BT ENTERPRISES

B.T. Enterprise Dept 2C
10B Carlough Road
Bohemia, N.Y. 11716
(516) 567-8155 (voice)
(516) 588-5836 (modem)

B.T. Enterprises is a division of Bi-Tech Enterprises Inc.

Dealer Inquires Welcome
Prices Subject to Change
N.Y.S. Residents Add Tax

Orders Only
800 645 1165
N.Y. Call (516) 567-8155

American Express, Carte Blanche, Diners Club, MasterCard and VISA accepted.

PRACTICAL BUSINESS PROGRAMS

S. M. Zimmerman and L. M. Conrad

Month #3: TRANS, A Program to Input Transactions

This month we will review the second group of programs in our general ledger set. The first program in this group is TRANS, the program which is the starting point for initializing a new set of data. Next month's program is CPA, C, the program that generates a balance sheet and profit and loss statement.

The last program in the set is TRIAL. This program produces a trial balance for the purpose of aiding in error searches.

The publication schedule for the entire general ledger system is as follows:

No	Month	Program
1.	.....	GLMENU (Controls use of routines).
2.	.....	NAME & START (Initializes chart of accounts and firms name).
3.	.....	TRANSACTION (Inputs monthly transactions).
4.	.....	CPA (Produces Balance sheet & Income and Expense statements, Profit & Loss statement).
5.	.....	UPDATE & YEAR (Changes chart of accounts and performs year end closing tasks).
6.	.....	TRIAL (Produces trial balance).
7.	.....	BALANCE (Balances check book).
8.	.....	MILES & MOVE (Calculates gas mileage and moves files between disks).

The operational procedures of TRANS will be reviewed.

RUNNING TRANS

Check (transaction) number, month+day, account debited, account credited and amount debited, amount credited are input for each transaction. No written description of the transaction was included. The reasons for this decision were; (1) The written information is already available; (2) To add more information would increase the difficulty of inputing data; (3) More data would take up more disk and internal space for storage.

The program starts with the following:

TRANSACTION PROGRAM
DATA NAME?

This is the point where a file is first given its name. Keep a single corporation on individual disks and use the name of the month for the transaction set being worked on. When you name the old balance sheet information use the month name plus a two (2) or the word BAL. If this is MAY, answer MAY to the above question.

MAXIMUM NUMBER OF TRANSACTIONS?

You may wish to experiment to find the exact limitation of your equipment with the number of items in your chart of accounts. The actual capacity of any given hardware system is a function of the operating system being used and the size of the chart of accounts.

If you happen to have a month with an exceptionally large number of transactions, it is always possible to break the time period down into weeks or even days to make the task fit the limitations of the equipment.

If you are in the middle of a run and find you did not provide enough room, you can record your data to the point where your are and then go through GLMENU and start over again with a greater number of transactions.

In order to see what happens when you run out of memory, answer 1000 to the above question. You will note that the computer tells you an error has been made and prints out a menu of choices you are not yet ready for. It gives you a chance to save files or to return to GLMENU, etc. Since you do not want any of these options, hit the BREAK key and then type RUN and hit ENTER.

This time answer 250 to the number of transactions question. This should fit into the limits of most systems and allow you to continue without error.

SELECT DISK OR KEY INPUT (D OR K)?

If you are starting a new month your input would very likely be started from the keyboard. We will assume you answered K to the above to illustrate this option.

CK NO.,MONTH+DAY,ACCOUNT-DEBIT,CREDIT,
AMOUNT-DEBIT,CREDIT

There are six items of input. All must be completed. The sum of the debits must equal the sum of the credits as in all accounting systems. If you have a compound entry, such as would be the case with a mortgage payment which includes both interest expense and reduction of loan two entry lines are required. When you have completed a series of lines and wish to see what you have done, or if you have made an error and wish to correct it, input a blank line or a series of zeros.

INPUT LINE NUMBERS TO BE LISTED OR 0,0 WHEN COMPLETE?

If you already had a file on disk, the above question would have been your first decision. We will assume a series of lines have been listed as 0,0 and was 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?

At any time you wish you may select any of the above options. Be careful to save your files before you select option -7 to return to GLMENU.

If you have problems with electric power failure, you should save your file from time to time. It is recommended you save your files on two disks for backup purposes. Also if you get the following message:

AN ERROR HAS OCCURRED SAVE YOUR FILES—ON NEW DISK IF NECESSARY

This means some type of error has occurred and you

have a problem. A fail safe system has been built into the program which allows you to save the work you have completed. If you are returned to the menu from the -2 save option with the message that an error was made it is likely the disk you are using is full or you have disk trouble. We keep a stand by disk just for this problem. What you must do is put your standby disk in the disk drive and then save your data. You may then KILL any unwanted files on your data disk and use MOVE or the COPY option as discussed above to get the file onto the disk where you want it.

The transaction menu gives you a number of options relative to inserting and deleting data. These options (-1 to -4) are self-explanatory. Option -5, the summing of the debits and credits, is like having a mini-trial balance program. The debits must equal credits. You may check for this at any time.

If you selected the -6 option the following would happen.

```
LINE COUNTER(Y/N)?
or SETUP PRINTER ENTER WHEN READY?
```

If you have an operating system with a line counter the first question about a line counter is needed. If not the second question is useful. Line 410 of our program contains the first question. If you do not have a line counting operating system subsitue the second question and leave out the CMD "FORMS(T)" instruction.

```
INPUT DATE OF RUN?
INPUT ORIGINAL BALANCE?
```

Once these two questions have been answered you will then given the option of which accounts to print out.

```
PRINTER MENU
        1       ALL ACCOUNTS
        2       SELECTED TRANSACTIONS
        3       SELECTED DATES
        4       SELECTED ACCOUNTS
?
```

This ability allows you to create a specialized journal as needed. We answered 1 to the above to obtain all the accounts for our sample run.

You will obtain a printout of the transactions you have just placed into the computer. The objective of requiring the inputing of the original balance is to aid in bank reconciliation procedures. You will find your computer will not always agree with the banks computer. The most common error we have made when using the system has been typing errors. The printer copy is a great aid in finding these errors.

Do not forget to record your results before using the -7 option and returning to GLMENU.

EXAMINING THE PROGRAM

In line 10 you will find the ONERROR GOTO 720 statement. During the process of debugging your copy of the program, it is recommended that you delete this statement. You will have trouble finding and tracing your errors if you do not do so. After you are satisfied the program is running correctly, be sure to type the ONERROR GOTO 720 statement into line 10. The objective of this statement is to allow the user the

opportunity to save his work if the system should fail due to a full disk, or for any other reason.

Lines 20 through 70 are designed to input preliminary information into the program. If the task is to update an existing file then lines 80 and 90 are used to bring this file into the computer.

Lines 100 through 140 are used to input new transactions. The task is terminated if the defined capacity is reached or if the user terminates the sequence by inputting a series of zeros.

Lines 150 and 160 are used to determine what lines are to be listed on the screen for review purposes. The task of line listing continues in lines 170-180. In line 190 the user is given a menu of alternate acts depending on the results of the review of the data.

Lines 200 through 250 routes the program flow according to the selection make in the above menu. If the user selected -2 for recording the program flows goes to line 680 where this task is performed, if the user selected -6 for printing the program flow goes to line 410 for this task. If the user selected -3 then the flow goes to 260 where this job is performed.

Upon the completion of each task the program returns to the menu in line 190.

```
1Ø CLEAR 8ØØ:ONERROR GOTO 72Ø  :REM    "TRANS"
2Ø A$="###   ####   ####   ###      ###   #####.##
   #####.##     #####.##"
3Ø CLS : PRINT "TRANSACTION PROGRAM ": INPUT "DATA NAME";F$:
INPUT "DISK";X$
4Ø FX$=F$+":"+X$: PRINT : INPUT "MAXIMUM NUMBER OF
TRANSACTIONS";D: DIM A%(D,4),A#(D,2)
5Ø INPUT "SELECT DISK OR KEY INPUT (D OR K)";T$
6Ø IF T$="K" THEN 11Ø
7Ø IF T$<>"D" THEN 5Ø
8Ø I=1:OPEN"I",1,FX$
9Ø 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)=Ø THEN 15Ø
1ØØ I=I+1:GOTO 9Ø
11Ø PRINT "CHECK NO., MONTH+DAY,ACCOUNT-DEBIT,CREDIT,
AMOUNT-DEBIT,CREDIT":K=1
12Ø FOR Z=K TO D:K=Z
13Ø INPUT A%(Z,1),A%(Z,2),A%(Z,3),A%(Z,4),A#(Z,1),A#(Z,2):
IF A%(Z,1)=Ø THEN 15Ø
14Ø NEXT Z
15Ø CLOSE 1: INPUT "INPUT LINE NUMBERS TO BE LISTED OR Ø,Ø
WHEN COMPLETE";A,B
16Ø IF A=Ø THEN 19Ø
17Ø CLS: PRINT "NO   TRANS    DATE ACC-DEBIT ACC-CREDIT
AMT-DEBIT  AMT-CREDIT"
18Ø FOR I=A TO B: PRINT USING A$;I,A%(I,1),A%(I,2),A%(I,3),
A%(I,4),A#(I,1),A#(I,2): NEXT I
19Ø INPUT "SELECT LINE NUMBER TO BE CHANGED, Ø TO LIST, -1
TO ADD DATA, -2 TO RECORD, -3 TO INSERT, -4 TO DELETE, -5
TO SUM DEBITS/CREDITS, -6 FOR PRINTING OR -7 TO RETURN TO
GLMENU";P
2ØØ IF P=-7 THEN LOAD"GLMENU",R
21Ø IF P=Ø THEN 15Ø
22Ø IF P=-1 THEN PRINT "CK NO,MO+DAY,ACCOUNT DEBIT-CREDIT,
AMOUNT DEBIT-CREDIT":GOTO 12Ø
23Ø IF P=-2 THEN 68Ø
24Ø IF P=-6 THEN 41Ø
25Ø IF P<>-3 THEN 32Ø
26Ø INPUT "NUMBER OF INSERTED LINE Ø TO DEFAULT";P:
IF P=Ø THEN 19Ø
27Ø K=K+1:W=Ø:FOR I=D-1 TO P STEP-1:IF A%(I,1)=Ø THEN 3ØØ
28Ø IF W=Ø THEN W=1: FOR J=1 TO 4: A%(I+2,J)=A%(I+1,J):
NEXT J: FOR J=1 TO 2:A#(I+2,J)=A#(I+1,J): NEXT J
29Ø FOR J=1 TO 4:A%(I+1,J)=A%(I,J): NEXT J:A#(I+1,1)=A#(I,1):
A#(I+1,2)=A#(I,2)
3ØØ NEXT I
31Ø GOTO 4ØØ
32Ø IF P<>-4 THEN 36Ø
33Ø INPUT "LINE NUMBER TO DELETE Ø TO DEFAULT";P: IF P=Ø
THEN 19Ø
34Ø K=K-1: FOR I=P TO D-1: IF A%(I,1)=Ø THEN FOR J=1 TO 4:
A%(I,J)=Ø: NEXT J: FOR J=1 TO 2:A#(I,J)=Ø: NEXT J: GOTO 18Ø
35Ø FOR J=1 TO 4: A%(I,J)=A%(I+1,J): NEXT J: A#(I,1)=
A#(I+1,1): A#(I,2)=A#(I+1,2): NEXT I: GOTO 18Ø
36Ø IF P<>-5 THEN 4ØØ
37Ø SD#=Ø: SC#=Ø: FOR I=1 TO D: IF A%(I,1)=Ø THEN 39Ø
38Ø SD#=SD#+A#(I,1):SC#=SC#+A#(I,2): NEXT I
39Ø BA$="###,###,###.##":PRINT"SUM DEBITS ";: PRINT USING
BA$;SD#;: PRINT "  SUM CREDITS ";: PRINT USING BA$;SC#:
GOTO 19Ø
4ØØ 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 18Ø
41Ø INPUT "LINE COUNTER (Y/N)";LC$: IF LC$="Y" THEN CMD
"FORMS(T)"
42Ø XX$=" NO. CK NO. MO+DAY  DEBIT  CREDIT  $DEBIT
       $CREDIT       $BALANCE"
43Ø INPUT "INPUT DATE OF RUN XX/XX/XX";DA$
44Ø Z$="      BALANCE    ##########.##"
45Ø INPUT "INPUT ORIGINAL BALANCE";BA#:SD#=Ø:SC#=Ø
46Ø LPRINT" ": LPRINT "   DATE    ";DA$: LPRINT USING Z$;BA#
47Ø LPRINT XX$
48Ø CLS : PRINT "PRINTER MENU": PRINT "     1     ALL
ACCOUNTS": PRINT "     2     SELECTED TRANSACTIONS": PRINT
"     3     SELECTED DATES" : PRINT "     4     SELECTED
ACCOUNT": INPUT GH%
49Ø IF GH%=1 THEN 56Ø
5ØØ IF GH%<>2 THEN 52Ø
51Ø INPUT "INPUT MINIMUM AND MAXIMUM TRANSACTION
NUMBERS";SS%,BB%:GOTO 56Ø
52Ø IF GH%<>3 THEN 54Ø
53Ø INPUT "INPUT EARLYEST DATE AND LATEST DATE (MODA)";
SS%,BB%: GOTO 56Ø
54Ø IF GH%<>4 THEN 48Ø
55Ø INPUT "ACCOUNT YOU WISH TO EXAMINE";SS%
56Ø FOR I=1 TO D: IF A%(I,3)=1 THEN BA#=BA#+A#(I,1)
57Ø IF A%(I,4)=1 THEN BA#=BA#-A#(I,2)
58Ø IF A%(I,1)=Ø THEN BA$="###,###,###.##": LPRINT "SUM
DEBITS ";: LPRINT USING BA$;SD#;: LPRINT "  SUM CREDITS
";:LPRINT USINGBA$;SC#:GOTO 19Ø
59Ø IF GH%=1 THEN 65Ø
6ØØ IF GH%=2 AND SS%<=A%(I,1) AND BB%>=A%(I,1) THEN 65Ø
61Ø IF GH%=3 AND SS%<=A%(I,2) AND BB%>=A%(I,2) THEN 65Ø
62Ø IF GH%=4 AND A%(I,3)=SS% THEN 65Ø
63Ø IF GH%=4 AND A%(I,4)=SS% THEN 65Ø
64Ø GOTO 66Ø
65Ø LPRINT USING A$;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
66Ø NEXT I: LPRINT "SUM DEBITS ";: LPRINT USING BA$;SD#,
"  SUM CREDITS  ";: LPRINT USING BA$;SC#
67Ø GOTO 19Ø
68Ø OPEN "O",1,FX$: FOR IX=1 TO D: FOR IQ=1 TO 2:
DZ#=INT(1ØØ*A#(IX,IQ))/1ØØ: DX#=A#(IX,IQ)-DZ#:A#(IX,IQ)=DZ#:
IF DX#>.ØØ5 THEN A#(IX,IQ)=A#(IX,IQ)+.Ø1
69Ø NEXT IQ,IX:IX=Ø
7ØØ 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 7ØØ
71Ø CLOSE 1: GOTO 19Ø
72Ø RESUME 73Ø
73Ø PRINT "AN ERROR HAS OCCURRED SAVE YOUR FILES--ON NEW DISK
IF NECESSARY": PRINT: GOTO 19Ø
```

continued on page 62

DOES STRING COMPRESSION HAVE YOU TIED UP IN KNOTS?

LET TRASHMAN CLEAN UP THE MESS!

TRASHMAN is a machine language utility for the TRS-80 Models I and III. It was written by Glenn Tesler, the author of FASTER, and can reduce BASIC's string compression time by 95% (see table below).

WHAT'S STRING COMPRESSION?

When a BASIC program changes a string (words, names, descriptions), it moves it to a new place in memory, and leaves a hole in the old place. Eventually, all available memory gets used up and BASIC has to push the strings together to free up some space. This takes time. Lots of time. The computer stops running for seconds or minutes, and you may even think it's "crashed". The keyboard won't work, and until all the strings have been collected, you just have to sit and wait. Then things run for a while, until string compression is needed again. And again.

If you're using your computer for business, that wastes your money. If you're using it personally, it wastes your time.

WHAT'S THE SOLUTION?

As soon as you start using TRASHMAN, those delays almost disappear. It uses less than 600 bytes of memory, plus 2 bytes for each active string. It works with other machine language programs and with all major operating systems. It's easy to use, comes with complete instructions, and can be copied to your own disks.

WHAT'S THE CATCH?

If a BASIC program uses only a few strings, very little time is wasted in string compression, and TRASHMAN won't be helpful. But, if hundreds of strings, including large string arrays, are used, TRASHMAN is just what you need.

# STRINGS	SECONDS DELAY NORMAL	SECONDS DELAY TRASHMAN	PERCENT IMPROVEMENT
250	11.8	0.7	94
500	45.8	1.6	96.5
1000	179.6	3.5	98
2000	713.2	7.8	98.9

TRASHMAN is available on disk for just **$39.95.**

(All timings done on TRS-80 Model I. Model III 15% faster, but pct. improvements identical. Listing of timing program available on request.)

SAVE TIME WITH FASTER

"FASTER" speeds up most TRS-80 BASIC programs by 20-50%. It's helped hundreds of satisfied people and it can help you. Detailed instructions make it easy to use. FASTER analyses your BASIC programs *while they run*, then displays a simple change, usually one line, that sequences program variables so the ROM will find them faster.

You can use FASTER to speed up programs you've bought, as well as programs of your own. Since it isn't a compiler, your BASIC programs can be read and changed afterwards. FASTER works on business programs, models, and games. The more complex your program, the better the results.

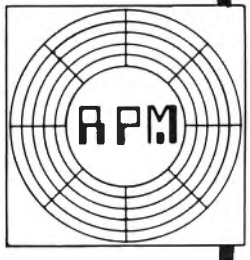
Does FASTER really work? Yes! Just check the reviews in *Personal Computing*, May, 1981, p. 116: **"FASTER is effective and easy to use"**; *80 U.S. Journal*, April, 1982, p. 106: **"I recommend FASTER to everyone"**; and *80 MICRO* (April, 1982, p. 40): **"If you...would like a significant increase in the run-time speed, then buy FASTER."**

FASTER runs on the TRS-80 Models I and III, 16-48K tape or disk, and all major operating systems. **$29.95**

"QUICK COMPRESS" takes only 276 bytes of memory, and removes the blanks and remarks from even the largest BASIC program in less than 3 seconds. It produces smaller, faster programs without altering their logic. **$19.95**

SPECIAL: FASTER and QUICK COMPRESS: **$39.95**

ERRATIC DISK DRIVES?



You can avoid unnecessary disk errors and repair bills by using **RPM.** This easy-to-use program measures the rotational speed and fluctuations of your disk drives, and warns you if they are running **too fast, too slow, or unevenly.**

Incorrect or erratic speed is a common cause of unexplained disk errors and loss of data. RPM's documentation explains how to **detect** *and* **correct** these problems quickly and easily. As *80 MICRO* (April, 1982, page 41) said: **"If your drives have problems I recommend RPM before paying to get it repaired."**

RPM is supplied on diskette for the TRS-80 Models I and III. We suggest you order a copy *before* you need it.

$24.95

ORDER FROM YOUR LOCAL SOFTWARE DEALER, OR CALL NOW, TOLL-FREE:

(800) 824-7888, Operator 422
CALIF: (800) 852-7777, Oper. 422
ALASKA/HAWAII: (800) 824-7919

FOR TECHNICAL INFORMATION CALL:
(213) 764-3131, or write to us.

PROSOFT®

Dept. D, Box 560, No. Hollywood, CA 91603

TERMS: We accept VISA, MasterCard, checks, or even cash. Please add $2.00 shipping/handling within U.S.A. or Canada, and $5.00 overseas. C.O.D. charge is $2.00 in U.S. only. We ship within one day of receiving orders.

MEMORY DISPLAY

Ken Vaske

This program will display memory contents on the video display screen. 256 bytes can be displayed at a time, with 16 lines of 16 bytes each line. The display is arranged in seven columns. The left column contains the address of the first byte in the line, in decimal notation. The next four columns display the contents of 16 bytes of memory in hex format, with four bytes per column. The next column displays the same 16 bytes in character format, with ASCII codes less than 32 and greater than 127 represented by a period. The last column contains the address of the first byte in the line, in hex notation.

In operation, the program will request a starting address from which to start the memory display. This address may be entered in either decimal or hex notation. Hex notation requires an "X" in the first position. After a screenful has been displayed, the program will pause and wait for a key to be pressed. Pressing "F" will cause the display of the next 256 bytes of memory. Pressing "B" will cause the display of the previous 256 bytes of memory. Pressing "E" will end processing. Pressing any other key will cause the program to request a new starting address.

If a printer is available, an optional hard copy may be requested. Hard copy will be printed in the same format as the screen display. When hard copy has been requested, an option is available to allow continuous printing until an ending address has been displayed. Using this option, the entire memory contents could be printed without interruption.

```
10Ø ' "MEMPRINT" BY KEN VASKE
11Ø ' P.O. BOX 3Ø7, BANCROFT, IA 5Ø517
12Ø CLEAR 5ØØ
13Ø DIM H$(16)
14Ø FOR M = 1 TO 16
15Ø     READ H$(M)
16Ø     NEXT M
17Ø DATA Ø,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
18Ø LET SP$ = " "
19Ø LET HX$ = STRING$(36,2Ø)
2ØØ LET AS$ = STRING$(16,2Ø)
21Ø LET CO$ = "N"
22Ø CLS
23Ø PRINT TAB(2Ø) "MEMORY DISPLAY"
24Ø PRINT TAB(21) "BY KEN VASKE"
25Ø PRINT
26Ø PRINT"THIS PROGRAM WILL DISPLAY 256 BYTES OF MEMORY
    AT A TIME"
27Ø PRINT"AFTER EACH SCREEN FULL IS DISPLAYED, THE FOLLOWING"
28Ø PRINT"KEYS MAY BE PRESSED TO PERFORM THE INDICATED
    FUNCTION"
29Ø PRINT
3ØØ PRINT"     F - TO DISPLAY NEXT 256 BYTES"
31Ø PRINT"     B - TO DISPLAY PREVIOUS 256 BYTES"
32Ø PRINT"     E - TO END PROCESSING"
33Ø PRINT"         PRESSING ANY OTHER KEY WILL ALLOW NEW"
34Ø PRINT"         STARTING ADDRESS TO BE ENTERED"
35Ø PRINT
36Ø INPUT"DO YOU WANT HARD COPY PRINTED?  Y OR N"; PR$
37Ø IF PR$ <> "Y" AND PR$ <> "N" THEN 36Ø
38Ø IF PR$ = "Y" INPUT "DO YOU WISH CONTINUOUS PRINTOUT?
    Y OR N"; CO$
39Ø IF CO$ <> "Y" AND CO$ <> "N" THEN 38Ø
4ØØ INPUT"ENTER BEGINNING ADDRESS"; A1$
41Ø IF LEN(A1$) <> 5 THEN PRINT "WRONG LENGTH ADDRESS"
    :GOTO4ØØ
42Ø IF LEFT$(A1$,1) = "X" THEN GOSUB 86Ø:A1 = A5
    :ELSE A1 = VAL(A1$)
43Ø IF A5 = 99999 GOTO 45Ø
44Ø IF CO$ = "N" GOTO 5ØØ
45Ø INPUT "ENTER ENDING ADDRESS"; A1$
46Ø IF LEN(A1$) <> 5 THEN PRINT "WRONG LENGTH ADDRESS"
    :GOTO4ØØ
47Ø IF LEFT$(A1$,1) = "X" THEN GOSUB 86Ø:A4 = A5
    :ELSE A4 = VAL(A1$)
48Ø IF A5 = 99999 GOTO 45Ø
49Ø ' ROUTINE TO FORMAT AND PRINT DISPLAY
5ØØ LET A = A1
51Ø CLS
52Ø FOR L = 1 TO 16
53Ø  GOSUB 1Ø1Ø
54Ø  LET AS$ = ""
55Ø  LET HX$ = ""
56Ø  FOR G = 1 TO 4
57Ø   FOR C = 1 TO 4
58Ø    IF X < 32768 THEN S = PEEK(A) ELSE S = PEEK(A-65536)
59Ø    T% = S / 16 : B% = S - (T% * 16)
6ØØ    LET HX$ = HX$ + H$(T% + 1) + H$(B% + 1)
61Ø    IF (S < 32) OR (S > 127) LET S = ASC(".")
62Ø    LET AS$ = AS$ + CHR$(S)
63Ø    A = A + 1
64Ø   NEXT C
65Ø  NEXT G
66Ø  LET A3 = A - 16
67Ø  IF L > 1 THEN PRINT TAB(64)
68Ø  PRINT USING "#####"; A3;
7ØØ  PRINT TAB(6) HX$; TAB(42) AS$; TAB(59) AH$;
71Ø  IF PR$ = "N" GOTO 74Ø
72Ø  LPRINT USING "#####"; A3;
73Ø  LPRINT TAB(6) HX$; TAB(42) AS$; TAB(59) AH$
74Ø NEXT L
75Ø IF CO$ = "Y" GOTO 82Ø
76Ø K$ = INKEY$ : IF K$ = "" THEN 76Ø
77Ø IF K$ = "F" THEN A1 = A1 + 256: GOTO 5ØØ
78Ø IF K$ = "B" THEN A1 = A1 - 256: GOTO 5ØØ
79Ø IF K$ = "E" THEN GOTO 84Ø
8ØØ CLS
81Ø GOTO 4ØØ
82Ø LET A1 = A1 + 256
83Ø IF A1 < A4 GOTO 5ØØ
84Ø END
85Ø ' CONVERT HEX TO DECIMAL
86Ø LET HC = 4Ø96
87Ø LET A5 = Ø
88Ø FOR B = 2 TO 5
89Ø  LET C$ = MID$(A1$,B,1)
9ØØ  FOR CC = 1 TO 16
91Ø   IF C$ = H$(CC) GOTO 96Ø
92Ø  NEXT CC
93Ø  PRINT "INVALID HEX CHARACTER"
94Ø  LET A5 = 99999
95Ø  RETURN
96Ø  A5 = A5 + ((CC - 1) * HC)
97Ø  HC = HC / 16
98Ø NEXT B
```

continued on page 42

GRAPHICS WITH THE DOT MATRIX PRINTER

Dr. Weldon J. Horton

One of the major advantages that the old reliable dot matrix printer has over its "letter quality" bretheren, other than generally lower cost, is the ability to produce graphics. Although many printers have various fonts of graphics characters, the ultimate graphics capability is the ability to individually address each print wire in the matrix head. However, the production of graphics through "bit image" data input takes a sound understanding of at least three numbering systems, base two, base ten and base sixteen. The following two programs were written for a TRS-80 Model III and an Epson MX-70 printer, but may easily be modified to run on any similar configuration, utilizing eight bit parallel data handling. They should also serve to illustrate the number base conversions necessary to utilize bit image graphics. The mathematics used for number base conversions are extracted from standard mathematical handbooks, which also contain detailed explanations as to their derivation and functioning.

The first item necessary is to determine the basis of communication between computer and printer. For the TRS-80 output is by means of "LPRINT CHR$(n)", where "n" represents the base ten number for a standard ASCII character, control code or the like. In the bit image mode of the printer, however, these base ten numbers are interpreted as base two, or binary numbers. Since this printer is limited to eight addressable print wires, we are only interested in binary numbers 00000000 to 11111111, which are the equivalent of base ten 0 to 255.

With communications established, the printer spacing is adjusted. Since we are going to address eight print positions, representing a vertical column of eight printing wires in the matrix head, spaced 1/72 of an inch apart, the line spacing must be set at 8/72 of an inch in order to insure that there is no blank space between horizontal lines of print, as is done in line 2200 of program 1. The base ten number "8", near the end of this line, input as CHR$(8), indicates the line spacing in 1/72-inch increments.

Since the MX-80 printer permits the mixing of text and bit image mode printing on the same line, line 2220 of program 1 indicates that the next character received is to be treated as binary, bit image data, rather than as text data. The last two CHR$(n) inputs in this line are a four digit base sixteen number with the two low order digits input first and the two high order digits input last. Because the MX-70 stores eighty characters in its buffer before printing takes place, we need to input CHR$(15), as in line 2240 of program 1, in order to have the input from the keyboard of one column of bit graphics data printed immediately from the buffer after input, which in this case is the variable "A".

Now that we are in the bit image mode, all base ten CHR$(n) inputs are treated as binary numbers. Where there is a "1" in the binary string a dot is printed. Where there is a "0" in the binary string, a dot is not printed. For example, the binary number 00000000 (base ten 0) would provide an eight high vertical dot column of blanks, and the binary number 11111111 (base ten 255) would print a column of eight dots.

Although program 1 is adequate as an example of printer bit graphics, it is really not very functional. Therefore, program 2 was developed to provide such functional niceties as multiple dot column input before printing, a capacity to save data for future use and an illustration of the somewhat peculair requirements for indicating how many bytes of information, up to 480 dot columns per line, are to be read as bit graphics before reverting to the text mode.

The initial input for the number of dot columns is in base ten, and is accumulated in program 2 by the variable "F" in line 2300. When it is desired to print a line of bit graphics characters, the base ten number "F" is converted into a base sixteen number by program 2, lines 3060 to 3080. Lines 3100 through 3140 insure that the base sixteen number has exactly four characters by inserting leading zeros. Next, line 3160 splits the four character base sixteen number into two parts, and places the two high order digits first and the two low order digits last. Both sets of base sixteen digits are now *individually* converted *back* to base ten by subroutine 3400 in lines 3180 and 3200. Printing may now proceed as in program 1, except that the number of bit image columns to be printed are now indicated in line 3260 of program 2 by the base ten variables "F2" and "G2".

Other innovations and refinements are, of course, possible in the two programs utilized here for illustrative purposes. However, these two programs do serve to open up the world of dot matrix printing a bit wider, and could easily lead to custom designed fonts, one key graphics from the keyboard to the printer, and the like. The only limits are in the approximately 380,000 possible dot positions per standard printed page and your imagination.

PROGRAM 1

```
10Ø CLS:CLEAR2ØØØ:DIMD$(2)
12Ø PRINT@334,"Dot Graphics - Direct Printer Input"
14Ø PRINT@469,"by Dr. Weldon J. Horton":FORI=1TO5ØØ:NEXTI
16Ø CLS:PRINT:PRINT:PRINT"This program allows you to input
    directly from the keyboard to the printer. Only Ø's and
    1's may be input to the printer. The LEFTmost digit
    represents the TOP dot position."
18Ø PRINT:PRINT"You may not input more than 8 digits. Leading
    Ø's may be omitted, trailing Ø's may NOT.":PRINT:PRINT
    :PRINT"(Press ENTER to continue)";
2ØØ INPUTK$
22Ø CLS
24Ø PRINT:PRINT:PRINT"  @ = next line      E = exit":PRINT
    :PRINT:INPUT"For ONE column: enter a maximum of EIGHT Ø's
    and 1's.";A$
26Ø IFA$="@"THENLPRINT:GOTO24Ø
28Ø IFA$="E"ORA$="e"THENEND
2ØØØ D$(1)="Ø":D$(2)=1
2Ø4Ø B$=A$:E=LEN(A$):IFE>8THENGOTO2Ø8Ø
2Ø6Ø FORD=1TOE:C$=RIGHT$(A$,1):FORB=1TO2:IFC$=D$(B)THEN21ØØ
    ELSENEXTB
2Ø8Ø A=Ø:PRINT:PRINT"*** MORE THAN EIGHT AND/OR ILLEGAL
    DIGIT. PLEASE RE-ENTER. ***":GOTO24Ø
21ØØ C=B-1:IFC>2THEN2Ø8Ø
212Ø A=A+INT(C*2↑(D-1)+.5)
214Ø A$=LEFT$(A$,LEN(A$)-1):NEXTD
```

```
22ØØ LPRINTCHR$(27);CHR$(65);CHR$(8);
2220 LPRINTCHR$(27);CHR$(75);CHR$(Ø1);CHR$(ØØ);
2240 LPRINTCHR$(A);CHR$(15);
2260 A=Ø:GOTO22Ø
```

PROGRAM 2

```
1ØØ CLS:CLEAR2ØØØ:DIMD$(16):DIMH(481):FORI=1TO16:READD$(I)
    :NEXTI
12Ø RESTORE:DATA Ø,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
14Ø PRINT@276,"PRINTER DOT GRAPHICS":PRINT@413,"by"
    :PRINT@469,"Dr. Weldon J. Horton":FORI=1TO5ØØ:NEXTI
16Ø CLS:PRINT:PRINT:PRINT"This program allows you to input
    directly from the keyboard to the printer, one line at a
    time. Only Ø's and 1's may be input to the printer. The
    LEFTmost digit represents the TOP dot position."
18Ø PRINT:PRINT"You may not input more than 8 digits for each
    vertical column on each line. Leading Ø's may be omitted,
    trailing Ø's may NOT.":PRINT:PRINT:PRINT"(Press any key
    to continue.)":GOSUB2ØØØØ
2ØØ CLS:PRINT@219,"MENU":PRINT@387,"@ = Return to Menu
    F = Fill a line with data":PRINT@451,"P = Print line
    on printer     S = Save on tape":PRINT@515,"L = Load in
    from tape     X = Load and print multiple lines"
21Ø PRINT@579,"C = Correct one column   & = Clear memory
    and rerun":X=Ø:X1=Ø:R=Ø:Z=Ø
215 GOSUB2ØØØØ
22Ø IFK$="P"ORK$="p"THENGOTO3ØØØELSEIFK$="S"ORK$="s"
    THENGOTO4ØØØ
24Ø IFK$="L"ORK$="l"THENGOTO5ØØØELSEIFK$="X"ORK$="x"
    THENGOTO6ØØØ
26Ø IFK$="&"THENGOTO1ØØELSEIFK$="C"ORK$="c"THENGOTO7ØØØ
5ØØ CLS
54Ø PRINT:PRINT:PRINT"          @ = Return":PRINT:PRINT
    :INPUT"For ONE column: enter a maximum of EIGHT Ø's
    and 1's";A$
56Ø IFA$="@"THENGOTO2ØØ
2Ø4Ø E=LEN(A$):IFE>8THENGOTO2Ø8Ø
2Ø6Ø FORD=1TOE:C$=RIGHT$(A$,1):FORB=1TO2:IFC$=D$(B)THEN21ØØ
    ELSENEXTB
2Ø8Ø A=Ø:PRINT:PRINT"*** MORE THAN EIGHT AND/OR ILLEGAL
    DIGIT. PLEASE RE-ENTER. ***":GOTO54Ø
21ØØ C=B-1:IFC>2THENGOTO2Ø8Ø
212Ø A=A+INT(C*2y(D-1)+.5)
214Ø A$=LEFT$(A$,LEN(A$)-1):NEXTD
216Ø IFZ>ØTHENGOTO7Ø6Ø
23ØØ F=F+1:H(F)=A:A=Ø:GOTO5ØØ
3ØØØ CLS:PRINT@394,"Set up printer.  @ = Return  # = Start
    printing":GOSUB2ØØØØ
3Ø2Ø IFK$="@"THENGOTO2ØØELSEIFK$<>"#"THENGOTO3ØØØ
3Ø4Ø CLS:PRINT@4ØØ,"******** PRINTER RUNNING *********"
3Ø6Ø G=Ø:F1=F
3Ø8Ø G=F1-INT(F1/16)*16:F1=INT(F1/16):E$=D$(G+1)+E$
    :IFF1>ØTHENGOTO3Ø8Ø
31ØØ IFLEN(E$)=3THENE$="Ø"+E$
312Ø IFLEN(E$)=2THENE$="ØØ"+E$
314Ø IFLEN(E$)=1THENE$="ØØØ"+E$
316Ø F$=RIGHT$(E$,2):G$=LEFT$(E$,2)
318Ø E$=F$:GOSUB34ØØ
32ØØ F2=R:E$=G$:GOSUB34ØØ
322Ø G2=R
324Ø LPRINTCHR$(27);CHR$(65);CHR$(8);
326Ø LPRINTCHR$(27);CHR$(75);CHR$(F2);CHR$(G2);
328Ø FORI=1TOF:LPRINTCHR$(H(I));:NEXTI
33ØØ LPRINTCHR$(15);:LPRINT
332Ø IFX>ØTHENGOTO6Ø4ØELSEGOTO2ØØ
34ØØ R=Ø:I=LEN(E$):FORJ=1TOI:M$=RIGHT$(E$,1):FORK=1TO16
    :IFM$=D$(K)THEN342ØELSENEXTK
342Ø ********************************************* MISSING LINE
344Ø RETURN ... ? ***************************** OBSCURED LINE
4ØØØ CLS:PRINT@145,"*** SAVE DATA ON TAPE ***":PRINT@334,"Set
     up tape recorder in *RECORD* mode":PRINT@486,"S = Start
    saving data":PRINT@6ØØ,"@ = Return"
4Ø2Ø GOSUB2ØØØØ
4Ø4Ø IFK$="@"THENGOTO2ØØ
4Ø6Ø IFK$="S"ORK$="s"THENGOTO4Ø8ØELSEGOTO4ØØØ
4Ø8Ø CLS:PRINT@458,"One moment please. I am saving data
    on tape."
41ØØ PRINT#-1,F
412Ø FORI=1TOF
414Ø PRINT#-1,H(I)
416Ø NEXTI
418Ø GOTO2ØØ
5ØØØ CLS:PRINT@145,"*** LOAD DATA IN FROM TAPE ***"
    :PRINT@33Ø,"Set up tape recorder in *PLAY* mode"
    :PRINT@468,"L = Start loading data":PRINT@6ØØ,
    "@ = Return"
5Ø2Ø GOSUB2ØØØØ
5Ø4Ø IFK$="@"THENGOTO2ØØ
5Ø6Ø IFK$="L"ORK$="l"THENGOTO5Ø7ØELSEGOTO5ØØØ
5Ø7Ø CLS:PRINT@456,"One moment please. I am loading data in
    from tape."
5Ø8Ø INPUT#-1,F
51ØØ FORI=1TOF
512Ø INPUT#-1,H(I)
514Ø NEXTI
516Ø IFX>ØTHENGOTO3Ø6ØELSEGOTO2ØØ
6ØØØ CLS:PRINT@78,"** LOAD AND PRINT MULTIPLE LINES **"
    :PRINT@216,"(Ø = Return)":PRINT@468," ":INPUT"Enter
    number of lines to be loaded and printed from tape.
    Loading and printing will commence immediately.";X
6Ø2Ø IFX=ØTHENGOTO2ØØELSEGOTO5Ø8Ø
6Ø4Ø X1=X1+1:IFX1=XTHENGOTO2ØØELSEGOTO5Ø8Ø
7ØØØ CLS:PRINT@148,"@ = Return. No change.":PRINT@33Ø," ";:
    INPUT"Input number of column to be changed";Z1$:
    IFZ1$="@"THENGOTO2ØØELSEZ=VAL(Z1$)
7Ø2Ø PRINT@458," ";:INPUT"Input new value of column";Z$
7Ø4Ø A$=Z$:GOTO56Ø
7Ø6Ø H(Z)=A:GOTO7ØØØ
2ØØØØ K$=""
2ØØ2Ø K$=INKEY$:IFK$=""THEN2ØØ2ØELSERETURN
```

Dr. Weldon J. Horton, Director
Learning Resource Center
Midland College
3600 N. Garfield
Midland, TX 79701 ■

MEMORY DISPLAY

cotinued from page 40

```
99Ø RETURN
1ØØØ ' CONVERT DECIMAL TO HEX
1Ø1Ø LET A2 = A
1Ø2Ø LET HC = 4Ø96
1Ø3Ø FOR B = 1 TO 4
1Ø4Ø  A3% = A2 / HC
1Ø5Ø  LET A3$(B) = H$(A3% + 1)
1Ø6Ø  LET A2 = A2 - (A3% * HC)
1Ø7Ø  LET HC = HC / 16
1Ø8Ø NEXT B
1Ø9Ø AH$ = A3$(1) + A3$(2) + A3$(3) + A3$(4)
11ØØ RETURN
```

Ken Vaske
P.O. Box 307
Bancroft, Iowa 50517 ■

POCKET COMPUTER CORNER

Steven M. Zimmerman, Leo M. Conrad, and Stanley M. Zimmerman

Space Ship Game

This pocket computer game may be played any place and at almost any time. If you choose, you may keep the game simple by limiting yourself to two dimensional space, a flat surface, or, if you choose, you may create a bigger problem by playing the game in up to five dimensional space.

The game consists of finding a space ship within the space defined. In the case of two dimensional space, the ship is hidden in a 10 by 10 grid. The grid is numbered from 0 to 9 as shown below:

```
                First dimension

      Ø 1 2 3 4 5 6 7 8 9
   Ø
   1
   2
   3
   4
   5
   6
   7
   8
   9
```

The computer selects two random numbers from 0 to 9 to locate the ship. The player must then "shoot" the space ship by selecting the same two random numbers. The computer keeps track of the number of shots needed to destroy the space ship.

In the two dimensional case the ship occupies one space in a 10 by 10 grid. You have a one in 10 x 10, or 100 chance of hitting the ship. In the three dimensional case the ship is in a 10 x 10 x 10 grid. There are 1,000 spaces. The chance of hitting the ship is one in a thousand. In the four dimensional case the odds are one out of 10,000 and in the five dimensional case the odds are one out of 100,000.

PLAYING THE GAME

The program starts by flashing the words SPACE SHIP on the computer's display. The next thing you see is the question:

(C)ONT (S)HIP (G)AME?

This menu allows you to continue playing the game you were in the middle of when you shut off the computer, selecting a new ememy ship to search for, or starting the game over again. The act of starting over again consists of inputting a new random seed number to start the computer's random number generator. Your pocket computer has limited memory, so there are no built-in error traps to prevent you from making errors. You must answer with the single letter inside the parentheses as shown. If you make an error, the computer assumes you selected the (S)HIP option and starts a new game for this purpose.

If you selected the (G)AME option by typing G <ENTER> you will see:

SEED NUMBER?

A seed number is used to start your random number generator. If you select a small number you will create a small start up problem. The first few numbers you generate will be zeros. This can be advoided by selecting a number greater than seven digits long. Type 9999999 <ENTER> as your answer to the above question to duplicate our results.

The next question is:

NUMBER OF DIMENSIONS?

The maximum number of dimensions allowed is five. Each dimension multiplies the number of locations by ten. It is possbile to play the game on a line, i.e. with a dimension of one. The odds in the single dimension game is one to ten of hitting the enemy ship. If you played the game using the seed number 9999999 the enemy ship will be located in position 9. When we played this game it only took us nine shots to hit the enemy ship.

Let's use two dimensional space for the sample run. Type 2 <ENTER>, and you hear the beeper beep twice (for two dimensions), and see the following flashing on the display:

FIRE 1.
COORDINATE 1.
?

Your answer must be a number from zero to nine. If you answer outside this range, you will not hit the enemy space ship and are just wasting a shot. Type 5 <ENTER> and you will see:

CORDINATE 2.
?

To complete the act of firing a shot, type 5 <ENTER> to continue. The computer will flash on the screen:

MISSED

and then stop with the message:

1. SHOTS USED

You must press the <ENTER> key to continue with the game. The game recycles with two beeps and the following question:

FIRE 2.
COORDINATE 1.
?

This time type 9 ENTER and the answer shown below:

CORDINATE 2.
? 4 ENTER

The computer will flash the following on the display:

KILLED THE KILON

and then stop with the following message:

2. SHOTS USED

When you press <ENTER>, you will start the game over again from the beginning.

The game is simple to use and should be a lot of fun for children of all ages.

EXAMINING THE PROGRAM

In your pocket computer, it is very important to use variables with care throughout the program. The following is a list of how the variables have been used in this program:

Variable	Use
A$	To direct the flow from the first menu
A	As a counter in FOR/NEXT statement
B	The random seed number
C	The number of dimensions
D A(4)	The value in the first dimension which locates the space ship
E A(5)	The value in the second dimension which locates the space ship
F A(6)	The value in the third dimension which locates the space ship
G A(7)	The value in the fourth dimension which locates the space ship
H A(8)	The value in the fifth dimension which locates the space ship
I	Locates the dimension and then counts the number of shots fired
J A(9)	The value in the first dimension which locates your shot.
K A(10)	The value in the second dimension which locates your shot.
L A(11)	The value in the third dimension which locates your shot.
M A(12)	The value in the fourth dimension which locates your shot.
N A(13)	The value in the fifth dimension which locates your shot.
P	Subscript counter for shot.
R	Subscript counter for space ship.
Q	A counter to see when a hit occurs. If Q is equal to C the number of dimensions all points are in agreement.

The letter A has two uses. Its first use is in the main menu. Once this use is complete the letter is reused for a counter in the FOR/NEXT loops for locating the space ship, firing the shots, and seeing if a hit has occured. In this program only an exact hit has value.

The program may be divided into five parts. Part I consists of lines 1, 2, and 3. Line one prints the title of the program on the display and then uses an INPUT statement to route the program to either (C)ONTine, to start a new (S)HIP, or to start a new (G)ame. If the (C)ONTine option is selected, line one routes the program to line 20. This is beyond the point in the program where a shot has been fired. The program first checks to see if a hit occured before allowing the player to make another shot.

Line 2 inputs a new seed number if the player tells the computer to start a new game. The seed number is multiplied by 10,000 and then made into a postive number in line 2. This aids in getting the random number generator started just in case the player selected a small number.

Line 3 is used to select the number of dimensions for the game. If the number selected is greater than five, the computer tells the user that the game may not proceed and then recycles the question.

Part II of the program is contained in lines 4 through 8. The task of this code is to select the location of the space ship. The variable A(I) is used, where I takes on the values of 4, 5, 6, 7, and 8 depending on the number of dimensions specified. If you are working in two dimensions, A(4) or D is the location of the ship in the first dimension and A(5) or E is the location of the ship in the second dimension. (Remember that, in the pocket computer, A is A(1), B is A(2), ... Z is A(26)).

The number produced by the computer is usually an eight digit number. In line 7 the value of B is divided by 10,000,000 and changed to an integer to yield a number between 0 and 9 inclusive. This completes the task of locating the space ship.

The next part of the program consists of lines 10 through 14. This code is used to locate the value of the shot fired at the space ship. Two PAUSE statements in lines 12 and 13 are used to instruct the player on how to answer the INPUT request in line 13.

Lines 20 through 23 perform the task of seeing if a hit has occured. The number of locations which are in agreement are counted by Q. An interesting adjustment to the game might be to print the value of Q at the conclusion of this task to tell the player if any of the dimensions are in agreement. It is recommended you add a line such as the following:

24:PAUSE "# GOOD COORDINATES ";Q

if you are interested in playing the game in the higher dimensions.

The last part of the game is in lines 30 and 31. If all points are in agreement, then line 30 tells you the enemy space ship has been hit and the game is over. The computer is routed to line 1 for an additional game by this line.

If the shot is a miss, line 31 tells you so and then directs you back to line 11 to continue the game.

PROGRAM LISTING

```
1:PAUSE "SPACE SHIP":INPUT "(C)OUNT (S)HIP (G)AME?" ;A$:
  IF A$="C" THEN 20
2:IF A$="G" INPUT "SEED NUMBER?";B:B=10¢5*ABS (B)
3:INPUT "NUMBER OF DIMENSIONS?";C:IF C>5 BEEP 2:PAUSE "MAX 5
  DIMENSIONS":GOTO 3
4:FOR A=1 TO C
5:B=23B-INT (23B/(10¢8+1))*(10¢8+1)
6:I=A+3
7:A(I)=INT (B/(10¢7))
8:NEXT A
10:I=0
11:I=I+1:BEEP C:PAUSE "FIRE ";I
12:FOR A=1 TO C:P=A+9
13:PAUSE "COORDINATE ";A:INPUT A(P)
```

continued on page 59

BOWLING STATISTICS LEDGER

John F. Rogers

0. Introduction

BOWLING STATISTICS LEDGER provides accumulative scoring data for bowling leagues. The conscientious League Secretary spends much time checking, recording, and printing the League Standings and Team Data pages each week. This program takes the data from the "recap sheets"—the results of a league session—and, with a minimum of keyboard work, produces all the information usually provided, plus a little more.

Since the program is written entirely in TRS-80 Model I Disk Basic, it can be used on both Model I's and Model III's, and with any DOS. (It was composed with NEWDOS/80 Version 2. In TRSDOS, remove the "logical record length" indicator in line 9700—the '40' in 'OPEN"R",1,F1$,40'—and in line 9800—the '46' of 'OPEN"R",2,F2$,46'.) Two disk drives are preferable, but one will suffice if a minimal System diskette is used. (One drive users must delete the +":1" in lines 9700 and 9800.) The program occupies about 10K of memory, reserves 2K for string space, and uses either .5K or 1K for file buffers. The printer must be able to print 132 columns, or the routine in lines 9300-9330 must be changed.

The program is full of prompts and checks that make it easy to enter the bowlers' scores accurately and swiftly. Just follow the screen requests, which are fully explained in this manual. Sample printouts are provided in the Appendices.

1. Getting Started

Turn the computer on, insert the System diskette in Drive 0, the data diskette in Drive 1, and press the RESET button. (If necessary, answer the date and time questions.) Under NEWDOS/80 Version 2, type BASIC,2V and press <ENTER>; under TRSDOS, type BASIC, then answer FILES? with 2; under DOSPLUS, type BASIC -F:2 and press <ENTER>. When READY appears, type RUN"BOWLSTAT/BAS" and press <ENTER>.

In a few seconds, a logo will appear on the screen:

BOWLING STATISTICS LEDGER
By
John F. Rogers

and in another few seconds, the blinking legend PRESS ANY KEY TO GO ON will be displayed at the bottom of the screen.

* There will be many times when you have to type in data requested. Whenever you see "?-" after a prompting message, pressing the <ENTER> key is necessary to enter typed-in data.

CAUTION: A computer is quite literal—that is, it believes what you tell it, exactly. There is no room for error on your part when entering data. Be sure an entry is correct before pressing <ENTER>. Often you won't be able to take it back. One irreversible error means starting over from the very beginning! Fortunately, this program gives you opportunities to change your mind after certain data are entered.

A *backup* of each data diskette is strongly recommended. That is, when you've finished with a league's entries, make a backup of the files on a separate diskette, which you should keep apart from the original.

An even better practice is to use a total of three data diskettes and alternate them as the original data receiver. For example, number the diskettes 1, 2, and 3. Use #1 to record the scores as they're entered, then backup to #2. The next week, use #2 to record scores, then backup to #3. The third week, use #3 to record scores, then backup to #1. Repeating such a cycle minimizes wear and tear on the diskettes and protects your accumulated statistics. If you have a power failure or some glitch is introduced to the Drive 1 diskette, you never totally lose more than two weeks' entries, because the unused diskette still has the data from two weeks prior, while at least one of the in-use diskettes has the previous week's data.

2. Initialization

The first question will be HOW MANY TEAMS ARE IN THIS LEAGUE? Reply with the correct number. Then answer WHAT IS THE NAME OF THIS LEAGUE? appropriately. (The program uses the first three letters of the league name to identify the league records stored on diskette. Therefore, only letters and/or numerals—no other characters, no spaces—may be used among the first three characters of a league name, the first of which *must* be a letter. If you're storing more than one league's records on a single diskette, be sure that the first three letters in the names are different. For example, LATE SHIFTERS is legal, but 1ST NATIONAL BANK and BY THE BAYOU are illegal league names.)

Answer the question WHAT IS THE BLIND SCORE? if it is a fixed number. A blind score may not be a fixed number but vary from week to week. Set up a dummy bowler, say BLIND SCORES, with an unused bowler number. If a team has a blind the first session of a new season, enter the dummy bowler name and number plus the blind scores and handicap (if any) as for an actual bowler (see Section 9). If the blind occurs after the first week, use the "new member" routine (see Section 4) to enter the dummy bowler name and number plus the blind scores and handicap (if any). Thereafter, enter any blind scores under the dummy bowler's number during the entry routine for actual bowlers. (If more than one blind occurs, use their total for each game as if for only one.)

WHAT IS THE HANDICAP PERCENTAGE (DECIMAL)? must have the percentage as a decimal—e.g., enter 85% as .85, *not* 85%. Next answer WHAT IS THE PAR FIGURE FOR COMPUTING HANDICAP? This is the number from which the average is subtracted and the result multiplied by the handicap percentage to compute the bowler's handicap. The usual number is 200.

HOW MANY GAMES ARE BOWLED? means "How many games does any one player bowl?" HOW MANY POINTS ARE POSSIBLE? may be different from the number of games bowled—total pins may count as a point won or lost; or, games between individuals may count as points won or lost.

WHAT IS THE SESSION DATE? *must* be typed in the format MM/DD/YY—e.g., April 16, 1982 would be entered as 04/16/82.

To double-check, the display will repeat the data just entered and ask if it's all correct. If *any* information is not as you want it, enter N and the entire sequence will repeat.

The question WILL THE ENTRIES BE THE START OF A NEW LEAGUE SEASON (Y OR N)? should be answered Y if you will be entering scores from the first session of a new league season. If you reply Y, the program will branch to a special subroutine for the first league session. (More later in Section 9.) If there is already information stored on the data diskette in Drive 1 from previous sessions of the current season, answer N.

After an N reply, you'll see IF YOU'RE READY TO ENTER SCORES and the blinking prompt PRESS ANY KEY TO CONTINUE. Do so when you're ready with the information.

3. Entering Team Data

Now you'll read SCORES FOR EACH PLAYER MUST BE ENTERED BY PLAYER NUMBER AND TEAM NUMBER, followed by TYPE TEAM NUMBER (1 - #) AND PRESS <ENTER>?. It doesn't matter in which order you enter team records, but be sure not to enter one team twice. Also, be sure to enter every team's number, even if there is no data to input. (Forfeits and postponements will be discussed later.) Blind scores be will entered later. Double-checking, you will be asked YOU WILL BE ENTERING SCORES FOR TEAM NUMBER (#) (TEAM NAME). IS THAT CORRECT (Y OR N)? If not, answer N and you'll see again TYPE TEAM NUMBER (1 - #) AND PRESS <ENTER>? Try again.

The next prompt will be IF THERE ARE NO SCORES TO ENTER, PLEASE ENTER ZERO (0) IN RESPONSE TO THE REQUEST. For instance, a player may have been too late to bowl in game one but did bowl the last two games; enter 0 to the request for Game 1 score. Or, a bowler may have had to leave after the second game; enter 0 for Game 3 score.

Next comes WILL THERE BE ANY NEW MEMBERS ADDED TO THIS TEAM'S ROSTER (Y OR N)? If someone has bowled for the first time this season but after the first session, answer Y here. You will be asked to PLEASE ENTER NEW ROSTER MEMBERS AFTER ENTERING END WHEN FINISHED WITH OLD MEMBERS. New bowlers' records must be handled differently from existing ones. That is, first enter records for those who have already bowled this season, then enter END, *then* enter new bowlers' scores when asked for them. (More in a moment.) If you answer N, there will be no further discourse about new bowlers for this team. (If a team had a postponement the *first* session, *all* will be "new members" if the make-up is entered before the next session.)

WHEN YOU FINISH WITH A TEAM'S ENTRIES, TYPE END HERE AND PRESS <ENTER>; IF NOT FINISHED, JUST PRESS <ENTER> (FOR FORFEITS, TYPE FFT AND PRESS <ENTER>; FOR POSTPONEMENTS, TYPE PSTP AND PRESS <ENTER>)? gets you started on a team's bowlers' scores. If you have more scores to enter for the current team, just press <ENTER> in response to this prompt. But when you've completed entering a team's scores, type END before pressing <ENTER>.

If a team forfeited its match, enter FFT (see Section 4.A). If a team postponed and hasn't made up the match yet, enter PSTP (see Section 4.B.).

4. Entering Bowler Data

Now you will enter the individual scores. (Blind scores will be handled separately.) First you will TYPE BOWLER'S NUMBER AND PRESS <ENTER>. It is necessary to keep a record of players by name *and* number.

To insure that you and the computer are in agreement, the screen will display TO CHECK - YOU WILL BE ENTERING SCORES FOR (NAME) OF TEAM NUMBER (#) - (TEAM) IS THAT CORRECT (Y OR N)? If everything is O. K., enter Y; if not, enter N and you will be asked to repeat this entry.

Assuming that you agreed with the computer, now appears on a cleared screen ENTERING SCORES FOR (NAME) OF (TEAM) and superimposed over one another:

TYPE GAME 1 SCORE AND PRESS <ENTER>?
TYPE GAME 2 SCORE AND PRESS <ENTER>?
TYPE GAME 3 SCORE AND PRESS <ENTER>? (etc.)

Then:

TYPE BOWLER'S ONE-GAME HANDICAP AND PRESS <ENTER>?

To double-check, the display will tell you THE (#) GAMES AND SCRATCH SERIES ARE..., below which the numbers you just entered will appear along with their total, plus ONE-GAME HANDICAP = (#) and the question ARE THEY CORRECT (Y OR N)? If you've made a mistake, enter N and the program will take you back to TYPE GAME 1 SCORE AND PRESS <ENTER>?; repeat the scores and handicap.

When you've finished with one bowler's scores, the computer will display the bowler's series, high game to date, and high series to date, then, when you're ready, take you back to the beginning of Step 4 for the next bowler. If you have entered scores for all the old bowlers, enter END in the place where it is indicated.

4.A. Forfeits

If a team forfeited, enter FFT in the place indicated. If at least one member bowled but the team still forfeited, hold the scores for the next session when the team does not forfeit, and enter them then, even if you must enter some bowlers twice. (A printout is not possible in this case; just repeat last session's.)

4.B. Postponements

If any teams postponed their match(es), enter PSTP in the place indicated. The computer will record nothing, but the teams *will* appear in the League Standings printout. (A printout is not possible in this case; just repeat last session's.)

If you want a new League Standings printout after entering *make-ups* of postponed matches, still enter *all* team numbers, not just the postponed teams. (Follow the PSPT procedure. That is, enter the team numbers and scores for the made-up match(es), and also enter all the other team numbers, entering PSPT in the place indicated.) Otherwise, enter only the make-up teams.

4.C. New Members

Now, if you replied Y earlier to the question about

NEW members of a team, you will be asked DO YOU NEED TO DELETE ANYBODY BEFORE ENTERING A NEW BOWLER? A team roster cannot contain more than 15 members. If you're adding a new bowler to an already full roster (15 members), you must delete the records of an inactive player. Here's where it is done—answer Y. If there are fewer than 15 members of the roster, answer N.

The deletion routine begins YOU MUST DELETE A BOWLER FROM THE ROSTER, and continues TYPE THE NUMBER OF THE BOWLER TO BE DROPPED? Refer to your records and enter the number of an inactive bowler. To substantiate, YOU WANT TO DROP BOWLER NUMBER (#) (NAME) (Y OR N)? appears. If this is the correct "droppee," enter Y, and the record will be cleared.

Whether you delete anyone or not, the screen clears and TYPE NAME OF NEW BOWLER FOR TEAM NUMBER (#) (TEAM) (MAXIMUM 24 CHARACTERS, INCLUDING SPACES) AND PRESS <ENTER>? (Any letters over 24 will be ignored, so abbreviate to fit if necessary.) Next comes TYPE NEW BOWLER NUMBER AND PRESS <ENTER>? (The number must be an unused one.) Now carefully enter the scores—the prompts are the same as before. This routine concludes with ANY MORE (Y OR N)? If you're finished with new members, enter N; otherwise, enter Y and the routine repeats.

5. Entering Blind Scores

WERE THERE ANY BLIND SCORES (Y OR N)? will be the first question after you finish entering actual bowlers' scores. If there are any and the blind score is a fixed number, answer Y and you will see, superimposed over one another,

HOW MANY BLINDS IN GAME 1?
HOW MANY BLINDS IN GAME 2?
HOW MANY BLINDS IN GAME 3? (etc.)

Answer appropriately for each separate game.

DO BLIND SCORES GET HANDICAP? Answer Y if they do, N otherwise. If you answer Y, TYPE THE TOTAL AMOUNT OF BLINDS' HANDICAP IN GAME (#) AND PRESS <ENTER>? will be displayed for each game. Add up all the blinds' handicap for game (#) and enter it, doing so for each separate game.

If the blind score is not a constant but changes from week to week, use the "dummy bowler" concept explained in Section 2, and answer N to the first question.

6. Summing Up

Finally, TYPE NUMBER OF POINTS WON BY TEAM AND PRESS <ENTER> (ENTER 1/2 POINT AS .5)? The program will compute the number of points lost, based upon your earlier reply to the number of possible points. Accumulated totals will be printed in both printouts (next two sections). The screen will display HI TEAM GAME TO DATE = (#) and HI TEAM SERIES TO DATE = (#)

7. Team Printout

Lastly appears DO YOU WANT A PRINTOUT NOW FOR TEAM NUMBER (#) (Y OR N)? If you do want a printout for this team, reply Y, and you'll get something like what you see below. (See further samples in Appendix A.) (Answer N following an FFT or PSTP entry.) In either case, the program returns to Step 3.

8. League Standings Printout

After you've gone through Steps 3 - 7 for each and every team in the league, the computer automatically goes to the routine beginning I'M READY TO PRINT OUT LEAGUE STANDINGS... (Be sure that the printer paper is in such a position that the list will not print across the perforations if you're using fanfold paper.) When *you're* ready, press any key and the computer will sort the new, updated records for a League Standings printout. The sort will take only a few seconds—the more teams in the league the longer the the sorting time. Be patient, please. (The sorting is on a percentage basis, not by games won, so teams with postponements will be ranked properly.) See more sample printouts in Appendix B.

9. First Session of a New Season

The main difference between entering the first session of a new season and entering later sessions is that you have to enter all the team and bowler names as well as the scores.

If you answer Y to WILL THE ENTRIES BE THE START OF A NEW LEAGUE SEASON (Section 2), the next prompt will be FIRST SESSION OF A NEW SEASON; YOU'LL HAVE TO ENTER ALL THE INFORMATION ASKED FOR. It is important to keep a record of each team by name *and* number. Also, you must keep each team's list of players by number as well as name. A team roster is limited to fifteen (15) members—use the numbers 1 to 15 with the names. (Use the numbers 1 to (#) of teams with the team names.)

Because the first three letters of a team's name are used to identify that team's records stored on diskette, be sure that no two team names have the same first three letters. Only letters and/or numerals are allowed—no other characters, no spaces—and the first character must be a letter. For instance, FIRST NATIONAL BANK and ONE MORE TIME are legal, but 1ST NATIONAL BANK, MR. C'S FACTORY, and A FAST FIVE are illegal. Also, enter the names without commas, please. Any characters over 24 will be ignored, so abbreviate to fit if necessary.

Now you're ready for TYPE IN THE TEAM NAME (MAXIMUM 24 CHARACTERS, INCLUDING SPACES). WHEN DONE WITH ALL TEAM RECORDS, ENTER END? Please observe all the rules for names mentioned above. Then you'll see TYPE IN THE TEAM NUMBER (1 - #) AND PRESS <ENTER>? Keep a record of the number associated with each team.

KEEPING NUMBER AND NAME TOGETHER IN YOUR RECORDS, TYPE IN NAME OF BOWLER (MAXIMUM 24 CHARACTERS). WHEN DONE WITH THIS TEAM, ENTER END (FOR FORFEITS, ENTER FFT) (FOR POSTPONEMENTS, ENTER PSTP)? (Refer to Section 4 for action on forfeits and postponements.) If you enter FFT, you'll only have to enter zero (0) for ...GAMES WON? to get this team in the League Standings printout. If you enter PSTP, no further action is necessary.

If you enter a bowler name, then TYPE BOWLER'S NUMBER AND PRESS <ENTER>? will appear. Next

you'll read BOWLER NUMBER (#) NAMED (NAME) OF (TEAM), followed by, superimposed over one another,

TYPE GAME 1 SCORE AND PRESS <ENTER>?
TYPE GAME 2 SCORE AND PRESS <ENTER>?
TYPE GAME 3 SCORE AND PRESS <ENTER>? (etc.)
TYPE BOWLER'S ONE-GAME HANDICAP AND PRESS <ENTER>?

To double-check, the display will tell you THE (#) GAMES AND SCRATCH SERIES ARE..., below which the numbers you just entered will appear along with their total, plus ONE-GAME HANDICAP = and the question ARE THEY CORRECT (Y OR NO)? If you've made a mistake, enter N and the program will take you back to TYPE GAME 1 SCORE AND PRESS <ENTER>; repeat the scores and handicap more carefully.

Then you'll go back for the next bowler, starting with KEEPING NUMBER AND NAME TOGETHER.... Entering END here will take you to WERE THERE ANY BLIND SCORES (Y OR NO)? If you reply Y, then the sequences described in Sections 5 and 6 will operate.

After finishing with a team, you'll be asked if you want a printout of this team (Section 7). You should answer Y here unless the team has forfeited its match.

The next move is back to the request for a team name: TYPE IN THE TEAM NAME.... When you've finished with all the teams, enter END here. That will lead to I'M READY TO PRINT OUT LEAGUE STANDINGS.... See Section 8.

SAMPLES OF SCREEN DISPLAYS

```
Number of Teams = 4

League Name = TEST ONE

Blind Score = 165

Handicap Percentage =  85%

Number of Games Bowled = 3

Number of Points Possible = 4

Session Date = 04/30/82

Are these facts all correct (Y or N)? x
```

From Section 2, Initialization:

Checking that entered data is all correct.

```
When you finish with a team's entries, type END here
 and press <ENTER>; if NOT finished, just press <ENTER>
(For forfeits, type FFT and press <ENTER>)
(For postponements, type PSTP and press <ENTER>?
```

From Section 3, Entering Team Data:

For ending a team's entries, or indicating a Forfeit or a Postponement.

```
When you finish with a team's entries, type END here
 and press <ENTER>; if NOT finished, just press <ENTER>
(For forfeits, type FFT and press <ENTER>)
(For postponements, type PSTP and press <ENTER>?

Type Bowler's Number and press <ENTER>? 1
```

Same as above, with request for Bowler's Number.

```
When you finish with a team's entries, type END here
 and press <ENTER>; if NOT finished, just press <ENTER>
(For forfeits, type FFT and press <ENTER>)
(For postponements, type PSTP and press <ENTER>?

Type Bowler's Number and press <ENTER>? 1

        To check - you will be entering scores for
         ALLEN MAITRE            of team number 1
                    - OILFIELD SALES & SERVICE
Is that correct (Y or N)? x
```

Same as above, with check on bowler whose scores will be entered.

```
Type in the Team Name (maximum 24 characters,
including spaces).  When done with all teams' records,
enter END? GROW JOHN'S
 Type in the Team Number ( 1 - 4 ) and press <ENTER>? 2
```

From Section 9, First Session of a New Season:
Entering Team Name and Number.

```
      Keeping Number and Name together in records,
type in Name of Bowler (maximum of 24 characters).  When done
with this team, enter END
(For forfeits, enter FFT)
(For postponements, enter PSTP)? JOHN ROGERS
 Type bowler's number and press <ENTER>? 5
```

From Section 9, First Session of a New Season:
Entering Bowler Name and Number

```
      Bowler number 5 named JOHN ROGERS
of GROW JOHN'S

Type Game 3 score and press <ENTER>?

Type bowler's one-game handicap and press <ENTER>? 1

Bowler's series = 817  High Game to date = 300
        High Series to date = 817
The 3 games and scratch series are...
   249   268   300  817
One-game handicap = 1   Are they correct (Y or N)? Y

.................. PRESS ANY KEY TO CONTINUE ..................

      Will there be any NEW members added to
 this team's roster (Y or N)? Y
 Please enter new roster members AFTER entering END when
 finished with OLD members

           .. PRESS ANY KEY TO CONTINUE ..
```

From Sections 4 & 9:
Entering bowlers scores.

From Section 3:
Checking for first-time bowlers.

```
When you finish with a team's entries, type END here
 and press <ENTER>; if NOT finished, just press <ENTER>
(For forfeits, type FFT and press <ENTER>)
(For postponements, type PSTP and press <ENTER>?

Type Bowler's Number and press <ENTER>? 7



        To check - you will be entering scores for
         D.D. DELAUNE            of team number 2
                    - GROW JOHN'S
Is that correct (Y or N)? Y
```

```
     Type name of new bowler for team number 2
GROW JOHN'S
    (maximum 24 characters, including spaces)
and press <ENTER>? LORD FAUNTLEROY
      Type new Bowler Number and press <ENTER>? 9

Type Game 3 score and press <ENTER>?
 Type bowler's one-game handicap and press <ENTER>? 1
 Bowler's series = 800  High Game to date = 300
       High Series to date = 800    ANY MORE (Y or N)? N
The 3 games and scratch series are...
   240   260   300  800
One-game handicap = 1   Are they correct (Y or N)? Y
```

From Section 4:
Entering a new bowler Name, Number, & Scores.

APPENDIX A

Team	Won	Lost	Total Pins	Average	Week of 05/12/82
Team # 1 GROW JOHN'S	7	1	5925	987	

Bowler	Handicap	Total Pins	Games	Average	Bowler	Handicap	Total Pins	Games	Average
1 TOMMY STEWART	26	1017	6	169	2 PRESTON COPE	45	440	3	146
3 GLYNN HOLLOWAY	36	471	3	157	4 MIKE BEADLE	29	991	6	165
5 JOHN ROGERS	4	585	3	195	6 D. D. DELAUNE	26	507	3	169
7 SING BEADLE	29	495	3	165	8 BARRY MARX	39	459	3	153

Team	Won	Lost	Total Pins	Average	Week of 05/12/82
Team # 3 AMERICAN SUPPLY	1	7	2874	958	

Bowler	Handicap	Total Pins	Games	Average	Bowler	Handicap	Total Pins	Games	Average
1 SAMMY BAGWELL	44	444	3	148	2 WAYNE BERGERON	48	431	3	143
3 BARRY BROUSSARD	56	404	3	134	4 JERRY BROUSSARD	56	400	3	133
5 BILL BROWN	52	415	3	138					

Team	Won	Lost	Total Pins	Average	Week of 05/12/82
Team # 2 OILFIELD SALES & SERVICE	2	6	5829	971	

Bowler	Handicap	Total Pins	Games	Average	Bowler	Handicap	Total Pins	Games	Average
1 ALLEN MAITRE	15	1093	6	182	2 ROLAND CRAPPELL	17	1080	6	180
3 SHORTY SNIDER	26	1015	6	169	4 MYRON DUPRE	39	462	3	154
5 TOMMY LEBOEUF	25	1024	6	170	6 BILL ALFRED	27	504	3	168

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 R... 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.

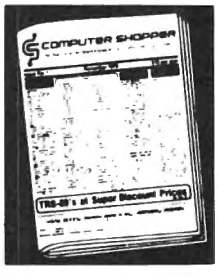
FOR SALE: Interdata (Perkin-Elmer) 7/16 Mini with 32KB core, front panel, 50A PWR supply. Includes HS tape reader, interfaces for LP, 2 (TTY), and RS-232 (Full duplex, programmable). Includes manuals and much SW (Basic, Fortran, OS, etc.,). $800 - After 6 P... 2035...

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... 33...

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.

BOWLING STATISTICS LEDGER

continued from page 50

Team	Won	Lost	Total Pins	Average	Week of 05/12/82
Team # 4 ALLEN'S TV CABLE	6	2	5781	963	

Bowler	Handicap	Total Pins	Games	Average	Bowler	Handicap	Total Pins	Games	Average
1 DALE BEADLE	36	945	6	157	2 CHRIS PRICE	37	939	6	156
3 DAVID PRICE	50	849	6	141	4 MARK PRICE	49	855	6	142
5 KEN RHYNE	39	927	6	154					

Team	Won	Lost	Total Pins	Average	Week of 05/19/82
Team # 1 GROW JOHN'S	11	1	8965	996	

Bowler	Handicap	Total Pins	Games	Average	Bowler	Handicap	Total Pins	Games	Average
1 TOMMY STEWART	22	1557	9	173	2 PRESTON COFE	45	878	6	146
3 GLYNN HOLLOWAY	36	471	3	157	4 MIKE BEADLE	30	1480	9	164
5 JOHN ROGERS	0	1198	6	199	6 D. D. DELAUNE	26	1017	6	169
7 SING BEADLE	29	495	3	165	8 BARRY MARX	39	459	3	153

Team	Won	Lost	Total Pins	Average	Week of 05/19/82
Team # 4 ALLEN'S TV CABLE	6	6	8631	959	

Bowler	Handicap	Total Pins	Games	Average	Bowler	Handicap	Total Pins	Games	Average
1 DALE BEADLE	33	1455	9	161	2 CHRIS PRICE	37	939	6	156
3 DAVID PRICE	50	849	6	141	4 MARK PRICE	49	855	6	142
5 KEN RHYNE	39	1389	9	154	6 MONTY NIETTE	67	363	3	121
7 TERRY NIETTE	68	357	3	119					

APPENDIX B

LEAGUE STANDINGS AFTER WEEK OF 05/12/82

	Team	Won	Lost	Total Pins	Average
1	GROW JOHN'S	7	1	5925	987
2	ALLEN'S TV CABLE	6	2	5781	963
3	OILFIELD SALES & SERVICE	2	6	5829	971
4	AMERICAN SUPPLY	1	7	2874	958

LEAGUE STANDINGS AFTER WEEK OF 05/19/82

	Team	Won	Lost	Total Pins	Average
1	GROW JOHN'S	11	1	8965	996
2	ALLEN'S TV CABLE	6	6	8631	959
3	OILFIELD SALES & SERVICE	2	6	5829	971
4	AMERICAN SUPPLY	1	7	2874	958

PROGRAM LISTING

```
1Ø ' League Bowling Statistics Program *
12 ' Copyright (C) 1982 by John F. Rogers *
2Ø CLEAR2ØØØ : EN=Ø : ONERRORGOTO11ØØØ
1ØØ CLS : PRINTCHR$(23) : PRINT@326,"BOWLING STATISTICS LEDGER";
 : PRINT@478,"BY"; : PRINT@594,"JOHN F. ROGERS"; : FORI=ØTOØØØ :
 NEXTI : GOSUB51ØØ
12Ø GOSUB56ØØ : AN$="" : DIMA9$(T8+1),PC(T8+1),GW(T8+1),
GL(T8+1),TT(T 8+1),AT(T8+1)
13Ø CLS : PRINT@42Ø," " : PRINT"WILL THE ENTRIES BE THE START OF
 A NEW LEAGUE SEASON (Y OR N)"; : INPUT AN$ : IFLEFT$(AN$,1)=
"Y"THENAN$="" : GOTO65ØØ
14Ø CLS : GOSUB52ØØ : PRINT@528,"If you're ready to enter
 scores,"; : GOSUB5ØØØ
15Ø CLS : GOSUB52ØØ : PRINT@72,"Scores for each player must be
 entered by"; : PRINT@13Ø,"player number and team number.";
155 PRINT@258,"Type team number ( 1 -";T8; : INPUT") and press
 <ENTER>";T9 : RT=T9 : IFT9<1ORT9>T8THEN155
16Ø GOSUB55ØØ : GOSUB8ØØØ : PRINT@39Ø,"You will be entering
 scores for Team Number";Z9; : PRINT@45Ø,TN$;". Is that
 correct (Y or N)"; : INPUTAN$ : IFLEFT$(AN$,1)="Y" THEN AN$=""
ELSE CLS : GOTO155
165 CLS : PRINT@328,"(If there are no scores to enter,"; : PRINT
@386,"please enter zero <Ø> in response to the request."; :
 GOSUB97ØØ : GOSUB5ØØØ
17Ø CLS : PRINT@39Ø,"Will there be any NEW members added to" :
 PRINT"this team's roster (Y or N)"; : INPUTAN$ : IFLEFT$(AN$,1)
="Y"THENPRINT"Please enter new roster members AFTER entering END
 when" : PRINT"finished with OLD members" : NB=99 : GOSUB5ØØØ
18Ø CLS : GOSUB52ØØ : EN$="" : PRINT@66,"When you finish with a
 team's entries, type END here"; : PRINT@13Ø," and press
 <ENTER>; if NOT finished, just press <ENTER>"; : PRINT@194,
"(For forfeits, type FFT and press <ENTER>)";
185 PRINT@258,"(For postponements, type PSTP and press <ENTER>";
 : INPUTEN$ : IFEN$="END"THEN23ØELSEIFEN$="PSTP"THEN57ØØELSE
IFEN$="FFT"THEN GT=GT-GB : GOTO25Ø
19Ø PRINT@386,"Type Bowler's Number and press <ENTER>"; : INPUT
B9 : GOSUB7ØØØ
2ØØ PRINT@715,"To check - you will be entering scores for "; :
 PRINT@78Ø,BN$;" of team number";Z9; : PRINT@854,"- ";TN$; :
 PRINT@898,"Is that correct (Y or N)"; : INPUTAN$ : IF
LEFT$(AN$,1)="N"THEN15Ø
21Ø GOSUB6ØØØ : PRINT@748,"W O R K I N G ..."; : GOSUB9ØØØ :
 AN$=""
22Ø GOSUB71ØØ : GOSUB555Ø : GOTO18Ø
23Ø IFNB=99THENGOSUB66ØØ
24Ø CLS : GOSUB52ØØ : PRINT@72,"Were there any blind scores (Y
 or N)"; : INPUTAN$ : IFLEFT$(AN$,1)="Y"THENAN$="" : GOSUB61ØØ
25Ø CLS : GOSUB52ØØ : PRINT@72,"Type number of points won by
 team and press <ENTER>"; : PRINT@13Ø,"(Enter 1/2 point as .5)";
 : INPUTWN : GOSUB92ØØ
26Ø GOSUB81ØØ : GOSUB5ØØØ
27Ø CLS : GOSUB52ØØ : PRINT@72,"Do you want a printout now for
 team number";T9; : PRINT@13Ø,"(Y or N)"; : INPUTAN$ : IF
```

```
LEFT$(AN$,1)="Y"THENAN$="" : GOSUB93ØØ
28Ø EN=EN+1 : IFEN=T8THENCLOSE1 : GOTO94ØØ
29Ø CLOSE 1 : GOSUB55ØØ : GOTO15Ø
5ØØØ Z@$=INKEY$ : PRINT@978," PRESS ANY KEY TO CONTINUE "; : FOR
I=ØTO45 : NEXTI : IFZØ$<>""THENPRINT@976,STRING$(31,32); :
 RETURN
5Ø1Ø PRINT@976,STRING$(31,143); : FORI=ØTO3 : NEXTI : GOTO5ØØØ
51ØØ Z@$=INKEY$ : PRINT@966," PRESS ANY KEY TO GO ON "; : FORI=Ø
TO45 : NEXTI : IFZØ$<>""THENPRINT@962,STRING$(28,32); : RETURN
511Ø PRINT@962,STRING$(28,143); : FORI=ØTO3 : NEXTI : GOTO51ØØ
52ØØ CLS : POKE1536Ø,191 : PRINT@1,STRING$(62,131); : POKE15423,
191 : POKE1632Ø,191 : PRINT@961,STRING$(62,176); : FORI=15424TO
16256STEP64 : POKEI,191 : POKEI+63,191 : NEXTI : POKE16383,191 :
 RETURN
53ØØ IFPEEK(14312)=63THENPRINT@852,STRING$(21,32); : RETURN
531Ø PRINT@852,"TURN ON LINE PRINTER!"; : GOTO53ØØ
54ØØ LPRINT" " : LPRINTTAB(17)"Team";TAB(4Ø)"Won";TAB(53)"Lost";
TAB(63)"Total Pins";STRING$(6,32);"Average";"     Week of ";DT$
541Ø RETURN
55ØØ SE=Ø : S1=Ø : K=Ø : H1=Ø : H2=Ø : H3=Ø : H4=Ø : TP=Ø :
 TG=Ø : GT=Ø : TT=Ø : T4=Ø : GW=Ø : WN=Ø : GL=Ø : FORI=1TOGB :
 G(I)=Ø : T(I)=Ø : B(I)=Ø : NEXTI
551Ø RETURN
555Ø SE=Ø : S1=Ø : H1=Ø : H2=Ø : TP=Ø : TG=Ø : FORI=1TOGB :
 G(I)=Ø : NEXTI
556Ø RETURN
56ØØ CLS : GOSUB52ØØ : PRINT@13Ø,"How many teams are in this
 league"; : INPUTT8
561Ø PRINT@258,"What is the name of the league"; : INPUTLN$ :
 LF$=LEFT$(LN$,3)
562Ø PRINT@386,"What is the blind score"; : INPUTBL
563Ø PRINT@514,"What is the handicap percentage (decimal)"; :
 INPUTHP : IFHP>1THENPRINT"Decimal please!"; : GOTO563Ø
5635 PRINT@581,"What is the par figure for computing handicap";
: INPUTPF
564Ø PRINT@642,"How many games are bowled"; : INPUTGB : PRINT@
77Ø,"How many points are possible"; : INPUTPP
565Ø PRINT@898,"What is the session date (MM/DD/YY)"; : INPUTDT$
 : IFMID$(DT$,3,1)<>"/"ORMID$(DT$,6,1)<>"/"THENPRINT@85Ø,
"INVALID ENTRY..."; : FORX=ØTO1ØØ : NEXTX : PRINT@898,STRING$
(6Ø,32); : PRINT@85Ø,STRING$(16,32); : GOTO565Ø
566Ø CLS : PRINT"Number of Teams =";T8 : PRINT : PRINT"League
 Name = ";LN$ : PRINT : PRINT"Blind Score =";BL : HP$=STR$
(HP*1ØØ)+"%" : PRINT : PRINT"Handicap Percentage = ";HP$
5665 PRINT"   Par Figure for Handicap Computation =";PF : PRINT
"Number of Games Bowled =";GB
567Ø PRINT : PRINT"Number of Points Possible =";PP : PRINT :
 PRINT"Session Date = ";DT$ : PRINT : PRINT"Are these facts all
 correct (Y or N)"; : INPUTAN$ : IFLEFT$(AN$,1)="Y"THENAN$="" :
 GOSUB98ØØ : RETURNELSECLOSE2 : GOTO56ØØ
568Ø RETURN
57ØØ GOSUB8ØØØ : IFGW+GL=ØTHENPC(RT)=ØELSEPC(RT)=GW/(GW+GL)
57Ø5 A9$(RT)=TN$ : GW(RT)=GW : GL(RT)=GL : TT(RT)=TT : AT(RT)=AT
 : GOSUB81ØØ
571Ø GOTO27Ø
58ØØ A9$(RT)=TN$ : PC(RT)=Ø : GW(RT)=Ø : GL(RT)=Ø : TT(RT)=Ø :
 AT(RT)=Ø : H3=Ø : H4=Ø : TT=Ø : AT=Ø : GW=Ø : GL=Ø : GT=Ø
581Ø GOSUB81ØØ : GOTO6585
6ØØØ CLS : GOSUB52ØØ : PRINT@7Ø,"Entering scores for ";BN$; :
 PRINT@148,"of ";TN$;
6Ø1Ø FORI=1TOGB : PRINT@322,"Type Game";I;"score and press
 <ENTER>"; : INPUTG(I) : PRINT@359,"   "; : NEXTI
6Ø2Ø PRINT@45Ø,"Type bowler's one-game handicap and press
 <ENTER>"; : INPUTGH : SE=Ø : FORI=1TOGB : SE=SE+G(I) : NEXTI
6Ø3Ø PRINT@7Ø6,"The";GB;"games and scratch series are..."; :
 FORI=1TOGB : PRINT@(766+6*I),G(I); : NEXTI : PRINTSE; : PRINT@
834,"One-game handicap =";GH;"  Are they correct (Y or N)"; :
 INPUTAN$ : IFLEFT$(AN$,1)="Y"THENAN$="" : RETURNELSEGOSUB555Ø :
 GOTO6ØØØ
61ØØ FORI=1TOGB : PRINT@194,"How many blinds in Game";I; :
 INPUTB(I) : T(I)=T(I)+B(I)*BL : NEXTI
611Ø PRINT@322,"Do blind scores get handicap"; : INPUTAN$ : IF
LEFT$(AN$,1)<>"Y"THENRETURN
612Ø FORI=1TOGB : PRINT@45Ø,"Type the total amount of blinds'
 handicap in Game";I; : PRINT@516," "; : INPUT"and press
 <ENTER>";BH(I) : T(I)=T(I)+BH(I) : PRINT@535,"   "; : NEXTI
613Ø RETURN
6499 ' First session of new season entry subroutine *
65ØØ CLS : PRINTCHR$(23) : PRINT@66,"FIRST SESSION OF A NEW
SEASON"; : PRINT@194,"You'll have to enter all the"; : PRINT
@326,"information asked for."; : GOSUB51ØØ
651Ø CLS : GOSUB52ØØ : PRINT@13Ø,"Type in the Team Name (maximum
 24 characters,"; : PRINT@194,"including spaces).  When done
 with all teams' records, "; : PRINT@258,"enter END"; : INPUTT
N$ : IFTN$="END"THEN94ØØ
6515 GOSUB97ØØ : PRINT@322,"Type in the Team Number ( 1 -";T8;")
 and press <ENTER>"; : INPUTT9 : RT=T9
652Ø CLS : GOSUB52ØØ : PRINT@72,"Keeping Number and Name
together in records,"; : PRINT@13Ø,"type in Name of Bowler
(maximum of 24 characters).  When done";
6524 PRINT@194,"with this team, enter END"; : PRINT@258,"(For
forfeits, enter FFT)"; : PRINT@322,"(For postponements, enter
PSTP)"; : INPUTBN$ : IFBN$="END"THEN657ØELSEIFBN$="FFT"THENGT=
GT-GB : GOTO6575ELSEIFBN$="PSTP"THEN58ØØ
6527 PRINT@386,"Type bowler's number and press <ENTER>"; : INPUT
B9 : K=K+1
653Ø CLS : GOSUB52ØØ : PRINT@72,"Bowler number";B9;"named ";BN$;
 : PRINT@13Ø,"of ";TN$;
654Ø FORI=1TOGB : PRINT@322,"Type Game";I;"score and press
 <ENTER>"; : INPUTG(I) : PRINT@359,"   "; : NEXTI
655Ø PRINT@45Ø,"Type bowler's one-game handicap and press
<ENTER>"; : INPUTGH : SE=Ø : FORI=1TOGB : SE=SE+G(I) : NEXTI
6555 PRINT@7Ø6,"The";GB;"games and scratch series are..."; : FOR
I=1TOGB : PRINT@(766+6*I),G(I); : NEXTI : PRINTSE; : PRINT@834,
"One-game handicap =";GH;"  Are they correct (Y or N)"; : INPUT
AN$ : IFLEFT$(AN$,1)="Y"THENAN$="" : GOTO656Ø : ELSEGOSUB555Ø :
GOTO654Ø
656Ø GOSUB9ØØØ : GOSUB71ØØ : IFK=15THEN657Ø
6565 GOSUB555Ø : GOTO652Ø
657Ø CLS : GOSUB52ØØ : PRINT@72,"Were there any blind scores (Y
or N)"; : INPUTAN$ : IFLEFT$(AN$,1)="Y"THENAN$="" : GOSUB61ØØ
6575 CLS : GOSUB52ØØ : PRINT@13Ø,"Type number of points won by
this team and press <ENTER>"; : PRINT@194,"(enter 1/2 point as
.5)"; : INPUTWN : GOSUB92ØØ : GOSUB81ØØ : GOSUB5ØØØ
658Ø CLS : GOSUB52ØØ : PRINT@194,"Do you want a printout now for
team number";T9; : PRINT@258,"(Y or N)"; : INPUTAN$ : IFLEFT$
(AN$,1)="Y"THENAN$="" : GOSUB93ØØ
6585 CLOSE 1 : GOSUB55ØØ : GOTO651Ø
66ØØ CLS : PRINT@7Ø,"Do you need to delete anybody before
entering a NEW bowler"; : INPUTAN$ : IFLEFT$(AN$,1)="Y"THEN
AN$="" : GOSUB67ØØ
66Ø5 CLS : GOSUB52ØØ : GOSUB555Ø : PRINT@72,"Type name of new
bowler for team number";T9; : PRINT@13Ø,TN$; : PRINT@198,
"(maximum 24 characters, including spaces)"; : PRINT@258,"and
press <ENTER>"; : INPUTBN$
661Ø PRINT@328,"Type new Bowler Number and press <ENTER>"; :
INPUTB9
6615 FORI=1TOGB : PRINT@45Ø,"Type Game";I;"score and press
<ENTER>"; : INPUTG(I) : PRINT@487,"   "; : NEXTI
662Ø PRINT@514,"Type bowler's one-game handicap and press
<ENTER>"; : INPUTGH : FORI=1TOGB : SE=SE+G(I) : NEXTI
6625 PRINT@7Ø6,"The";GB;"games and scratch series are..."; : FOR
I=1TOGB : PRINT@(766+6*I),G(I); : NEXTI : PRINTSE; : PRINT@834,
"One-game handicap =";GH;"  Are they correct (Y or N)"; : INPUT
AN$ : IFLEFT$(AN$,1)="Y"THENAN$="" : GOTO663ØELSEGOSUB555Ø :
GOTO6615
663Ø GOSUB9ØØØ : GOSUB71ØØ
664Ø PRINT@68Ø,"ANY MORE (Y or N)"; : INPUTAN$ : IFLEFT$(AN$,1)=
```

```
"Y"THENAN$="" : GOTO66ØØ
665Ø NB=Ø : RETURN
67ØØ CLS : PRINT@262,"You must delete a bowler from the roster.
"; : PRINT@45Ø,"Type the NUMBER of the bowler to be dropped" :
INPUTB9
671Ø GOSUB7ØØØ : PRINT : PRINT"You want to drop bowler number";
B9;BN$;"  (Y or N)"; : INPUTAN$ : IFLEFT$(AN$,1)="N"THEN67ØØ
672Ø BN$=STRING$(24,32) : H1=Ø : H2=Ø : TP=Ø : TG=Ø : AV=Ø : HD=
Ø : GOSUB71ØØ
673Ø RETURN
7ØØØ ' GET subroutine for bowlers' records *
7Ø15 IFEOF(1)THENPRINT@96Ø,"END OF FILE "F1$;
7Ø2Ø GET 1,B9
7Ø3Ø X9=CVI(A$) : Y9=CVI(B$) : BN$=C$ : H1=CVI(D$) : H2=CVI(E$)
: TP=CVI(F$) : TG=CVI(G$) : AV=CVI(H$) : HD=CVI(I$)
7Ø5Ø RETURN
71ØØ ' PUT subroutine for bowlers' records *
712Ø LSETA$=MKI$(T9) : LSETB$=MKI$(B9) : LSETC$=BN$ : LSETD$=
MKI$(H1) : LSETE$=MKI$(H2) : LSETF$=MKI$(TP) : LSETG$=MKI$(TG) :
LSETH$=MKI$(AV) : LSETI$=MKI$(HD)
713Ø PUT 1,B9
715Ø RETURN
8ØØØ ' GET subroutine for teams' records *
8Ø1Ø GET 2,RT
8Ø15 IFEOF(2)THENPRINT@96Ø,"END OF FILE "F2$;
8Ø3Ø Z9=CVI(M$) : TN$=N$ : H3=CVI(O$) : H4=CVI(P$) : TT=CVS(Q$)
: AT=CVI(R$) : GW=CVS(S$) : GL=CVS(T$) : GT=CVI(U$)
8Ø5Ø RETURN
81ØØ ' PUT subroutine for teams' records *
812Ø LSETM$=MKI$(T9) : LSETN$=TN$ : LSETO$=MKI$(H3) : LSETP$=
MKI$(H4) : LSETQ$=MKS$(TT) : LSETR$=MKI$(AT) : LSETS$=MKS$(GW) :
LSETT$=MKS$(GL) : LSETU$=MKI$(GT)
813Ø PUT 2,RT
814Ø RETURN
9ØØØ TG=TG+GB : FORI=1TOGB : IFG(I)=ØTHENTG=TG-1
9Ø1Ø NEXTI
9Ø2Ø FORI=1TOGB : S1=S1+G(I) : NEXTI : IFS1>H2THENH2=S1
9Ø3Ø FORI=1TOGB : IFG(I)>H1THENH1=G(I)
9Ø4Ø NEXTI
9Ø5Ø TP=TP+S1 : AV=INT(TP/TG) : HD=INT(HP*(PF-AV)) : IFHD<ØTHEN
HD=Ø
9Ø6Ø FORI=1TOGB : T(I)=T(I)+G(I)+GH : NEXTI
9Ø7Ø PRINT@578,"Bowler's series =";S1;" High Game to date =";H1;
: PRINT@65Ø,"High Series to date =";H2;
9Ø8Ø GOSUB5ØØØ : RETURN
92ØØ GT=GT+GB : FORI=1TOGB : T4=T4+T(I) : NEXTI : GW=GW+WN : LT=
PP-WN : GL=GL+LT : TT=TT+T4 : PRINT@641," "; :  : FORI=1TOGB :
PRINTTAB(7*I)"Game";I; : NEXTI : PRINT" Total"; : PRINT@7Ø5," ";
: FORI=1TOGB : PRINTTAB(7*I)T(I); : NEXTI : PRINT"   ";T4;
921Ø AT=INT(TT/GT) : IFT4>H4THENH4=T4
922Ø FORI=1TOGB : IFT(I)>H3THENH3=T(I)
923Ø NEXTI
924Ø PRINT@77Ø,"Hi Team Game to date=";H3;" Hi Team Series to
date=";H4;
925Ø PC(RT)=GW/(GW+GL) : A9$(RT)=TN$ : GW(RT)=GW : GL(RT)=GL :
TT(RT)=TT : AT(RT)=AT
926Ø RETURN
927Ø RETURN
93ØØ GOSUB53ØØ : LPRINTCHR$(27)+CHR$(2Ø) : GOSUB54ØØ : LPRINT
"Team #";T9;TN$;TAB(4Ø)GW;TAB(53)GL;TAB(63)"  ";TT;STRING$
(8,32);AT
931Ø LPRINTTAB(3)"Bowler";TAB(23)"Handicap";TAB(32)"Total Pins";
TAB(44)"Games";TAB(52)"Average";TAB(63)"Bowler";STRING$(17,32);
"Handicap";"  Total Pins";"  Games";"  Average" : BØ=1
932Ø FORI=1TOLOF(1)STEP2 : B9=I : GOSUB7ØØØ : LPRINTB9;BN$;
TAB(24)HD;TAB(34)USING"#####";TP;TAB(45)USING"##";TG;TAB(53)AV;:
B9=B9+1 : IFEOF(1)THENLPRINT" " : RETURN
933Ø GOSUB7ØØØ : LPRINT  ";B9;BN$;" ";HD;STRING$(6,32);USING
"#####";TP;"      ";USING"##";TG;"    ";AV : NEXTI
934Ø RETURN
94ØØ CLS : PRINTCHR$(23) : PRINT@516,"I'm ready to print out"; :
PRINT@648,"League Standings..."; : GOSUB51ØØ : PRINT@776,
"W O R K I N G ...";
95ØØ M=T8+1
951Ø M=INT(M/2) : IFM=ØTHEN961Ø
952Ø FORST=1TOM
953Ø I=ST : J=ST+M : SW=Ø
9535 IFPC(I)=PC(J)THENIFAT(I)>AT(J)THEN956ØELSE955Ø
954Ø IFPC(I)>PC(J)THEN956Ø
955Ø SW=1 : XX=PC(I) : PC(I)=PC(J) : PC(J)=XX : XX$=A9$(I) :
A9$(I)=A9$(J) : A9$(J)=XX$ : YY=GW(I) : GW(I)=GW(J) : GW(J)=YY :
ZZ=GL(I) : GL(I)=GL(J) : GL(J)=ZZ : Z4=TT(I) : TT(I)=TT(J) :
TT(J)=Z4 : Z5=AT(I) : AT(I)=AT(J) : AT(J)=Z5
956Ø I=J : J=J+M : IFJ<T8+1THEN9535
957Ø IFSW=ØTHEN959Ø
958Ø GOTO953Ø
959Ø NEXTST
96ØØ GOTO951Ø
961Ø PRINT@859,"SORT DONE";
962Ø GOSUB53ØØ : LPRINT" " : LPRINTTAB(19)"LEAGUE STANDINGS
AFTER WEEK OF ";DT$ : LPRINTTAB(5)"Team";TAB(3Ø)"Won";TAB(37)
"Lost";TAB(45)"Total Pins";TAB(59)"Average" : LPRINTSTRING$
(66,48)
963Ø FORL=1TOT8 : LPRINTL;A9$(L);TAB(3Ø)GW(L);TAB(38)GL(L);
TAB(46)TT(L);TAB(6Ø)AT(L) : NEXTL
964Ø CLS : PRINTCHR$(23) : PRINT@454,"THIS LEAGUE FINISHED" :
CLOSE2 : CLOSE : END
97ØØ F1$="BOWLERS/"+LEFT$(TN$,3)+" : 1" : OPEN"R",1,F1$,4Ø :
FIELD 1,2 AS A$,2 AS B$,24 AS C$,2 AS D$,2 AS E$,2 AS F$,2 AS
G$,2 AS H$,2 AS I$
971Ø RETURN
98ØØ F2$="TEAMS/"+LF$+" : 1" : OPEN"R",2,F2$,46 : FIELD2,2 AS
M$,24 AS N$,2 AS O$, 2 AS P$,4 AS Q$,2 AS R$,4 AS S$,4 AS T$,
2 AS U$
981Ø RETURN
11ØØØ CLS : PRINT@45Ø,"AN ERROR WITH CODE ";ERR/2+1:PRINT :
PRINT"HAS OCCURRED IN LINE ";ERL:GOSUB5ØØØ
11Ø1Ø RESUME13Ø
```

John F. Rogers
600 Seventh Street
Morgan City, LA 70380 ■

BEGINNER'S CORNER

continued from page 33

RENUM U	The basic text is searched for errors or for Undefined lines. No alterations to the code are performed.
RENUM X	This option will not declare any errors due to calls to undefined line numbers that may or may not be outside the specified range being renumbered.
RENUM U,X	Checks for errors (syntax, bad line number, no line numbered ####) but will not declare numbers out of range as an error.

****

The purpose of the X option is for programmers who use the overlay system of programming. If you write small sections of code that call (GOTO or GOSUB) to subroutines in some other sections, then this option will let you renumber without error messages, because once an error is encountered the line renumbering process stops.

EXAMPLE 5 has the commands for NEWDOS80 1.0

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

        NEWDOS8Ø 1.Ø
Where only a " , " appears, it means that the default
value is used.
        RENUM ,
        RENUM new starting #
        RENUM ,increment
        RENUM ,,old starting #
        RENUM ,,,old ending #
  Here comes the good stuff, combining these nightmares.
        RENUM new starting #,new increment
        RENUM new starting #,,old starting #
        RENUM new starting #,,,old ending #
ETC ETC ETC ETC ETC (I hope you get the idea)
**************************************************************
```

The documentation for these two operating systems begins to differ at this point. NEWDOS80 2.0 became simplified, whereas the documentation for NEWDOS80 1.0 could have been less verbose with the results being further clarity. (Could you repeat that?)

The difference between a block move and a renumbering of lines is this: in a block move, references to lines within the BASIC code will not be changed. In line renumbering, all references to lines within the specified block or range of code will automatically be changed.

For example, let's say that line 10 is a loop that goes (I beg your pardon) back to (redundant?) line 10. (I thought I said that.) The idea is that you want to renumber it to line 20. When this is accomplished, the reference within the line to line 10 will also be changed to line 20.

What happens if an error is found? Good question. On error, as you would expect, an error message is displayed. They look something like EXAMPLE 6.

```
**************************************************************
EXAMPLE 6

        #####/U         There is no text with line #####
        #####/X         Syntax error in line #####
        #####/S         line number ##### is a bad line
                         number
**************************************************************
```

The default values for the new starting numbers are in EXAMPLE 7.

```
**************************************************************
EXAMPLE 7

LABEL                                   DEFAULT VALUE
New starting # and Increment #           1Ø
Old starting # and Old ending #          Ø
None of these numbers may be greater than 65529
OR LESS THAN 1. (This means no line Ø.)
**************************************************************
```

I don't want to get too deep into a tutorial on using RENUM (for now), because that isn't the point of this particular article. Let it be enough to say that using the RENUM command is like an extended AUTO command, where the new starting number is where you want the code relocated. The increment step determines the numeric difference between lines i.e RENUM 100,20. The starting line is 100 and the next line will be 120.

If you use the default (,) character, then the step will be by ten. The old starting number and the old ending number set up a range within which the RENUM command must work (i.e RENUM,,1500,1800). What this does is to tell the interpreter that the range cannot include line numbers less than 1500 or more than 1800.

OK. Now that some of the basics have been shown, we get to comparison shop and explore those other DOS's.

```
**************************************************************
EXAMPLE 8
TRSDOS   DOSPLUS   MULTIDOS   NEWDOS(2.Ø)  RESULTS
         3.3
U-A      U-A       U-A        U-A          List previous line
D-A      D-A       D-A        D-A          List next line
.        .         . or L     .            List current line
,        ,         , or E     ,            EDIT current line
S-U-A    S-U-A     S-U-A      S-U-A        List first line
         or ;                 or ;
S-D-A+Z  /         S-D-A      S-D-A or /   List last line
L##      L##       List##     L##          List line ####
N/A      N/A       /          N/A          List where break
                                           occurred
E##      E##       EDIT       E##          Edit line ####

N/A      N/A       D          N/A          Delete current line
D##      D##       D##        D##          Delete line ####
A##,##   A##,##    N/A        A##,##       Auto numbering begin##
                                            increment##
N/A   CMD"RENUM    see text   RENUM        RENUM
N/A   CMD"RENUM,!    :        RENUM U      Same as RENUM U
N/A      N/A       N/A        @            List page down
N/A      N/A       N/A        :            List previous page
N/A      N/A       P          L            List page from current
                                            line
N/A      N/A       P##        L##-         List page from line ##
CONT     CONT      C          CONT         Continue execution
RUN      RUN       R          RUN          Run program
**************************************************************
```

Boy, that's some list. OK, now we can see some of those differences at a glance. Let me explain the "see text" in the Multidos column. The difference is really very simple. To renumber a a section of code simply use this format ": new start, increment, old start, old end". I didn't have enough room in the chart to show this.

Well, I've covered a lot of territory, but I'm not quite finished yet. There are a number of functions I'd like to cover in Multidos that are quite unique and very powerful. They're called GLOBAL EDITs, and they're similar to things that word processors can do, so I think I'll save it for next time, when I'll discuss using your BASIC editor as a word processor. Until next time, good reserved words to you.

Spencer Koenig
153-27 73 Avenue
Flushing NY 11367 ■

SHORT-TYPE

Frank Tymon

The home computer assumes an ever-greater importance. Data processing, Word Processing, communications! All are finding their niche. Efficiency in the equipment use becomes ever more important. This article covers efficient use of the word processing capability.

SPEED

Speed! Accuracy! Effectiveness!

Essential! The essence of Word Processing, but rarely attained in full. Short-Type, the missing element, speeds up your output immediately! The following pages explain how to use and profit from Short-Type. But first some background.

EFFICIENCY

Efficiency in the office is a key selling point for Word Processing. Similarly, the home owner of a small computer is concerned with efficiency. Developers of Word Processing routines are continuously looking for viable improvements in their product. But potential for significant increases in efficiency is often overlooked. One such area I identify as a form of mechanized shorthand.

GROWTH

Word processing continually becomes a more widely used tool. Office after office finds the initial estimate of usage is low. As users become familiar with the capabilities of Word Processors, they find more ways to profit from them. Typically, the initial user finds that he has grossly underestimated the Word Processing workload. Subsequently his co-workers recognize the ease with which he is getting results and become believers. There is a domino effect. The rapidity with which this domino effect occurs is often amazing. An initial machine is often overloaded within a year. An upgrade or replacement is then necessary.

Certainly the ease with which a draft can be corrected, rewritten, and printed in final form is a major selling point. When we add, for example, mailing list functions, a new set of users are hooked. As time passes each succeeding release of Word Processing software incorporates ever more usefulness. Mathematical routines, list processing, and glossary are the current buzzwords, and new capabilities are arising daily.

HISTORY

Of course, many of the current Word Processing characteristics derive from two predecessor activities — secretarial work and data processing. This has both good and bad aspects. For example, the Word Processor keyboards have retained the inefficiencies of the QWERTY keyboard. But they have incorporated extensions, such as programming function keys. The use of this equivalent to subroutines is a major advantage. the list of advances could be checked off over many areas. Innovativeness has been largely that of extension, rather than quantum jumps. Even so, innovativeness is there. Interestingly enough, many of the advances are occurring in the home computer area.

The use of the function keys has been expecially effective for triggering complex activities with a single keystroke. Merging, copying, moving — have all all been tied to one, or at least only a very few keystrokes. The simple "Delete" function is generally only a 1 or 2 keystroke instruction. "Insert" has similar characteristics.

However, another area of potential value has been only lightly touched on. This area has a great deal of potential in various situations. And it has a great deal of flexibility. To some extent it may be pre-programmed into the particular Word Processor. But it lends itself to local implementation and development also. I call it Short-Type.

Perhaps Short-Type can best be explained through illustration. I am involved in typing a letter which repetitively uses the word "then". Unfortunately, I have developed a quirk. Whenever I type "th", or at least quite often, I end up with "ht". I transpose the letters.

An example of htis could easily be developed. But, hten, what can be done? Hte appropriate hting would be for me to correct my problem. However, htere is difficulty in teaching an old dog new tricks. Htus, I repetitively produce typing with the quality of htis paragraph.

Ah, but all is not lost!

I recognize my weakness, and I also have a Word Processor. I can go back and change each of the entries. This is a tremendous advance over only a few years ago, when I would be busily erasing, opaqueing, and retyping the entries. However, perhaps there is a better way yet.

And indeed there is. In just about all modern word processing programs we have a useful routine called, among other terms, "Replace". And in addition, we have some degree of global replace ability. So now I can go back and, using global replace, turn each "ht" into a "th". Again, a marked improvement in efficiency.

There is an even more recent advance, called the Dictionary, which could examine my input, detect my spelling errors, and correct them. As this is not a standard part of the normal Word Processing package, let us consider other approaches to solving my personal "quirk".

Ah, but all is not lost!

I recognize my weakness, and I also have a Word Processor. I can go back and change each of the entries. This is a tremendous advance over only a few years ago, when I would be busily erasing, opaqueing, and retyping the entries. However, perhaps there is a better way yet.

And indeed there is. In just about all modern word processing programs we have a useful routine called, among other terms, "Replace". And in addition, we have some degree of global replace ability. So now I can go back and, using global replace, turn each "ht" into a "th". Again, a marked improvement in efficiency.

There is an even more recent advance, called the Dictionary, which could examine my input, detect my spelling errors, and correct them. As this is not a standard part of the normal Word Processing package, let us consider other approaches to solving my personal

"quirk".

It is at this point that Short-Type comes into play. Rather than brute-force blundering through, let us consider the problem. I do not always make a mistake with the leading "th". "The", for example, is rarely typed wrong. But "then" is rarely typed right. I have a "then"-quirk. This oversimplifies, but gives me a chance to illustrate a minor advantage of Short-Type. In the next paragraph I rather frequently use the word "hten" — rather, "then".

Accepting what I have said so far, hten, let us proceed. Now and hten we need to analyze what we are doing. Hten, we must determine if there is a better way. If there is, we must hten pursue it.

Knowing my weakness, suppose I don't worry about hten — oops! — then. Rather, suppose I insert + whenever the word occurs. Then, at the end of the document, I use the global replace, and plug in "then" for every occurrence of "+". Let's give it a try. I'll retype the above paragraph, putting "+" in for each occurrence of "then". Subsequently, I'll use a global replace to insert "then" for the symbol.

Accepting what I have said so far, +, let us proceed. Now and + we need to analyze what we are doing. +, we must determine if there is a better way. If there is, we must + pursue it.

In preparing this article, I am using SCRIPSIT. The REPEAT comand provides me a degree of globality, and BREAK followed by R> allows me to replace entries. Below, I have a copy of the above paragraph after using the REPEAT command, followed by BREAK R>+>then, and ENTER.

Accepting what I have said so far, then, let us proceed. Now and then we need to analyze what we are doing. then, we must determine if there is a better way. If there is, we must then pursue it.

Our handy-dandy usage of Short-Type has resulted in my typing relatively few occurrences of "then", and has allowed me substitute a single symbol for the entire word. It has also resulted in the word being correct in all its occurrences. You have undoubtedly noticed a glitch in the result. When "then" introduced a sentence, it didn't begin with a capital letter. So we do need to polish up the product. This is a relatively minor problem as compared with the original one, however.

USES

The illustration is actually trite (nevertheless, the simple procedure still can prove useful!). But there are more realistic problems which can be solved by this approach. For example, examine the next paragraph.

Antidisestablishmenttarianism is a somewhat long word which one seldom types. But if one typed antidisestablishmenttarianism frequently — and even if one spelled it right — one might tire of such frequent typing. I have no quarrel to pick with antidisestablishmenttarianism. I really think antidisestablishmenttarianism is an impressive word. But antidisestablishmenttarianism is not a word I would like to type often.

Suppose, as before, we use "+" to represent the word giving us trouble, "antidisestablishmenttarianism". Subsequently, we go to a global replace function and replace our simple symbol with the complicated word. In the next couple of paragraphs we illustrate:

+ is a somewhat long word which one seldom types. But if one typed + frequently — and even if one spelled it right — one might tire of such frequent typing. I have no quarrel to pick with +. I really think + is an impressive word. But + is not a word I would like to type often.

antidisestablishmenttarianism is a somewhat long word which one seldom types. But if one typed antidisestablishmenttarianism frequently — and even if one spelled it right — one might tire of such frequent typing. I have no quarrel to pick with antidisestablishmenttarianism. I really think antidisestablishmenttarianism is an impressive word. But antidisestablishmenttarianism is not a word I would like to type often.

These are but two simple examples of the concept. Admittedly, you do not reverse t and h when you type hten — I mean, then. And I bet you never bothered to type "antidisestablishmenttarianism" before! However, if you have prepared a chemistry paper of any length, or a biology report, or a variety of other technical reports — then you know how often grotesquely long technical words repetitively pop up. This simple technique cuts that problem down to size.

EXTENSIONS

By extension, we handle the occurrence of two such problems. We merely assign one symbol to one of the villains, and a different symbol to the other. We might use, say, "+" for "then" and "–" for "antidisestablishmentarianism". With 2 passes of our global replace we now have resolved both problems. But, to play safe, let us test it.

The next paragraph has a few +'s and –'s dotted here and there, and the paragraph after that is the same — except we have gone back and globally adjusted to replace both signs by the appropriate entry.

Now and + we type such trash as –. –, fortunately, +, doesn't happen too often. But, +, if – did, we still must do our thing.

Now and then we type such trash as antidisestablishmenttarianism. antidisestablishmenttarianism, fortunately, then, doesn't happen too often. But, then, if antidisestablishmenttarianism did, we still must do our thing.

TYPICAL USAGE

I believe we have made our point. This technique is indeed a useful tool in the office. And wherever the Word Processor is used. Being lazy, I use it often when typing documents. I might refer to one use which you probably didn't recognize. Do you really believe I repetitively typed "Word Processor" throughout this document? No way! Knowing that it would show up from time to time, I used a symbol instead. I could have been "#", but then I would have had to replace it in this paragraph, so I used a different symbol. However, in the next paragraph, I'll use "#", then replace it in the following paragraph. Replace it with a global replace, of course. This emphasizes a point. The replacement for the symbol need not be only a word. It can be a pair of words, a phrase, or the like.

When writing about # one may either type it out or use a symbol. Of course, the symbol for # is short. It is hard to misspell. I rarely misspell #. That isn't true for other spelling. I have occasionally misspelled &, but that

is more difficult, of course.

When writing about Word Processing one may either type it out or use a symbol. Of course, the symbol for Word Processing is short. It is hard to misspell. I rarely misspell Word Processing. That isn't true for other spelling. I have occasionally misspelled &, but that is more difficult, of course.

SECTION SEPARATION

For internal housekeeping in your letter, article, or whatever document, you might need to separate sections. There are a variety of ways to do this. A new page at the end of each section, for example. Or a section number inserted. One simple way is to insert a line across the page. Let's save a few keystrokes, however, by using Short-Type. At the end of this paragraph, and of the subsequent one, I'll insert "<". I'll leave the symbol as it is at the end of this paragraph, to illustrate how it would appear in practice, but I'll use a global replace at the end of the subsequent paragraph.

<

Short–Type is a powerful concept. The few illustrations so far establish its importance. But is would be useful even if used only for partial words. And, in fact, with properly trained operators, the time savings using partial words might exceed all others.

We could, certainly, pretty-up our line, and probably would do so. Or we might only use a partial line in the center of the page. The key point is that a single entry suffices to input the entire line. Let's continue to look at the use of our approach for partial words.

PARTIAL WORDS

How would we use Short–Type for partial words? Simple!

Examine a few common words. Establishment, operation, going, appearance, etc. Suppose our secretary had been trained to type "+" when a word contained "ment", "–" if the word contained "tion", "*" in those cases when "ing" ends a word, and "$" where "ance" appears. The secretary would be typing a single keystroke for the suffix, one symbol instead of three or four. Even one symbol for such suffixes as "ly", "ful", "ate", and others. And, for an occasional prefix, such as "pre", "con", etc., we might substitute ">" "@", or the like. How many, of the millions of keystrokes struck each year, could be replaced?

Using only a subset of the possibilities, let's try this out. In the next paragraph I'll use the above symbols. Then I'll redo the paragraph, and apply a global replace. The product should be educational.

Ac–, do*, leads to advance+. The treat+ you receive depends on accept$ of your work.

Action, doing, leads to advancement. The treatment you receive depends on acceptance of your work.

Nor does this exhaust the possibilities! Such common words as "the", "and", "for", "etc.", etc., lend themselves to the same treatment.

"The" occurs quite frequently. So let us use "*" to represent it. The following paragraph is a before-change illustration of the concept, and the paragraph after that shows the result of the global replace.

* time to do * thing is now. * average person delays. * result is bad. * use of "*" is widespread.

The time to do The thing is now. The average person delays. The result is bad. The use of "The" is widespread.

You may have noticed that I capitalized "The", since it occurs most often at the start of a sentence. Of course, a final cleanup would be necessary.

PLACE–HOLDER

There are a variety of other useful devices associated with Short-Type. One very important concept is that of a place-holder. Suppose you have frequent need to refer to some appropriate document, but it is as yet unnamed. Merely insert a symbol in the places where it is to appear, and you have it essentially taken care of.

Of a similar nature is the concurrent development of a document and its contents. The skeleton of the document can be developed, symbols can be inserted in the appropriate unknown areas, and the product brought almost to a conclusion in step with the actual work. When specifics are available, it is simple to then insert them.

It is likely that, in one form or another, all the above uses of Short-Type would be useful at a given facility. However, their selective use can substantially improve the efficiency with which Word Processing is applied. Even a partial implementation can result in substantial increase in office productivity.

CODES

Certainly we tend to run out of symbols in a more extensive, complex document. The answer then becomes the use of codes, where 2 or more symbols, letters, or numbers become the surrogates for the extended entries.

Being realistic, however, we must note that the greatest gain occurs in the initial uses of this approach. As you make the system more and more complicated, the marginal advantages gained become smaller and smaller. In the more complex situation it is necessary to structure the codes, to train the users, etc. Each situation must be analyzed to determine the relative advantages.

TERM DICTIONARY

I would visualize that, in the not-too-distant future, a form of dictionary will be developed which incorporates elements of this approach. A subset of symbols and/or codes will be associated with certain of the frequent dictionary entries. When these are typed, an automatic global replace will occur. Actually, there is a type of dictionary called a term dictionary, which equates codes to technical terms.

FREQUENCY DISTRIBUTION

It's important to recognize the relative frequency of words, and of suffixes and prefixes. For example, the two words, "the" and "and" constitute roughly 10 percent of a normal document. Codifying these with, perhaps, an asterisk and a pound sign, saves two keystrokes for the occurrence of each. Thus, for our

document, we have saved ⅗ of 10 percent — about 7 percent. Not earthshaking, yet significant. Realistically, when we look at the next most frequently occurring words, many of them are only two characters long, and savings are minimal. In the first 25 percent, however, the words "that", "you", and "for" occur. So we may use single digit codes for these and further increase our efficiency. Unfortunately, we do not see much opportunity for increased efficiency by coding other common words. We need another approach. (However, if you are working in a technical area, there may be many words worth coding.)

We can use a coding approach with suffixes and prefixes. For example, "–ould" is a relatively frequently occurring suffix. A single or double digit code could well be used here. A prefix such as "after–" is common. There are certainly others, and you might, for your own typing, develop a set of codes which you use repetitively for saving time.

However, we can easily go overboard. After the most common usages are coded, it becomes increasingly difficult to find a good candidate for coding. But you can look at the type of documents you produce, and then decide whether to go to greater depths.

If you use your home computer for Word Processing, then don't overlook Short-Type. After all, you don't want to waste time and effort typing #, or +, or –. Or, Heaven forbid, &!

Frank Tymon
1213 Topaz Way
Santa Monica, California 93454 ■

ASK RICHARD

continued from page 28

your computer. Very often, large machinery connected on the same circuit as a computer will create difficulties with the computer when the machinery is turned on or off. If a dedicated line is used, no machinery can interfere with the computer.

(NOTE: If you are experiencing electrical problems or crashed disks which occur at specific intervals throughout the day, there is a good chance heavy machinery is the cause. For example, I recently spoke with someone who was using a computer in a school. Whenever the boiler was turned on in the morning his disks would crash. Simply moving the computer to a different outlet solved this problem.)

Questions from readers on all aspects of personal computing are welcomed. I will try to reply to all inquiries, either personally or through this column. Please enclose a self-addressed, stamped envelope with your letter.

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

POCKET COMPUTER CORNER

continued from page 44

```
14:NEXT A
20 Q=0
21:FOR A=1 TO C:P=A+9:R=A+3
22:IF A(P)=A(R) LET Q=Q+1
23:NEXT A
30:IF Q=C PAUSE "KILLED THE KILON":BEEP I:PRINT I;
"SHOTS USED":
   GOTO 1
31:PAUSE "MISSED":BEEP I:PRINT I;" SHOTS USED":GOTO 11
```

SUMMARY

This program can be used as a small fun game which can be keyed into the pocket computer in a few minutes. It can also be used as a base for building more complex games. In either case, we hope you have fun using and playing the game.

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

Stanley M. Zimmerman
Olensky Bros. Computers
3763 Airport Blvd.
Mobile, Alabama 36608 ■

TWO BASIC PROGRAMS

William H. Patrick

ALPHABET PUZZLE

This is the computer version of the dime store game or rearranging the alphabetic characters on a square with one empty position. Rearrange the letters in order, and the computer will replay all of your moves. It's like watching a time-lapse movie of what you did. Movement is done simply by pressing the key of the letter you wish to move and an arrow key for the direction in which you want to move it. Remember, you can only move to the empty square!

```
Ø REM ALPHAPUZ/BAS - A MANIPULATIVE ALPHABETICAL PUZZLE BY
      WILLIAM H. PATRICK, RT7 PARADISE CAMP ROAD,
HARRODSBURG,        KENTUCKY 4Ø33Ø, JUNE 1981
1Ø CLS
14 CLEAR 3ØØØ
15 DIM S$(16),SS$(16),RS$(16),YM$(5ØØ),YD$(5ØØ)
2Ø REM SET SQUARE CHARACTERS
3Ø FOR I=1 TO 16: READ SS$(I): S$(I)=SS$(I): NEXT: DATA
A,B,C,D,E,F,G,H,I,J,K,L,M,N,O," "
35 REM PRINT INSTRUCTIONS
36 CLS: PRINT @ 32Ø,"   YOU WILL BE GIVEN THE FIRST 15 LETTERS
OF THE ALPHABET IN A ": PRINT" 4X4 SQUARE. YOUR TASK IS TO
ARRANGE THEM IN ALPHABETICAL ORDER."
37 PRINT "TO MOVE A LETTER, SIMPLY PRESS THE LETTER AND AN
ARROW KEY FOR ": PRINT "THE DIRECTION YOU WANT THE LETTER TO
MOVE."
38 PRINT "YOU MAY ONLY MOVE TO THE BLANK SQUARE.": PRINT
TAB(2),"GOOD LUCK!"
4Ø RANDOM: FOR X=1 TO 1ØØ: C1=RND(16): C2=RND(16): K$=S$(C1):
S$(C1)=S$(C2): S$(C2)=K$: NEXT: REM MIX UP
45 FOR I=1 TO 16: RS$(I)=S$(I):NEXT
5Ø INPUT"<ENTER> TO BEGIN";PG$:CLS
9Ø GOSUB 4ØØ
1ØØ REM MOVE
11Ø PRINT @ 8ØØ,"LETTER";: IF PG$="R" THEN A$=YM$(MV) ELSE
A$=INKEY$ :IF A$>"O" OR A$<"A" OR LEN(A$)<>1  THEN 11Ø ELSE
PRINT " "A$ :YM$(MV)=A$
111 IF PG$="R" THEN PRINT " "A$
12Ø FOR I=1 TO 16:IF A$=S$(I) THEN SQ=I : GOTO13Ø ELSE NEXT
13Ø PRINT @ 864, "DIRECTION";:IF PG$="R" THEN B$=YD$(MV) ELSE
B$=INKEY$: IF B$="" THEN 13Ø ELSE YD$(MV)=B$
14Ø IF ASC(B$)<>9 THEN 16Ø
141 REM RIGHT ARROW WAS PRESSED
142 PRINT CHR$(94);
143 IF ABS(SQ/4-INT(SQ/4))<.1 THEN GOSUB 1ØØØ:GOTO 11Ø
144 S2=SQ+1:IF S$(S2)<>" " THEN GOSUB 1ØØØ:GOTO 11Ø
145 GOSUB 11ØØ
146 GOTO 3ØØ
16Ø IF ASC(B$)<>8 THEN 18Ø
161 REM LEFT ARROW WAS PRESSED
162 PRINT CHR$(93);
163 IF (SQ-INT(SQ/4)*4)=1  THEN GOSUB 1ØØØ: GOTO 11Ø
164 S2=SQ-1:IF S$(S2)<>" " THEN GOSUB 1ØØØ: GOTO 11Ø
165 GOSUB 11ØØ
166 GOTO3ØØ
18Ø IF ASC(B$)<>1Ø THEN 2ØØ
181 REM DOWN ARROW PRESSED
182 PRINT CHR$(92);
183 IF SQ>=13 THEN GOSUB 1ØØØ: GOTO 11Ø
184 S2=SQ+4:IF S$(S2)<>" " THEN GOSUB 1ØØØ: GOTO 11Ø
185 GOSUB 11ØØ
186 GOTO 3ØØ
2ØØ IF ASC(B$)<>91 THEN 11Ø
2Ø1 PRINT CHR$(91)
2Ø2 IF SQ<=4 THEN GOSUB 1ØØØ: GOTO 11Ø
2Ø3 S2=SQ-4:IF S$(S2)<>" " THEN GOSUB 1ØØØ: GOTO 11Ø
2Ø4 GOSUB 11ØØ
3ØØ REM COUNT AND MAKE MOVE
3Ø5 FOR CT=1 TO 2
31Ø P1=INT(SQ/4):P2=SQ-P1*4
315 IF P2=Ø THEN P1=P1-1:P2=4
32Ø P=64*(P1+6)+2Ø+P2*5
33Ø PRINT @ P,S$(SQ);
34Ø SQ=S2
35Ø NEXT CT
36Ø MV=MV+1
37Ø PRINT @ 992,"MOVE ";MV;
38Ø REM CHECK TO SEE IF FINISHED
381 FOR I=1 TO 16
382 IF S$(I)<>SS$(I) THEN PRINT @ 8ØØ,"          ";: PRINT
@ 864,"          ";: GOTO 11Ø
383 NEXT I
384 PRINT "CONGRATULATIONS!"
385 INPUT "WOULD YOU LIKE ANOTHER TRY WITH THE SAME BOARD? OR
REVIEW YOUR  MOVES? (ENTER Y,N OR R)";PG$:IF LEFT$(PG$,1)="Y"
 OR PG$="R" THEN CLS: MV=Ø:FOR I=1 TO 16:S$(I)=RS$(I): NEXT:
GOTO 9Ø:ELSE END
4ØØ REM DRAW BOARD
4Ø1 FOR P1=1 TO 4
4Ø2 FOR P2=1 TO 4
4Ø3 P=64*(P1+5)+2Ø+P2*5
4Ø4 PRINT @ P,S$((P1-1)*4+P2)
4Ø5 NEXT P2,P1
4Ø6 RETURN
1ØØØ  PRINT @ 928,"INVALID MOVE":FOR KT=1 TO 3ØØ: NEXT: PRINT
@ 928,"          ";: PRINT @ 8ØØ,"          ";: PRINT @ 864,
"          ";
1Ø1 RETURN
11ØØ S$(SQ)=" ":S$(S2)=A$: RETURN: REM SWAP
```

FRACTION CALCULATOR

This program is a calculator that accepts fractions and mixed numbers, and gives you a fractional answer simplified to lowest terms. Use it to do your fraction homework, where a calculator won't help. Examples and inputting instructions are given within the program.

```
Ø CLEAR 1ØØØ
1Ø CLS:DEFINT A-Z:DEFDBL W,N,D
2Ø PRINT @ 64*5,"FOUR FUNCTION FRACTION CALCULATOR"
3Ø PRINT TAB(1), "BY WILLIAM H. PATRICK"
4Ø PRINT TAB(1), "   RT 7 PARADISE CAMP ROAD"
5Ø PRINT TAB(1), "   HARRODSBURG, KENTUCKY 4Ø33Ø"
6Ø FOR TIME =1 TO 1ØØØ: NEXT: CLS
7Ø PRINT @ 64*5, "FOUR FUNCTION FRACTION CALCULATOR -"
8Ø PRINT : PRINT "CORRECT ENTRY OF NUMBERS IS EXTREMELY
IMPORTANT!"
9Ø PRINT:INPUT"ENTER 'I' FOR INSTRUCTIONS";PG$
1ØØ IF PG$="I" THEN GOSUB 1ØØØØ
11Ø CLS
12Ø PRINT "FOUR FUNCTION FRACTION CALCULATOR ": PRINT "ENTRY
READY>"
13Ø GOSUB 2ØØØØ:W1=W:N1=N:D1=D
```

```
135 FOR TIME=1 TO 500: NEXT: PRINT @800, STRING$(152,32);
136 PRINT @ 900,"OPERATION";
140 OP$=INKEY$:IF OP$="" THEN 140
160 GOSUB 20000:W2=W:N2=N:D2=D
170 PRINT @ 800,STRING$(152,32);:A$=INKEY$:IF A$<>"=" THEN
PRINT @ 900,"= EXPECTED";:GOTO 170
180 PRINT @ 900,"CALCULATING ...";
185 CLS: PRINT @ 512, W1"   "N1" / "D1;"     " OP$"    "W2"
  "N2" / "D2
190 IF OP$="+" THEN 1000
200 IF OP$="-" THEN 2000
210 IF OP$="*" THEN 3000
220 IF OP$="/" THEN 4000
225 PRINT "NO SUCH OPERATION":FOR TIME =1 TO 1000: NEXT:
GOTO 260
230 PRINT @ 640, "ANSWER : ";W"  "N" / "D
240 PRINT: PRINT "PRESS 'M' TO STORE IN MEMORY, <SPACE BAR>
TO ENTER NEW PROBLEM";: GOSUB 30000
250 IF A$="M" THEN MW=W:MN=N:MD=D ELSE 260
260 CLS:GOTO 120
1000 REM ADD
1010 IF D1<>D2  THEN D=D1*D2:N1=D2*N1:N2=D1*N2 ELSE D=D1
1020 N=N1+N2:W=W1+W2
1030 IF D>N THEN 1050
1040 A=INT(N/D):N=N-D*A: W=W+A
1050 RN=N:RD=D:GOSUB 40000: N=RN: D=RD: REM REDUCE
1060 GOTO 230
2000 REM SUBTRACT
2010 IF D1<>D2 THEN D=D1*D2:N1=D2*N1:N2=D1*N2 ELSE D=D1
2020 IF N1>N2 THEN 2050
2030 IF W1=0 THEN PRINT "CAN'T SUBTRACT!": FOR TIME=1
TO 1000: NEXT : GOTO 260
2040 W1=W1-1:N1=N1+D:REM BORROW
2050 N=N1-N2:W=W1-W2
2060 RN=N:RD=D:GOSUB 40000:N=RN:D=RD
2070 GOTO 230
3000 REM MULTIPLY
3010 N1=D1*W1+N1:N2=D2*W2+N2
3020 N=N1*N2:D=D1*D2
3030 W=INT(N/D):N=N-W*D
3040 RN=N:RD=D:GOSUB 40000:N=RN:D=RD
3050 GOTO 230
4000 REM DIVISION
4010 N1=D1*W1+N1:H=D2:D2=D2*W2+N2:N2=H
4020 GOTO 3020
5000 REM MEMORY
5001 W=MW:N=MN:D=MD:GOTO 20210
10000 REM INSTRUCTIONS
10010 CLS
10020 PRINT "TO ENTER A MIXED NUMBER -": PRINT "PRESS WHOLE
NUMBER DIGITS": PRINT "<SPACE BAR>": PRINT "NUMERATOR": PRINT
"     /": PRINT "DENOMINATOR": PRINT "<SPACE BAR>": GOSUB
50000
10021 PRINT: PRINT "EXAMPLE -": PRINT "TO ENTER 3  1/2":
PRINT: PRINT "PRESS": PRINT "3": PRINT "<SPACE BAR>": PRINT
"1": PRINT "/": PRINT "2": PRINT "<SPACE BAR>"
10030 GOSUB 50000
10040 PRINT "TO ENTER A WHOLE NUMBER - ": PRINT "PRESS WHOLE
NUMBER DIGITS": PRINT "<SPACE BAR>": PRINT "<SPACE BAR>"
10045 GOSUB 50000
10046 PRINT "EXAMPLE -": PRINT " TO ENTER   3": PRINT "PRESS
": PRINT "3": PRINT "<SPACE BAR>": PRINT "<SPACE BAR>"
10050 GOSUB 50000
10060 PRINT "TO ENTER A FRACTION -": PRINT "<SPACE BAR>":
PRINT "NUMERATOR": PRINT "   /": PRINT "DENOMINATOR": PRINT
"<SPACE BAR>"
```

continued on page 62

COMPUTRONICS CLASSIFIED

CLASSIFIED ADVERTISING

Introductory Rates (per insertion)	1X	3X	6X	12X
Special discount Price (to 25 words)	$20	$15	$12	$10
Special Discount Add'l charge/word	$.45	$.40	$.35	$.30

To figure cost of ad, consider words like "a", "the", "etc." as one word each. Telephone number with area code counts as two words. Please type or print your ad and send along with payment in full (check, money order, Visa, MasterCard, or American Express) to *H & E Computronics, Inc., Classified Advertising Department, 50 North Pascack Road, Spring Valley, NY 10977.* Your ad will begin in the next available issue.

PROGRAMMER'S AIDE: easy way to keep track of variables. 5 books per package, $4.95. Cypress Computer, 5244 Oxford, Cypress CA 90630. Cash, check, or money order.

STOCK ANALYSIS PROGRAMS FOR TRS-80. For more information, write to:
R. DeCrick
7655 Whispering Brook C21
Portage, MI 49081

"PEEK AND POKE" MORE POWERFUL PROGRAMS". Easy-to-use guide to over 50 special memory addresses for video, cursor, keyboard, USR, memory, more! Only $4.00. David Lewis, Box 88-HEC, Shady, NY 12479.

YOUR AD CAN APPEAR HERE for as little as $10 per month.

PROGRAMS PUBLISHED IN *Computronics* **on diskettes:** $12.00 buys a diskette with **all** the programs published in any single issue of *Computronics* **with corrections.** Specify the issue number you want and whether you want Model I or Model III media (not available for other models). Box A, H & E Computronics, 50 North Pascack Road, Spring Valley, NY 10977, or call toll free order number (800) 431-2818.

TRS-80 Model I Level II with 48K RAM, upper/lower case, disk doubler installed, and one disk drive, all in excellent working condition, $1595.00. Call (914) 634-1821.

TRS-80 Model I Video Displays with **green screens,** $40 each. Box D, H & E Computronics, 50 North Pascack Road, Spring Valley, NY 10977.

TWO BASIC PROGRAMS

continued from page 61

```
10065 GOSUB 50000
10066 PRINT "EXAMPLE -": PRINT: PRINT "TO ENTER  1/2": PRINT:
PRINT "PRESS": PRINT "<SPACE BAR>": PRINT "1": PRINT "/":
PRINT "2": PRINT "<SPACE BAR>"
10070 GOSUB 50000
10080 PRINT "IN SHORT, THE WHOLE NUMBER AND THE FRACTION MUST
BE FOLLOWED": PRINT "BY A <SPACE BAR>. IF EITHER THE WHOLE
NUMBER OR FRACTION IS TO": PRINT "BE OMITTED, AN ADDITIONAL
SPACE BAR MUST BE PRESSED WHERE": PRINT "THE MISSING NUMBER
WOULD HAVE GONE."
10090 PRINT: PRINT "PRESS ANY KEY TO CONTINUE INSTRUCTIONS":
GOSUB 30000:CLS
10100 PRINT "THE CALCULATOR WILL ONLY OPERATE ON TWO NUMBERS
AT A TIME": PRINT "ENTER      NUMBER   OPERATION  NUMBER  ="
10110 PRINT: PRINT "AN ANSWER MAY BE PLACED IN MEMORY FOR A
LATER CALCULATION": PRINT "BY PRESSING THE 'M' WHEN THE
ANSWER IS DISPLAYED"
10120 PRINT: PRINT "PRESS ANY KEY TO BEGIN";:GOSUB 30000:CLS
20000 REM INPUT NUMBER SUBROUTINE
20010 W=0:N=0:D=0
20019 PRINT @ 900,"WHOLE NUMBER";
20020 GOSUB 30000:IF A$="M" THEN 5000
20030 IF A$=" " THEN A$="":GOTO  20100 ELSE W=W*10+VAL(A$)
20040 GOTO 20020
20100 PRINT @ 836, W;: PRINT @ 920, "NUMERATOR    ";: GOSUB
30000: IF A$=" " THEN 20200 ELSE IF A$="/" THEN 20150
20110 N=N*10+VAL(A$)
20120 GOTO 20100
20150 PRINT @ 856, N;: PRINT @ 940,"DENOMINATOR      ";:
GOSUB 30000: IF A$=" " THEN A$="": GOTO 20200
20160 D=D*10+VAL(A$)
20170 GOTO 20150
20200 IF D=0 THEN D=1
20205 PRINT @ 878,D;
20210 RETURN
30000 REM INKEY$ SUBROUTINE
30001 A$=INKEY$:IF A$="" THEN 30001 ELSE RETURN
40000 REM REDUCE RN/RD SUBROUTINE
40010 S=RN:L=RD
40012 IF S=0 THEN 40060
40020 Q=INT(L/S):R=L-Q*S
40030 IF R=0 THEN 40050
40040 L=S:S=R:GOTO 40020
40050 RN=RN/S:RD=RD/S
40060 RETURN
50000 PRINT: PRINT "PRESS ANY KEY TO CONTINUE": GOSUB 30000:
CLS: RETURN
```

William H. Patrick
Rt. 7 Paradise Camp Road
Harrodsburg, KY 40330 ■

PRACTICAL BUSINESS PROGRAMS

continued from page 38

SUMMARY

The program reviewed this month was TRANS, the program which is the starting point for initializing a new set of data. This program allows the user to either update or initiate a new file. It has a routine built in for the purpose of checking to see if the sum of the debits is equal to the sum of the credits at any time. The method of data input follows the double entry bookkeeping system concept.

With this program you can start to make use of the general ledger program. An analysis of a set of transactions may be made for the purpose of checking the balance obtained by the bank in your cash account.

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 ■

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	Your social security number
	Present home address (Number and street, including apartment number, or rural route)		Spouse's social security no.
	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 ___ **Note:** *Checking "Yes" will not increase your tax or reduce your refund.*
If joint return, does your spouse want $1 to go to this fund? . . . Yes ___ No ___

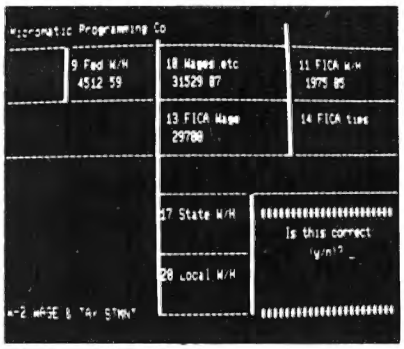
MICROMATIC PROGRAMMING COMPANY

TRS-80* Owners Do Your Own Taxes Like An Expert with TAX/SAVER II™

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp.

FOR MODELS I or III

Now With Printout!



TAX/SAVER II™ — The tax help program for the layman, the professional accountant or tax preparer.

New and expanded, TAX/SAVER II™ offers a different approach to preparing a tax return. Like the original, the new version has the tax regulations programmed in so it is more than just a calculator. Designed for non-accountants, TAX/SAVER II™ asks you questions, just as an accountant does. Based on your answers, it leads you through the tax maze to your lowest legal tax. Then it tells you how to fill in your return, line by line, or it will output to a printer.

TAX/SAVER II™ also has speed features for those doing more than one return. Optional program instruction and tax text make TAX/SAVER II™ the practical system for professional preparers as well. TAX/SAVER II™ has full disk storage of data files (with optional password protection).

The manual includes information on special tax areas, lists of possible deductions and a tax glossary.

TAX/SAVER II™ compares itemized deductions to national averages; automatically computes certain limitations - for example, on medical deductions and contributions; checks for excess FICA; helps determine dependents. Yet, TAX/SAVER II™ offers the privacy and convenience of home use.

The user-oriented design with special screen formatting makes data entry, verification and correction easy. Yet you are always in control. You can skip any help features or parts of the program that you don't need.

TAX/SAVER II™

- Completes long and short forms (1040 & 1040A)
- Itemized Deductions - Schedule A
- Interest & Dividends - Schedule B
- Tax Calculation - Tables, Rates
- Tax Savings Methods - Income Averaging, Maximum Tax, Alternative Tax
- Business Income - Schedules C & SE
- Capital Gains - Schedule D
- Allows you the privacy of your own home
- Lets you help friends and relatives with their taxes
- Has built-in aids. Answers specific questions like "Is my father my dependent?" and "Are my deductions reasonable?"
- Manual includes 1982 tax forms, information on special tax areas, lists of possible deductions, and glossary of tax terms
- Completes long and short forms including itemized deductions, excess FICA, earned income credit, community property, tax calculation (comparing all possible filing statuses in one run)
- Tax regulations are programmed in by our team of accountants. Just type in your figures & you've done your own tax return
- Helps you find the lowest tax
- Discounts on yearly updates
- Prints out on standard IRS forms overlays or plain tractor feed paper
- Accepts totals from all other tax forms not listed here

REVIEWS:

"...well designed and easy to use" - D. Lubar, Creative Computing 1/81
"...TAX/SAVER™ may very well live up to its name"
M. Tannenbaum, CPA, 80 Microcomputing 2/81

CUSTOMERS:

"For the price it can't be beat. I am looking forward to next year. With this program I finish my filing in 2 hours. Thank you."

"Excellent presentation. Tutorial style is one of its most attractive attributes. Finally, a program I can really use!"

PROFESSIONALS:

"This is the perfect program for those doing taxes for others for reasonable fees. It was obviously written by folks that know both programming and tax law." "TAX/SAVER™ is superior."

NEW

PROFESSIONAL TAX/FORECASTER™

TAX/FORECASTER™, the quick tax estimator (with printout) for both 1982 and 1983 lets you revise its estimate by merely changing one or more lines. Use it as a tax planner either together with TAX/SAVER II™ or by itself. TAX/FORECASTER™ lets you quickly ask all of your WHAT IF? Questions and instantly recalculates your taxes. A great tax preparation aid for both the layman and professional. Includes Income Averaging and disk storage of Client files.

• Both TAX/SAVER II™ and TAX/FORECASTER™ are tax-deductible • Discounts are given on yearly updates • Free tax newsletter is issued annually.
With the combined package TAX/SAVER II™ and TAX/FORECASTER™ you can now have, at an affordable price, the power to predict, control and reduce your tax liability.

- ☐ TAX/SAVER II™ (MOD I & III) $139.95 - Manual Included
- ☐ PROFESSIONAL TAX/FORECASTER™ (Requires 48K) (MOD I & III) $99.95 ($84.95 if purchased with TAX/SAVER II™)
- ☐ UPDATE for Registered TAX/SAVER II™ Owners $83.95
- ☐ TAX FORM OVERLAYS $39.95

SYSTEM REQUIREMENTS

- ☐ TRS-80 Model I with 32K and 2 disk drives
- ☐ TRS-80 Model III with 32K and 2 disk drives

COMPUTRONICS INC
MATHEMATICAL APPLICATIONS GROUP

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

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

master charge · VISA · AMERICAN EXPRESS

24 HOUR ORDER LINE
(914) 425-1535

★ *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 AND MEXICO
★ ADD PROPER POSTAGE OUTSIDE U.S., CANADA & MEXICO

— ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE —
— DELIVERY SUBJECT TO AVAILABILITY —

H&E COMPUTRONICS INC.

• EVERYTHING FOR YOUR TRS-80* • ATARI* • APPLE* • PET* • CP/M* •

* TRS-80 is a trademark of the Radio 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.

100 SUPER PROGRAMS

MASTER PAC 100
2nd EDITION (COMPLETELY REVISED)

FOR YOUR TRS-80™ LEVEL II MICROCOMPUTER

ALL ON CASSETTE OR DISKETTE

BUSINESS AND PERSONAL FINANCE

1. CHECKBOOK MAINTENANCE
2. TIME FOR MONEY TO DOUBLE
3. FEDERAL FICA & WITHHOLDING TAX
3. COMPUTATIONS
4. HOME BUDGET ANALYSIS
5. ANNUITY COMPUTATION
6. UNIT-PRICING
7. CHANGE FROM PURCHASE
8. NEBS CHECK PRINTER
9. DAYS BETWEEN DATES
10. MORTGAGE AMORTIZATION TABLE
11. INVENTORY CONTROL
12. PORTFOLIO VALUE COMPUTATIONS
13. VALUE OF A SHARE OF STOCK
14. SALES RECORD KEEPING SYSTEM
15. FUTURE VALUE OF AN INVESTMENT
16. EFFECTIVE INTEREST RATE (LOAN)
17. PRESENT VALUE OF A FUTURE AMOUNT
18. RATE OF RETURN-VARIABLE INFLOW
19. RATE OF RETURN-CONSTANT INFLOW
20. REGULAR WITHDRAWAL FROM INVESTMENT
21. STRAIGHT LINE DEPRECIATION
22. SUM OF DIGITS DEPRECIATION
23. DECLINING BALANCE DEPRECIATION
24. BREAK EVEN ANALYSIS
25. SALVAGE VALUE OF INVESTMENT
26. PAYMENT ON A LOAN
27. FUTURE SALES PROJECTIONS
28. CREDIT CARD FILE
29. ECONOMIC ORDER QUANTITY (EOQ) INVENTORY MODEL
30. VALUE OF HOUSE CONTENTS
31. TEXT EDITOR
32. MONTHLY CALENDAR
33. DAY OF WEEK
34. CASH FLOW VS. DEPRECIATION
35. COMPLETE MAIL SYSTEM
36. INTEREST RATE ON A LEASE

BUSINESS

PERSONAL FINANCE

STATISTICS AND MATHEMATICS

37. RANDOM SAMPLE SELECTION
38. ANGLO-METIC CONVERSION
39. MEAN, STANDARD DEVIATION, MAXIMUM AND MINIMUM
40. SIMPLE LINEAR REGRESSION
41. MULTIPLE REGRESSION ANALYSIS
42. GEOMETRIC REGRESSION
43. EXPONENTIAL REGRESSION
44. SIMPLE MOVING AVERAGE
45. SIMPLE T-TEST
46. CHI-SQUARE TEST
47. NORMAL PROBABILITIES
48. BINOMIAL PROBABILITY
49. POISSON PROBABILITY
50. MATRIX ADDITION AND SUBTRACTION
51. MATRIX TRANSPOSE
52. MATRIX INVERSE
53. MATRIX MULTIPLICATION
54. SOLUTION OF SIMULTANEOUS EQUATIONS
55. QUADRATIC FORMULA
56. LINEAR EQUATION SOLUTIONS
57. ROOT HALF INTERVAL SEARCH
58. ROOTS OF POLYNOMIALS
59. ROOTS-NEWTON'S METHODS
60. PRIME FACTORS OF INTEGER
61. LEAST COMMON DENOMINATOR
62. RADIAN-DEGREE CONVERSION
63. NUMERICAL INTEGRATION

STATISTICS

MATH

UTILITIES

64. QUICK SORT ROUTINE
65. PROGRAM STORAGE INDEX
66. MULTIPLE CHOICE QUIZ BUILDER
67. FORM LETTER WRITER
68. SHELL SORT
69. CASSETTE LABEL MAKER
70. CODES MESSAGES
71. MERGE TWO FILES
72. SORT WITH REPLACEMENT

GRAPHICS

73. DRAWS BAR GRAPH
74. DRAWS HISTOGRAM
75. MOVING BANNER DISPLAY

GAMBLING AND GAMES

76. RANDOM SPORTS QUIZ
77. GOVERNMENT QUIZ
78. HORSE RACE
79. MAGIC SQUARE
80. ARITHMETIC TEACHER
81. HIGH LOW GAMBLE
82. UNSCRAMBLE LETTERS
83. HANGMAN
84. GAME OF NIM
85. RUSSIAN ROULETTE
86. ROULETTE GAME
87. ONE-ARMED BANDIT
88. HIT THE TARGET
89. WALKING DRUNK
90. STATE CAPITAL QUIZ
91. TIC-TAC-TOE
92. DICE GAME
93. LUNAR LANDAR GAME
94. BIORHYTHM
95. HORSE SELECTOR (CLASS CALCULATOR)
96. RANDOM DICE ROLL
97. RANDOM ROULETTE ROLL
98. RANDOM CARD DEALER
99. GUESS THE NUMBER
100. WHITE OUT SCREEN

GAMBLING

INCLUDES 110 PAGE USER MANUAL

GUARANTEED SATISFACTION

30-DAY MONEY BACK GUARANTEE ON ALL SOFTWARE

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

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

PLEASE SEND ME:

- ☐ MASTER PAC 100 CASSETTE VERSION ..... $ 99.95
- ☐ MASTER PAC 100 DISKETTE VERSION ..... $ 99.95
- ☐ MASTER PAC 100 (MODEL II DISKETTE VERSION) ... $149.95

master charge

AMERICAN EXPRESS

VISA

24 HOUR ORDER LINE

(914) 425-1535

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

(800) 431-2818

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

CREDIT CARD NUMBER ..... EXP. DATE.....

SIGNATURE.....

NAME .....

ADDRESS .....CITY.....STATE ..... ZIP .....

*** ADD $3 FOR POSTAGE & HANDLING ADD $4 FOR C.O.D. OR NON UPS AREAS ADD $5 CANADA & MEXICO EXACT POSTAGE ELSEWHERE ***

H&E COMPUTRONICS INC.

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

* 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. · * OSBORNE is a trademark of Osborne Corp.

BUSINESS PAC 100

★ *All orders processed within 24-Hours*
★ *30-Day money back guarantee*

100 Ready-To-Run Business Programs

(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	ANNU1	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	REQWITH	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

☐ **TRS-80 Cassette Version** — **$99.95**
☐ **TRS-80 (Mod-I or III), Pet, Apple or Atari Versions** — **$99.95**
☐ **TRS-80 Mod-II, IBM, Osborne and CP/M Versions** — **$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
DEALER INQUIRIES INVITED

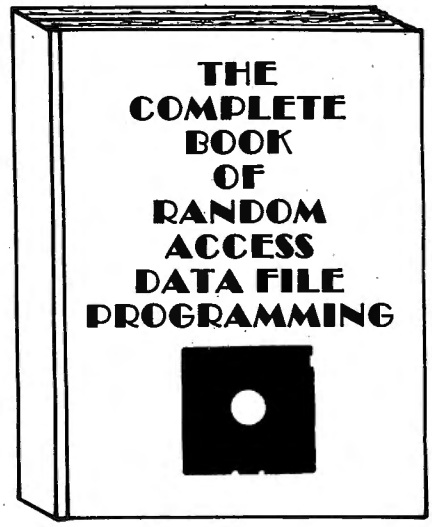
ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE
DELIVERY SUBJECT TO AVAILABILITY

The Complete Book Of Random Access Data File Programming

For TRS-80*, IBM Personal Computer*, Osborne*, and all Microsoft BASIC* computers

The last word on disk random access and file handling techniques, this series is intended for everyone — beginning programmers, businessmen and professionals will learn how to create custom programs to handle inventories, mailing lists, work scheduling, record keeping, and many other tasks, while more experienced programmers will learn advanced, professional programming techniques for faster, more efficient data storage and retrieval.

Although random access file handling is a matter of some complexity, the subject has been treated in a simple and down-to-earth fashion, so that anyone with some small familiarity with programming in Microsoft BASIC will be able to cope with the material. Each stage of learning uses a sample program as a starting point. The programs grow in capability and complexity as the books progress into all of the various aspects of file handling and record manipulation. An extensive effort has been made to keep the material coherent and every program line is explained in detail.



Volume I: Basic File Handling Techniques

- The writing of a Menu to Summarize program functions
- The writing of a screen format to accept data for records
- The creation of the basic record
- The FIELD and LSET routines for buffer preparation
- The writing of the record to disk in a random access mode
- The ability to change or edit a record
- The LPRINT capability from disk using three different formats
- Deleting a record from a random file
- Sorting the random file
- Searching the random file by name or key field
- The ability to search in a "NEXT or PRIOR" fashion
- The ability to purge deleted records from a disk file
- The ability to calculate with data from a disk file
- The provision for future expansion of the data fields
- The use of flags to prevent program crashes
- Date setting, printer on-line and many other routines to make a program run like a commercially-written program

Volume II: Advanced File Handling Techniques

- Blocking & de-blocking, Shell-Metzner sort, In-place screen editing, recovery of deleted record space
- Alpha-index record retrieval, fast machine/BASIC sort
- Linked list record structure and sort-merge, deleted record removal and file reorganization
- Multi-key file reorganization and record searching
- Relational database programming--comprehensive self-balancing accounting system with printouts
- Hashcoded data file manipulation--(probably the fastest method of data retrieval). Hashing the input key and recovery method explained
- Span-blocking techniques (allows; creation of records longer than 256 bytes without wasted space

The Complete Book Of Random Access Data File Programming

Volume I: Basic File Handling Techniques	**$29.95**
optional Vol. I Program Disk for Model I/III	**$28.50**
optional Vol. I Program Disk for Model II	**$32.50**
Volume II: Advanced File Handling Techniques	**$29.95**
optional Vol. II Program Disks for Models I, II or III	**$49.95**

THE COMPUTRONICS INC.
50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

24 HOUR ORDER LINE
(914) 425-1535

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

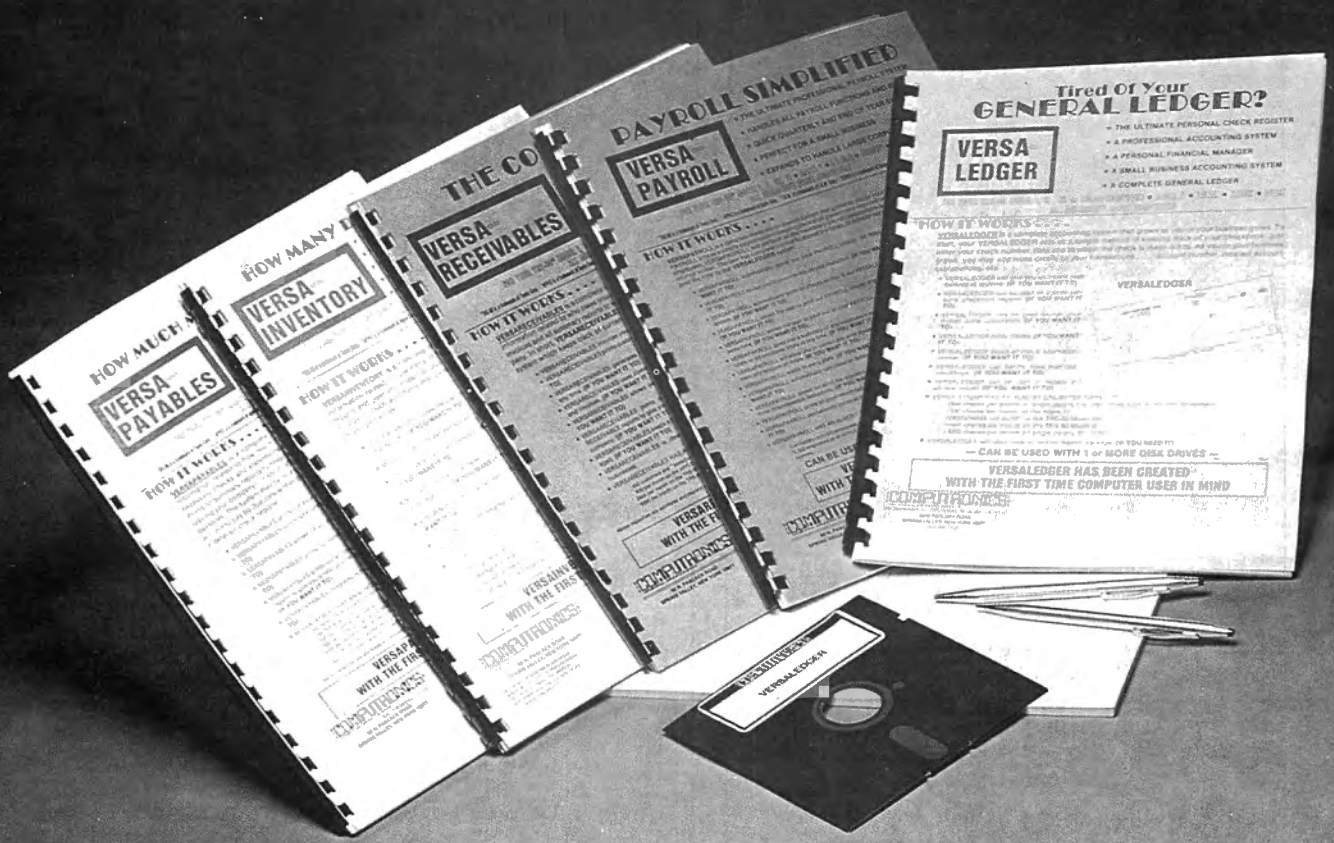
ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA AND MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO

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

— 30-DAY MONEY BACK GUARANTEE —

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 UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

VISA

DEALER INQUIRIES WELCOME

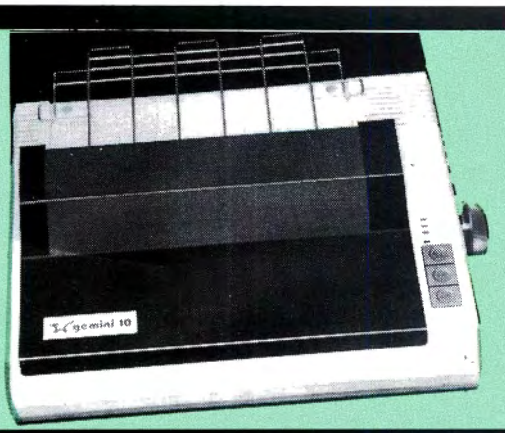
All prices and specifications subject to change / Delivery subject to availability.

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.

ADVERTISING DIRECTORY



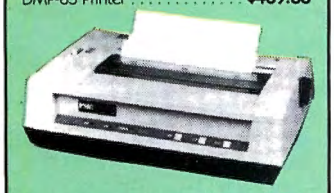
Star Micronics **GEMINI-10**

$419.88 UPS DELIVERED

- 100 characters per second, bi-directional, logic-seeking printhead action (48 lines/min.) with 2K print buffer, expanable to 4K on-board
- 9 x 9 matrix produces proportional, 10, 12, 17 cpi with true descenders, double width, double strike, italics, & special graphics characters
- 120 x 144 hi-resolution dot-addressable graphics matrix
- Subscripts, superscripts, underlining, backspace, plus 2K user-programmable character ROM, perf skip, vert/horz tabs
- Friction/tractor standard; handles 3-part forms (8.5")

PRINTERS

Anadex DP-9500A	$1459.88
Anadex DP-9501A	$1459.88
Anadex DP-9629A	$1549.88
Centronics 122-1	$829.88
Centronics 122-3	$949.88
Centronics 352	$1649.88
Centronics 353	$2324.88
C. Itoh Prowriter	$499.88
w/RS-232C	$609.88
C. Itoh Prowriter 2	$734.88
w/RS-232C	$789.88
C.Itoh F-10 Starwriter, 40 cps Parallel or RS-232C	$1499.88
C.Itoh F-10 Printmaster, 55 cps Parallel or RS-232C	$1799.88
F-10 Tractor	$289.88
Daisywriter 2000	$1089.88
Daisywriter Tractor	$149.88
Daisywriter Cable	$49.88
Diablo 620	$1269.88
Diablo 630	$1969.88
Diablo 630 KSR	$2694.88
630 Tractor	$314.88
DMP-85 Printer	$469.88



The "generic" version of the NEC & Prowriter. Features 120 cps, bi-directional, logic-seeking print action. 9 pin printhead produces 4 fonts (including proportional), 8 sizes, subscripts, superscripts, underlining & Greek/math fonts. Dot-addressable graphics matrix is 160 x 144 dpi, with 1/144" line feed. Friction & tractor feed is standard (takes 9½" width paper).

IDS Microprism	$679.88
IDS Prism 80	$1104.88
Prism 80 w/graphics	$1339.88
Prism 80 w/sheetfeed	$1459.88
Prism 80 w/4-color	$1539.88
IDS Prism 132	$1269.88
Prism 132 w/graphics	$1339.88
Prism 132 w/sheetfeed	$1459.88
Prism 132 w/4-color	$1699.88

PRINTERS

Microline 80	$349.88
Microline 82A	$439.88
80/82A Tractor	$59.88
82A Roll Paper Holder	$49.88
Microline 83A	$694.88
82A/83A Okigraph ROM	$44.88
Microline 84 w/graphics & tractor parallel, 200 cps	$1044.88
RS-232C, 200 cps	$1164.88
NEC PC-8023A	$509.88
NEC 3510	$1929.88
NEC 3530	$1809.88
NEC 3550	$2199.88
3500 Tractor	$239.88
Smith Corona TP-1 10 or 12 cpi, parallel or RS-232C	$599.88

CALL FOR PRICES on Epson, DIP, MPI, Datasouth, & other printers.

TRS-80 HARDWARE

TEC Drives	$239.88
Hayes Chronograph	$214.88
Model I/III cable	$29.88
Model II/16 cable	$39.88
TRS-80 Color cable	$29.88
TRS-80 Color RS-232C-to-Parallel Converter & cable	$99.88

MODEMS

The **LYNX** TRS-80 direct-connect modem features auto-dial, auto-answer,

MODEMS

0-300 baud transmission. Will work without RS-232C interface on Model IIIs.
Comes complete with hardware.

LYNX TRS-80 Modem $229.88

Hayes Smartmodems feature programability in any language, auto dial & auto answer, full & half duplex, keyboard control, user-selectable prameters, 7 LED status lights & an audio monitor. Comes complete with power supply and modular telephone cable.

300 Baud	$239.88
1200 Baud	$569.88
Signalman Mark 1	$89.88

TRS-80 SOFTWARE

MICROTERM

A terminal program from Micro-Systems Software, makers of DOSPLUS. Microterm supports the features on auto-dial & answer modems like the Lynx & Smartmodem. Features pre-programmed dial & transmit, direct file transfer, 34K capture buffer, and high operational baud rate (near 9600). Specify Model I or Model III when ordering

Microterm $79.88

DOSPLUS 3.4

The preferred disk operating system for Model I or III. Features BASIC array sort, (multi-key, multi-array), active "DO", device routing, DOS command repeat, etc. Exceptional. Specify Model I or III, single or double track, 40 or 80 track drive

DOSPLUS 3.4 ..... $119.88

TRS-80 SOFTWARE

NEWSCRIPT 7.0

Prosoft's NEWSCRIPT is the best word processing software available for the TRS-80 Model I or III. NEWSCRIPT supports the features of most popular printers, including NEC, C. Itoh, Okidata, Epson, Centronics, etc.

NEWSCRIPT features true proportional spacing, single and double width type, subscripts, superscripts, underlining, boldface, multiple pitches, full-screen editing, global search & replace, and customized "form" letters that also will create a MAIL LABEL (see below).

A typeahead/printahead buffer maximizes printer & computer speed. Plain English commands simplify operation and editing. NEWSCRIPT comes with a complete manual, including many applications, and support from the authors. Specify Model I or III when ordering.

NEWSCRIPT $109.99

MAIL LABEL OPTION

Create mailing labels from NEWSCRIPT files. Format 2 up, 3 up, etc. for use on envelopes, packages. Not a mail list program with sort options. Available only with purchase of NEWSCRIPT (not as an upgrade).

Mailing Label Option $15.00

SCRIPSIT/NEWSCRIPT CONVERSION

Takes either Electric Pencil or Scripsit files and converts them to NEWSCRIPT files. Upgrade your older library of files with this handy program.

Scriptsit Conversion $15.00

ELECTRIC WEBSTER

A 50,000 word dictionary for NEWSCRIPT. It can be selected from the main menu, used, then returns you to main menu. The Electric Webster features spell checking, options on change, & a "browse" feature allowing you to choose spellings or to enter your own. Fully compatible with NEWSCRIPT.

Electric Webster $134.88

Information & Orders
(603)-673-8857

Orders Only: (800)-343-0726

NO HIDDEN CHARGES

FREE UPS shipping on all orders—No extra charge to use credit cards—All equipment shipped factory fresh with manufacturer's warranty—COD orders accepted ($10 fee added)—No purchase orders accepted—No foreign or APO orders accepted—Minimum $50 per order—This ad prepared in November; prices subject to change.

HIGH TECHNOLOGY AT AFFORDABLE PRICES

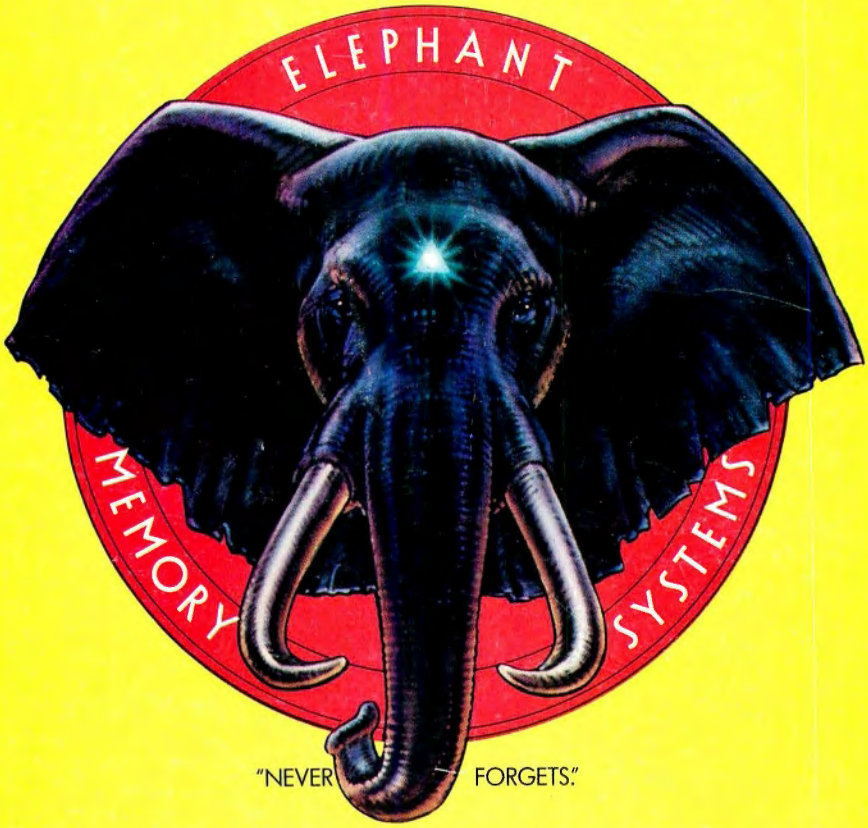
MasterCard VISA

THE BOTTOM LINE

MEMBER

MILFORD NH 03055-0423

REMEMBER:



MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to make life miserable for everyone in the disk-making business.

How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

ELEPHANT.™ HEAVY DUTY DISKS.

For a free poster-size portrait of our powerful pachyderm, please write us.
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.