Software Giveaway with Purchase of this Magazine

See bound-in card behind front cover

HOME COMPUTER™ magazine

FOCUSING EXCLUSIVELY ON ● APPLE ● COMMODORE ● IBM ● TEXAS INSTRUMENTS

Vol.4 No.1

$3.00 in USA
$3.50 in Canada

PCjr: How to Buy, Operate, and Program the Peanut

Spritely Tricks to Pep Up Commodore Programs

Creating a Real Data Base on the Unexpanded VIC-20

Apple 3-D Graphics Made Easy

Super Sound & Music on the TI-99/4A

- LOGO Poetry 'n Motion
- A Bonanza of Key-In-and-Run Software for Apple II, C-64, VIC-20, IBM PCjr and TI-99/4A
- Computer Assisted Savings • Game Reviews Galore
- Word Processing at Home

THE SECRETS OF PERFECT MEMORY: ONE AND ONE HALF EARTH DOLLARS

AT LAST: THE WHOLE TRUTH ABOUT FLOPPIES.

Amazing book reveals all!

How to keep from brainwashing your disk so it never loses it's memory.

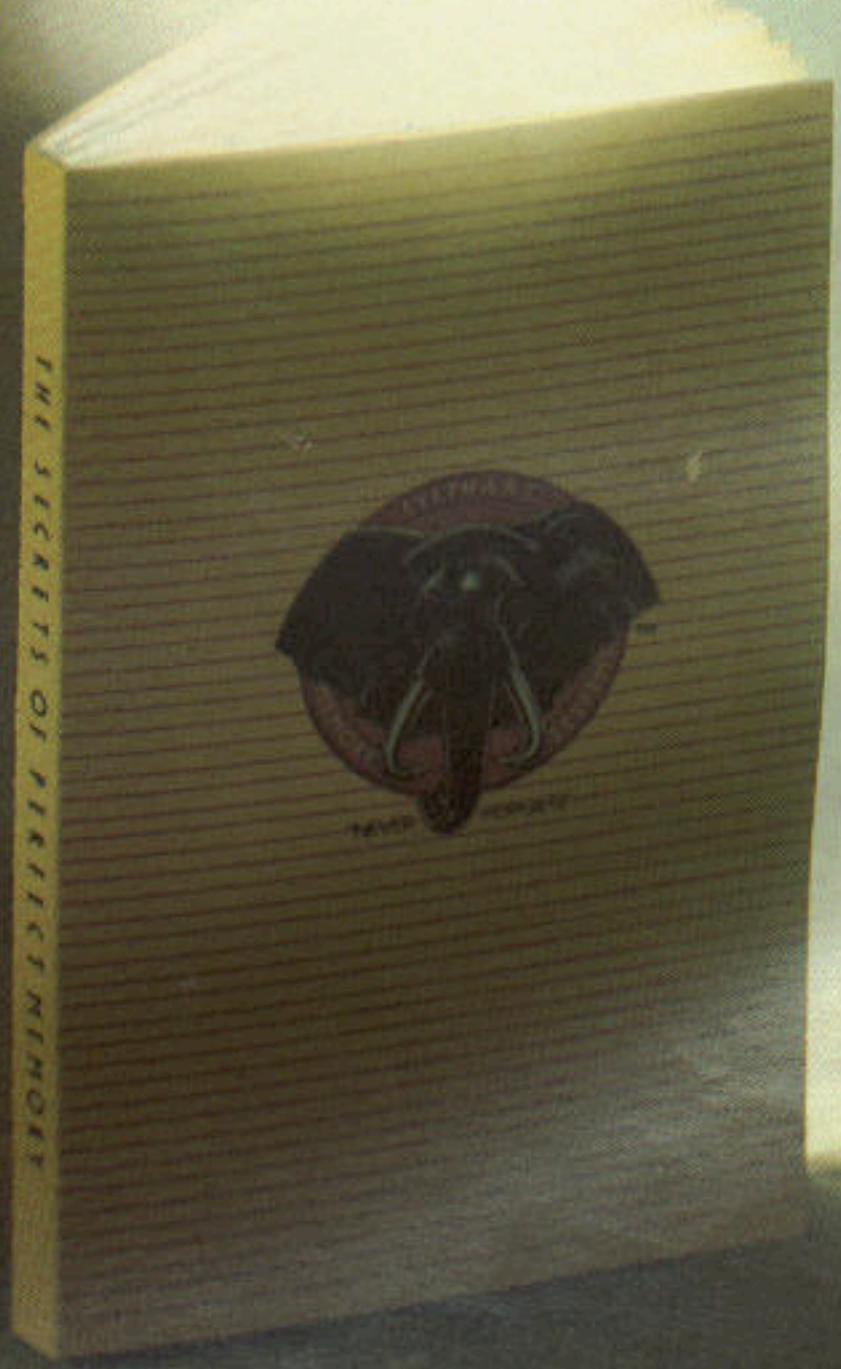
How fingerprints can actually damage disks. Unretouched Kirlian photographs of UFO's (Unidentified Floppy Objects)! The incredible importance of making copies: the Department of Redundancy Department — and what goes on when it goes on! Powerful secret methods that scientists claim can actually prevent computer amnesia! All this, and much more . . .

In short, it's an 80-page plain-English, graphically stunning, pocket-sized definitive guide to the care and feeding of flexible disks.

For The Book, ask your nearest computer store that sells Elephant™ disks, and bring along one and one half earth dollars.

For the name of the store, ask us.

ELEPHANT MEMORY SYSTEMS®. Marketed exclusively by Leading Edge Products, Inc., Information Systems and Supplies Division, 55 Providence Highway, Norwood, MA 02062. Call toll free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150, Telex 951-624.



HOME COMPUTER MAGAZINE'S SENSATIONAL SOFTWARE GIVEAWAY

YOUR CHOICE OF THE PROGRAMS FROM THIS ISSUE AT THE UNBELIEVABLE LOW PRICE OF ONLY $3.95!

PROOF OF PURCHASE NECESSARY

To participate in our monthly Software Giveaway, you need to be a bona-fide purchaser of *Home Computer Magazine* and follow the directions on this form.

PLEASE PRINT

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

Have you taken advantage of our Software Giveaway before? ☐ Yes ☐ No

Indicate your choice of media: ☐ ON TAPE™ ☐ ON DISK™

Indicate which machine the media is for: (Pick one)
☐ Apple II Series ☐ C-64 ☐ VIC-20 ☐ IBM PC ☐ PCjr ☐ TI-99/4A

Fill in this magazine's *Volume No.* ____ and *Issue No.* ____. You will receive all programs (which have versions for your selected machine) whose listings appear in this issue. Please also list, in order of preference, these same programs—helping us determine future offerings.

1. _____ 4. _____
2. _____ 5. _____
3. _____ 6. _____

IMPORTANT: Complete the questionnaire on the reverse side. Tear out this complete page and enclose in an envelope along with $3.95. Payment must be made by check, money order or VISA/MasterCard. Proof of purchase (subscriber label/number, sales receipt or reasonable facsimile thereof) must also accompany this form.

☐ Questionnaire is completed.
☐ Enclosed is check or money order for $3.95.
☐ Enclosed is my proof of purchase.
☐ Bill my ☐ VISA ☐ MasterCard

Account No.

Signature _____ Exp. Date _____

OFFER GOOD ONLY IN USA
(Canada & Mexico please inquire)

Mail this page to:
Software Giveaway
Home Computer Magazine
P.O. Box 5537
Eugene, OR 97405

TO PROPERLY HANDLE THE VOLUME OF SOFTWARE REQUESTS, ORDERS MUST BE SENT IN ON THIS FORM, NO PHOTO COPIES. SORRY, WE CANNOT ACCEPT TELEPHONE ORDERS FOR THIS SERVICE.

SG/2-84

ON TAPE / ON DISK—$3.95 each issue.

ABSOLUTELY FREE!
2-6 MONTHS OF MAGAZINE SOFTWARE ON TAPE™ OR ON DISK™ ARE FREE WITH YOUR PAID SUBSCRIPTION

With a 1-year subscription to *Home Computer Magazine* you get 2 FREE months of ON TAPE or ON DISK. Subscribe for 2 years and receive 4 months. And with a 3-year subscription we'll give you 6 full months of this convenient software on cassette tape or floppy disk.

NEW SUBSCRIBER OFFER
(Valid in USA only)

PLEASE PRINT

Name _____

Address _____

City _____ State _____ Zip _____

☐ Check or Money Order Enclosed
MUST BE US FUNDS DRAWN ON A US BANK

Bill my ☐ VISA ☐ MasterCard

Account No.

Signature _____ Exp. Date _____

SATISFACTION GUARANTEED—or the unfilled portion of your subscription will be refunded, less the cost of any premium items you have received

Indicate your choice of media: ☐ ON TAPE ☐ ON DISK
Indicate which computer media is for: (Pick one)
☐ Apple II Series ☐ C-64 ☐ VIC-20 ☐ IBM PC ☐ PCjr ☐ TI-99/4A
Choose desired subscription term:
1-yr ☐ $25—12 issues + 2 months ON TAPE / ON DISK
2-yr ☐ $45—24 issues + 4 months ON TAPE / ON DISK
3-yr ☐ $63—36 issues + 6 months ON TAPE / ON DISK

Enclose payment or credit card information and mail with completed form to:

Home Computer Magazine
P.O. Box 5537 • Eugene, OR 97405

Or use our TOLL-FREE ORDER LINE*
FOR VISA/MasterCard orders only: 1-800-828-2212
In Oregon, Alaska, Hawaii Tel. (503)485-8796
*Please make reference to this ad when placing phone orders.
Allow 6-8 weeks for your first issue.

SGSO/2-84

HOME COMPUTER™ magazine

QUESTIONNAIRE

Complete and mail to: Home Computer Magazine • P.O. Box 5537 • Eugene, Oregon 97405

FOR ALL READERS

1. **Where did you obtain this copy of Home Computer Magazine?** ☐Subscriber ☐Supermarket ☐Bookstore ☐Users group ☐Newsstand ☐Computer Store ☐Friend ☐Library ☐Other ___
2. **What types of software are you most interested in?** ☐Educational ☐Entertainment ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other ___
3. **Are you** ☐Male ☐Female ☐14 or younger ☐15-24 ☐25-34 ☐35-44 ☐45-54 ☐55+
4. **Annual Household Income?** ☐Under $10,000 ☐$10,000-$14,999 ☐$15,000-$19,999 ☐$20,000-$24,999 ☐$25,000-$29,999 ☐$30,000-$39,999 ☐$40,000-$49,999 ☐$50,000+
5. **Occupation?** ☐Professional ☐Management ☐Teacher ☐Student ☐Other ___
6. **What is your ZIP code?** ☐☐☐☐☐
7. **What is the current month and year?** ___ ___
8. **Do you presently own a Home Computer?** ☐No ☐Yes. It is a ☐TI-99/4A ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐Other ___

FOR READERS WHO PLAN TO BUY A HOME COMPUTER

9. **Which model do you think you'll purchase?** ☐Apple IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other
10. **When do you expect that purchase to be?** ☐less than 3 months ☐3-6 months ☐7-12 months ☐at least 1 year
11. **What do you anticipate your primary use of a home computer will be?** ☐Entertainment ☐Education ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other ___

FOR PRESENT HOME COMPUTER USERS

12. **Which home computer(s) do you currently own?** ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other ___
13. **What is the primary use of your home computer?** ☐Entertainment ☐Education ☐Computer Literacy ☐Business ☐Job-Related Applications ☐Household Management ☐Other ___
14. **How often is your computer in use?** ☐Less than 1 hour per week ☐1-4 hours ☐5-10 hours ☐11-15 hours ☐16-20 hours ☐over 20 hours
15. **On the average, about how many program listings in each issue of HCM do you key into your computer and use?** ☐None ☐1 ☐2 or 3 ☐4 or more
16. **What peripherals do you currently use?** ☐Disk System ☐Printer ☐Modem ☐Monochrome/Color Monitor ☐Other ___
17. **What do you expect to buy within the next year?** ☐Software ☐Disk system ☐Printer ☐Modem ☐Books ☐Magnetic Media ☐Monochrome/Color Monitor ☐Furniture & Accessories
18. **How much do you expect to spend on computer-related products during the next year?** ☐Less than $25 ☐$25-$49 ☐$50-$99 ☐$100-$249 ☐$250-$490 ☐$500-$999 ☐$1000-$2499 ☐$2500 or more

OPTIONAL: If you would like to help us by participating in a telephone interview, please include your telephone number (___) ___-____ here and the most convenient time you can be reached ___:___ AM PM

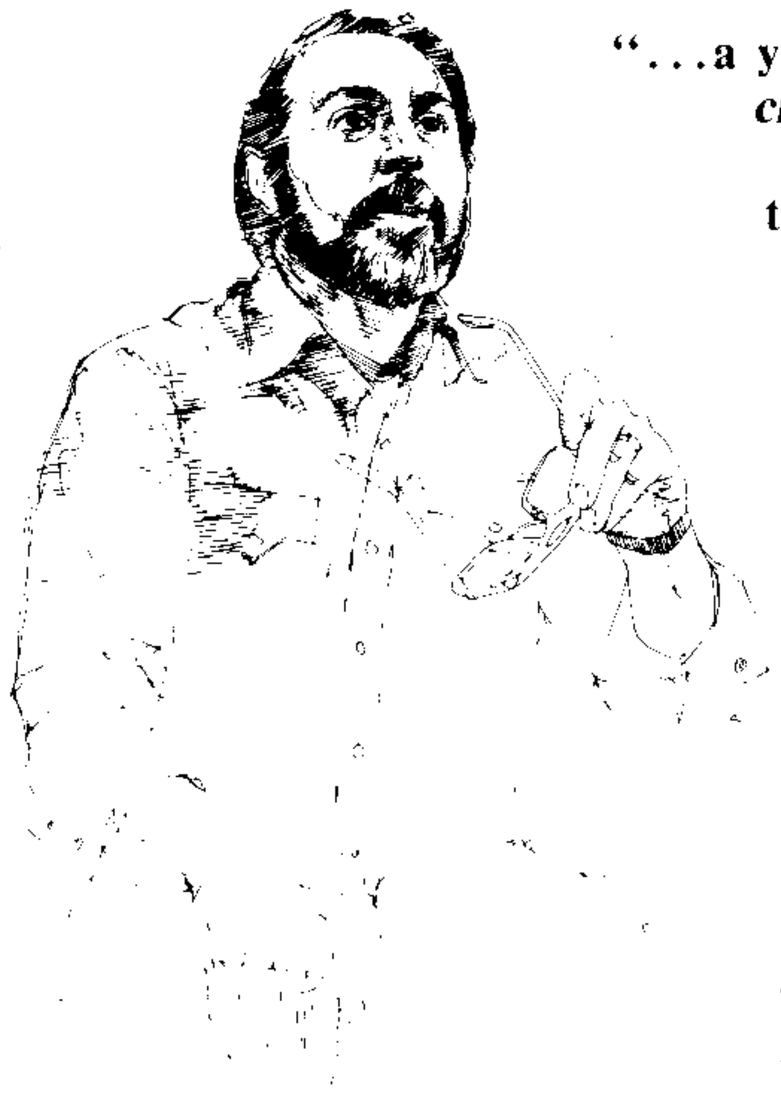
TIGERVISION®

Tiger Electronic Toys, Inc.
909 Orchard
Mundelein, IL 60060
(312) 949-8100

On Screen

By Gary M. Kaplan
Publisher & Editor-in-Chief

It was the best of times;
It was the worst of times.
—Charles Dickens, *A Tale of Two Cities*



"...a young, volatile world in which *change* is the only constant... an unforgiving world where the life and death of products depend on time-critical windows of opportunity."

It was the worst of times for Texas Instruments to exit the consumer computer marketplace: Their Home Computer was, at last, positioned correctly; an extremely visible, no-nonsense network TV campaign presaged strong holiday sales; dozens of the industry's finest software titles had been "cherry-picked," converted, and scheduled for pre-Christmas release; and a new generation of powerful, upwardly compatible hardware was about to be launched.

Ironically, it was also the best of times —that is, the best of times for hundreds of thousands of holiday shoppers looking for quality, affordable gifts or virtually no-risk tickets of entry to the intriguing world of home computing.

In the young, volatile world of the home computer industry, *change* is the only constant. It's an unforgiving world where the life and death of products depend on time-critical windows of opportunity—a place where even the smallest slip can prove fatal to the weak, and where a mountain of silica possessed by the strong must eventually drain through the corporate hour-glass of fiscal responsibility and stockholder impatience.

In such a dynamic environment, the worst thing any participant can do is stand still. Computer publications are no exception. A user magazine must be flexible enough to accommodate new reader interests, and perceptive enough to know when to alter its course.

This magazine is now on an exciting new heading. What started as *99'er Magazine* nearly four years ago—and later became *99'er Home Computer Magazine*—is now, simply, *Home Computer Magazine*. Our heritage is reflected in the name change: We pioneered the concept of a magazine exclusively for *home* users of computers, and nurtured it to become the world's leading machine-specific computer magazine. Now, we have broadened our TI-exclusive beat to include the three brands which we feel will be most in demand for home use during the remainder of the 80's: Apple, Commodore, and IBM. And as evidenced by this issue, we haven't sacrificed one bit of the quantity, quality, or comprehensiveness of our traditional 99/4A-related content.

Over the last several years, a close rapport and mutual respect has developed between us and our readers. We've listened to you, and learned. Today, with millions of published words behind us, the magazine's editorial, technical, and design teams have the requisite knowledge and sensitivity to make *Home Computer Magazine* the most valuable information resource—bar none—for present and future home users.

I dare to make this claim—and the editorial promise to keep it valid—because of the care and planning that go into each issue. Crafting a quality magazine with enduring reference value is our life's work; it is far removed from the prevalent practice of belching out prodigious masses of paper that are thrown together as "catalogs" *solely* for the purpose of carrying advertising messages for a span of thirty days.

We absolutely have no room within our pages for editorial "filler." The fat stays out. Instead, you get a well-balanced diet of facts—not fluff! You'll find carefully written and illustrated articles that are interrelated to, and integrated with other material in the same issue. Our goal is to produce a well-orchestrated learning environment that encourages self-paced exploration, and is, above all, *fun*.

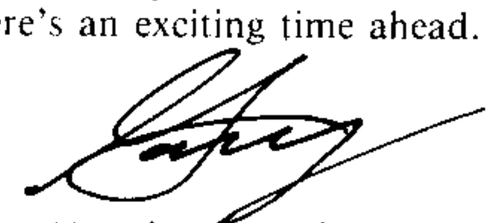
Then, of course, there is our software. Although we do have more key-in-and-run programs than anyone else, it is the unique way we present program listings—for maximum clarity and minimum typing errors—that has made us famous.

Even though many thousands of readers have indicated that they both enjoy and learn from the type-in-and-save process, we know that there are many more of you who don't have the time for it. *Home Computer Magazine* has a practical solution: You may order all of an issue's programs for your particular computer ON TAPE™ or ON DISK™ for an unbelievably low price—less than $4.

With our exclusive, in-depth coverage of Apple, Commodore, IBM and Texas Instruments, each issue of *Home Computer Magazine* is really *four* system-specific magazines in one wrapper—harnessing the power of "synergy" to make the value of the *whole* greater than the sum of its component parts. This unique approach allows readers to better understand their own computers through comparison to other machines. The contents of each issue is carefully balanced to deliver the proverbial "something for everyone"—translated for users of all four computer brands —whether absolute beginner or seasoned pro.

For our old as well as many new readers, I expect the coming years to be the best of times—the best of times to own a computer, and to learn, play, experience, and explore the magical world of home computing with your one essential resource, *Home Computer Magazine*.

So stay with us; there's an exciting time ahead.



Group Grapevine

News, information and upcoming events of home computer users groups around the world.

Looking to join a users group, exchange newsletters or software, increase your users group's membership or pep up your next meeting's agenda? For the latest users group news, put your ear to the Group Grapevine. And if you have a message to put out to other groups, if you are starting a new group, or have an interesting item to share, send a note or picture—or better yet, a group newsletter—to the Users Group Editor, Home Computer Magazine, 1500 Valley Drive, Suite 250, Eugene, OR 97401, (503) 485-8796

The phone lines and electronic bulletin boards have been buzzing as users groups around the world speculate, compare notes, and try to get more information on the fate of their machines. Groups responded to TI's announcement in various ways. Some called emergency meetings, others manned TI product availability hotlines. ("Ten consoles at $49.95 spotted at J.C. Penney. Unconfirmed report of peripheral expansion box at K-Mart.") Others went out into the streets to offer assistance to the droves of potential new TI-owners eager to get in on the "deal of the century." Everyone we talked to, however, agreed upon one thing: Now, more than ever, users groups will play a major role in user support.

Right now, the main quest is for information. To fill that need, Don Veith of TEX-BUG is forming an organization of users group presidents in the Western U.S., Western Canada, and Australia. The **Western Users Group Association (WRUGA)** will serve as a clearinghouse for information—to stop some of those wild rumors that are circulating, provide information on third-party releases, and more. To join the group, contact Don at 3535 So. H St. #93, Bakersfield, CA 93304. Hopefully, an enterprising easterner will start a similar group for the eastern half of the TI world.

Many of the new user groups starting up are hungry for basic information—how to go about starting up a group, writing by-laws, contacting prospective members, etc. There is good news for you brand new groups: TI's excellent start-up packet is still available. You can receive it by calling the toll-free TI CARES number (800) 842-2737. If you can't get through at first, don't despair...remember, lots of people are phoning that line.

Meanwhile, our letters and calls tell us that record numbers of TI users are looking for groups to join, and starting up new groups of their own. The influx of new owners eager for basic information and old-timers hungry for product news means more and bigger users groups all over the world.

Among the new groups we have heard from is the **Nittany Users of TI (NUTI)** in Pennsylvania. They are starting the first TI organization in what was, heretofore, exclusively Apple country. Interested parties in that neck of the woods can contact Linda Becker at 625 Berkshire Dr., State College, PA. A group has also sprung up in the streets of El Paso. To get in touch with the **Sun City 99/4A Computer Club of El Paso**, contact their president, Michael Elsner, in care of the Sun City Computer Club, P.O. Box 6966, El Paso, TX 79906.

The Brevard Users Group a new group located near the Kennedy Space Center, sports a space shuttle logo on their newsletter. The club owns three TI systems, a software library of 100+ programs, and promises programming classes, stimulating guest speakers, low dues, group software/hardware discounts, and a good time. So, for a good time, write B.U.G., P.O. Box 1402, Palm Bay, FL 32906.

Anyone who has ever felt himself at the mercy of a capricious, all-powerful Home Computer will appreciate the **Salt Lake and Valley 99'er Users Group's** acronym. Members of SLaVe (who refer to themselves as SLaVes) sent us a copy of their newsletter this month. Among the interesting tips inside was a subroutine to let those of you with Speech Synthesizers and Terminal Emulators list your programs with speech. This sounds great for double-checking those long, type-in-and-run listings without getting terminal eyestrain. Potential SLaVes can contact the group through Brent Case at 1874 West Homestead Farms #4, West Valley, UT 84119, or (801) 973-8480.

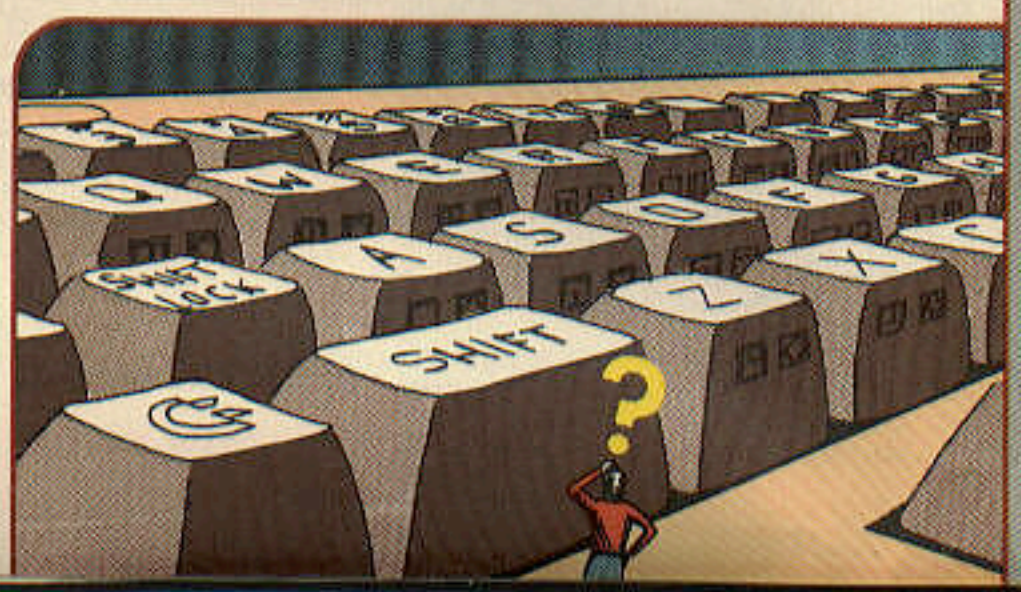
The Holidays have meant increased activity for Commodore groups. What with eager shoppers and holiday sales, it seemed the perfect time to exchange an old piece of software or machine accessory for that shiny new peripheral you've been coveting. The **Commodore Houston Users Group** (CHUG) had a bright idea for some holiday shopping and swapping. They held a swap meet to let their members buy, sell, or trade software and hardware. Sounds like a painless way to cut down on that last minute desperation shopping, trade in that well-meant (but all wrong) birthday present, and start the new year with some inspiring new software. CHUG has been in existence for about two years and boasts about 500 members (we didn't ask how many show up for meetings). According to group spokesman, John Walker, they are an informal group who prefer talking with one another to bringing in outside speakers. They have held BASIC, hardware, and Tech Talk clinics to focus on topics of special interest. You can contact this group at 8738 Wildforest, Houston, TX 77088, (713) 999-3650.

The **VIC Commodore User Club** in Mariesville, MI, was pleased to receive a $50 donation from the Mariesville Police Department at their last meeting. The reason? The group recently developed a program to help their men in blue keep track of scores and divisions at a police target shooting meet. An ultra-quick sort subroutine was used to provide instant information on past and present shooting scores. The Mariesville group is one of that rare breed of hybrid organizations—users groups who welcome any

HOME COMPUTER™ magazine

FEATURES

LOGO TIMES™



member regardless of machine type. Acording to M. Gautier, one of their more active members, the multi-machine mix works out surprisingly well. You can reach this polyglot group at 486 Michigan Ave. Mariesville, MI 48040, (313) 364-6804.

The mammoth **Toronto Commodore User Group** is already making plans for their annual conference, slated for May 18. For the few Commodore people who have never heard of this 10,000 member group, we should mention their gargantuan software library (containing some 5000-6000 programs) which any group or individual can join. Library Members receive monthly diskettes of public domain software. Anyone interested in joining either the library or group can contact Chris Bennet at 381 Lawrence Ave. West, Toronto, Ontario, Canada, M5M 1B9, (416) 782-9252.

Less huge, but equally active groups have also been busy this month. The **Napa Valley Commodore Computer Club** got together to take a look at the VIC Tree cartridge and Easy Script Word Processing Package . You can contact this group at 2680 Jefferson St. Napa, CA 94558, (707) 252-6281. In Virginia, the **Peninsula Commodore 64 Users Group** were treated to an impressive graphics demonstration. A commercial artist showed how the Doodle program could be used with a track ball to move lines on the screen in a manner similar to a light pen or artist's brush. According to Richard Wilmoth, the combination of artistic talent and technical innovation made for some dazzling displays. For more information on the group, contact him at 124 Burnham Place, Newport News, VA 23606, (804) 595-7315.

One of our newer groups, the 6-month-old **Commodore User Group** in Santa Cruz, CA, met this month to watch a local computer dealer demonstrate how to clean your disk drive. Nothing like bringing in an expert to get the inside track on proper maintenance. You can contact the group through Bud Massey at 2301 Mission St., Santa Cruz, CA 95060, (408) 425-8054.

It may not seem like much of a problem, but some of the larger Apple groups seem to be finding themselves with a surplus of money these days. When you think about it, a group of 1000 that charges $28 per year in dues is pulling in a healthy income of $28,000 yearly. Add to that the extra revenue from diskette sales, etc., and you can understand why these big Apple groups have something of a reverse cash flow problem. What to do? Of course, you can just let the money sit there or print a 4-color newsletter, but most groups prefer to put their money to use to benefit the members.

The **Dallas Apple Corps** is one of the more well-heeled groups that are considering some exciting new ways to put their cash to work. One radical new idea that has been suggested is paying the officers a mini-salary. While some may balk at the idea, consider all the time your program chairman puts into planning each meeting, wheedling people to make presentations and lining up speakers. And what about the poor soul in charge of the newsletter—when was the last time you sent in some unsolicited copy to help him out? A small salary would be a just reward for these hard-working people who really keep the group going, and it would offer incentive to improve performance—chances are a lot more people would be interested in a paid position. . . and members could expect first-rate performance from their newly commisioned officers.

Another option for the wealthy group is paying to bring in speakers—why not spring for a plane ticket and bring in Steve Wozniak to explain, once and for all, how to edit program lines? Or how about a more concrete investment? The Dallas Apple Corps has already put some of their surplus cash to good use by purchasing some equipment—2 monitors and a 10-foot diagonal projector. These lone-star Apple barons are as active as they are prosperous: They have special interest groups on business and personal software, spreadsheets, advanced topics, the C language, advanced Applesoft, beginning on the Apple, Pascal, CP/M, education, and science & technology. At their last meeting, the Dallas Apple Corps people enjoyed a presentation on a 68000 board (with a quarter megabyte) for the Apple. You can reach this dynamic group in care of Apple Corps of Dallas, P.O. Box 5537, Richardson, TX 75080.

Education is a strong interest for the **Birmingham Apple Corps** members. It was such an important topic for them that they started a spin-off group, **Apple Blossom**, which is exclusively for Apple-oriented educators. That little offshoot now numbers over 100 members. The original group still retains an educational bent, however, providing regular classes in BASIC and assembly language. At their last meeting, the group discussed the Z-80 chip and listened to a representative from 3M discuss the properties of different disks. You can reach the Birmingham Apple Corps at P.O. Box 55421, Birmingham, AL 35255.

The **Mini'App'les** (in Minneapolis, of course) were treated to a presentation on artificial intelligence by two professors from the University of Minnesota. This large conglomerate of small Mineapolis groups includes special interest groups on Pascal, business applications, investment applications, education, medical applications, Visi-Calc, CP/M, telecommunications, and applications for the handicapped. You can contact the group by writing to Mini'App'les, P.O. Box 796, Hopkins, MN 55343.

jr

Users groups for the PCjr? Not yet, but if the **IBM PC Users Group** in Eugene, Oregon is any indication, the PCjr will find a warm spot waiting in the hearts of PC groups across the country. At their last meeting, Rocky Mertin, a representative from IBM, showed slides and answered questions about this long-awaited addition to the IBM family. One thing is for sure, the interest in this new machine is high, and all signs point to a rash of "extended family" PC groups, spin-offs, and brand new groups exclusively for the new PCjr home user. We hope to hear from these new groups as soon as they spring up—and we'll be closely covering their rapid progress in each monthly installment of Group Grapevine.

HCM

Your Guide to Typing in Programs from *HCM*

Within these pages is a software bonanza: entertainment, education, home and business applications, utilities, and tutorials—just for you. All you need to do is type them into your computer. *HCM* has taken most of the strain out of this process with:

- Typeset listings with numbers in boldface type.
- A bold, double vertical bar separating the line numbers from the program statements in BASIC listings.
- A vertical background grid to aid entry of the spaces.

Look at the Key-in Reference (Figure 1 below) to see how each character actually appears in the listing. By checking any questionable characters with the Key-in Reference, you can reduce errors to a minimum.

Figure 1: KEY-IN-REFERENCE

Before You Begin

Since *HCM* publishes programs for several different computers, the first thing you should do is make sure that you are looking at the listing designed for your machine. If, for example, you have an Apple IIe, make sure you look for the following black bar above the listing: APPLE II Series The computer model name will likewise appear on each subsequent page of each listing, so always look for the name before you begin typing from a new page of listings.

Before you begin typing in the program, you will want to set up a system to save your program. Whether you are using a cassette or diskette storage system, now is the time to be certain it is properly connected, powered up, and loaded with a blank cassette or an initialized disk. As you type in your program, you should get in the habit of saving your work after every twenty or so lines.

One of the most common errors in entering a listing is typing one symbol for another. These transpositions include substituting the letter O for the number 0, the letter I for the number 1, the letter S for the $, and the uppercase B for the number 8. This last error is especially likely when working in hexadecimal numbers which are composed of 0-9 and the uppercase letters A-F.

The listings in *HCM* are always the same number of characters wide, but, the number of characters put on any line of the video display will vary from computer to computer. Don't try to make your listing *look like* the type-set listing—instead make sure you key in the listings character for character and space for space.

A Special Note on Listings for the C-64 and VIC-20

Commodore uses more than 90 special symbols to represent various keyboard operations: for instance, the symbol [♥] in a program represents the operation of holding down the [SHIFT] key and pressing the key which has CLR on its upper half (second key from the right on the top row). This operation clears the screen.

Rather than reproducing these symbols, *HCM*'s listings include key-stroke instructions between two hands with pointing fingers. For example, when you find ☛ SHIFT CLR ☚ in an *HCM* listing, you will know to hold down the [SHIFT] key and press the key with CLR on it.

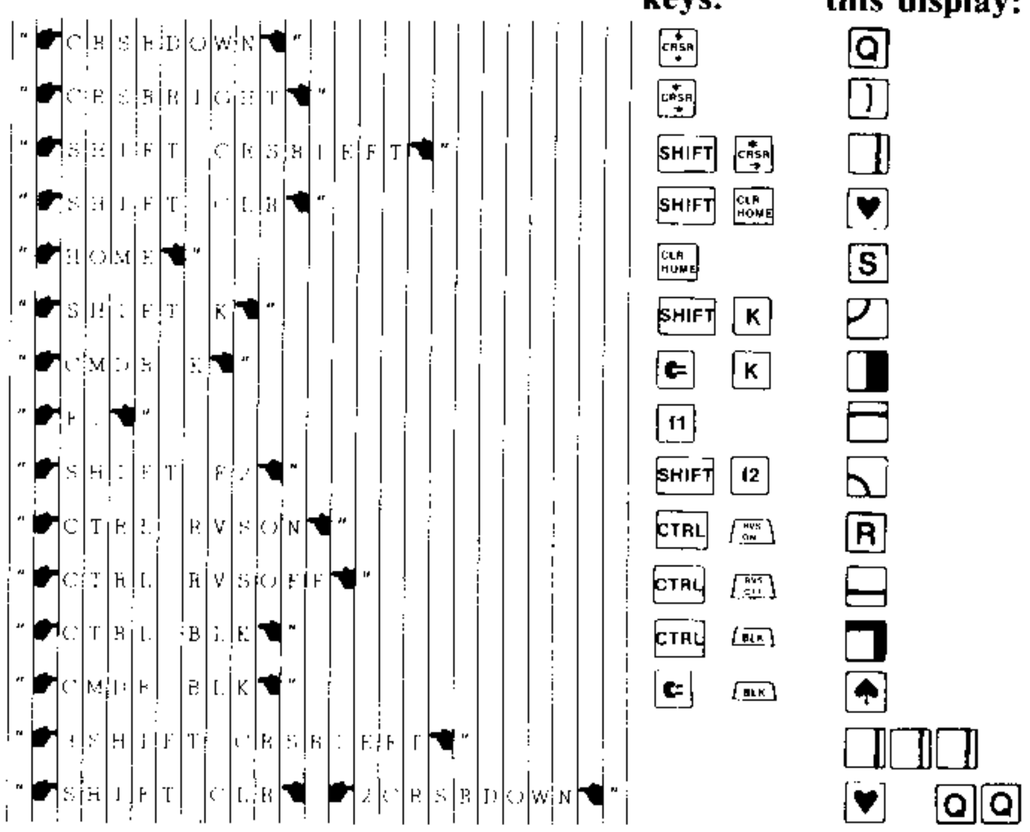
A number is included if you need to repeat the operation: ☛ 8SHIFT CRSRLEFT ☚ tells you to hold the [SHIFT] key down and press the cursor left key (on the bottom right of the keyboard) eight times.

When you come to the hand symbols, remember:

- Each operation is enclosed in its own set of hand symbols.
- If any key action requires you to press two keys, press the control key or the Commodore key or the shift key. *first* and hold it down before pressing the second key.
- Not to enter spaces or anything else *within* a set of hands.
- Everything between a pair of hand symbols is set in a different typeface.

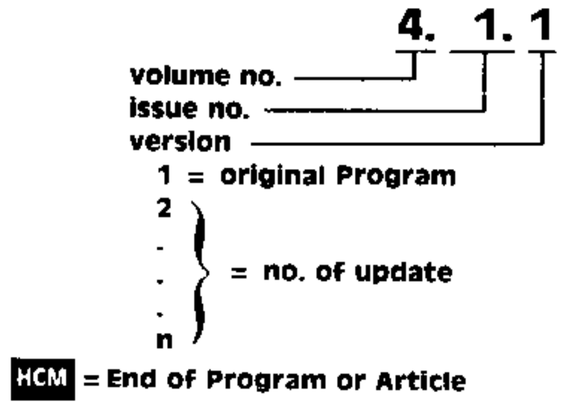
In Figure 2 below, we have included a chart showing you a representative sample of the symbols that appear when you use keystrokes enclosed by the hand symbols.

Figure 2: SPECIAL SYMBOLS CHART FOR C-64 and VIC 20



Program Identification

Each program header (the first few lines of the program) contains information giving the language the program is written in (e.g., TI Extended BASIC, Applesoft, etc.) and any special system components that are required (special memory cards, Speech Synthesizer, etc.). The first two digits of the version number tell you in which volume and issue of *HCM* the program *initially* appeared. The third digit of the version number indicates the version of the program. When a program initially appears, in *HCM*, it is version 1. Any subsequent revisions to the program if later published in the magazine or in the software available on magnetic medium from *HCM* will bear a revised version number.



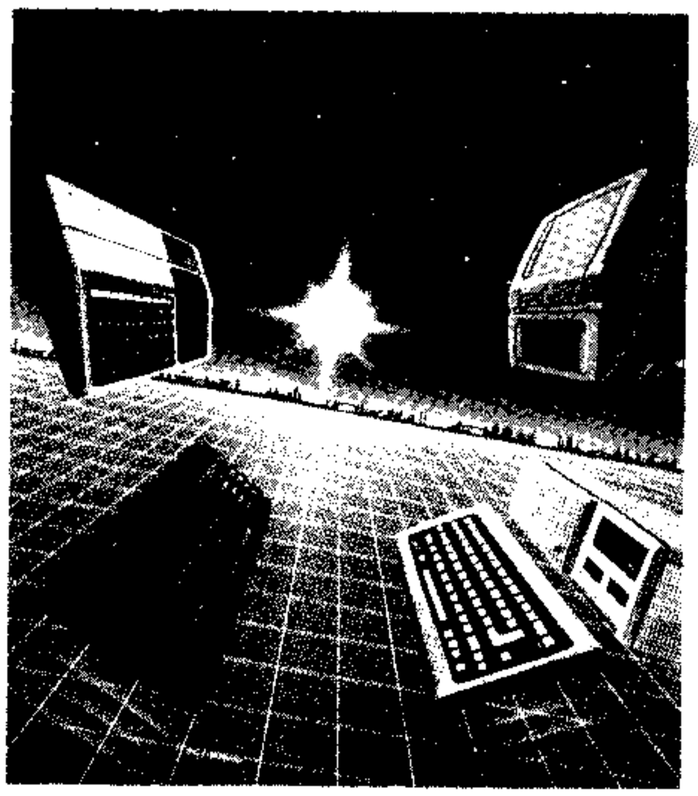
HCM

HOME COMPUTER Magazine

Home Computer Magazine is published monthly by Emerald Valley Publishing Co., P.O. Box 5537, Eugene, OR 97405. The editorial office is located at 1500 Valley River Drive, Suite 250, Eugene, OR 97401. (Tel. 503-485-8796). Subscription rates in U.S. and its possessions are $25 for one year, $45 for two years, and $63 for three years. In Canada and Mexico add $7 per year. Other foreign countries $43 for one year surface mail. Inquire for air delivery. Single copy price in U.S. and its possessions is $3.00, and $3.50 in Canada and Mexico. Foreign subscription payment should be in United States funds drawn on a U.S. bank. Application to mail at second-class postage rates is pending at Eugene, OR 97401.

POSTMASTER: Send all address changes to **Home Computer Magazine**, P. O. Box 5537, Eugene, OR 97405. Subscribers should send all correspondence about subscriptions to above address.

Address all editorial correspondence to the Editor at **Home Computer Magazine**, 1500 Valley River Drive, Suite 250, Eugene, OR 97401. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage and self-addressed envelope. Not responsible for lost manuscripts, photos, or program media. Opinions expressed by the authors are not necessarily those of **Home Computer Magazine**. All mail directed to the "Letters to the Editor" column will be treated as unconditionally assigned for publication, copyright purposes, and use in any other publication or brochure, and are subject to **Home Computer Magazine's** unrestricted right to edit and comment. **Home Computer Magazine** assumes no liability for errors in articles or advertisements. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by **Home Computer Magazine** or the publisher unless explicitly stated.



Outside HCM

The airbrush artistry of Eric Martin depicts our commitment to the future. A computer that begins humbly on a design grid may rise rapidly to the top in this dynamic industry. Four home computers stand out on the high-tech horizon. In *Home Computer Magazine* we chart their progress, reflect the evolution of the industry, and interpret how these new developments and applications will affect you, the reader.

Inside HCM

This month we bring you a home computer concert—written in four-part harmony. Center stage, the principal players are warming up for a performance you can enjoy in your own home. The program has been specially selected to expand your home computer repertoire.

To guide you through this issue, here are a few notes on a musical theme. *Uncle Larry's Fiddle Tunes* will have you toe-tappin' in time to the familiar strains of "Arkansas Traveler" and nine other fiddlin' favorites. Go ahead and hum along as you key in the complete BASIC program listing for your own computer.

With these country classics still ringing in your ears, head back (not Bach) in time, to discover Professor Holl's *Pocket Canon*, an electronic version of Pachelbel's Canon in D. And if you have a more current beat in mind, we have two programs for budding composers. *I Write the Songs*, in Extended BASIC, and *Just Assemble Melody*, in Assembly Language, let you write your own music and see it displayed on the screen. If you are still a bit intimidated by Assembly Language, be sure to check out part 3 of our tutorial, *Have No Fear: Assembly Language Won't Byte*.

Another language makes a command performance in *Lyrical LOGO*. Your computer can become a poet, lyricist, or just a total LOGO-phile. Or it can help you build skyscrapers in *What Is LOGO?* Then LOGO blasts off in *LOGO Shoots for the Moon*. The tips in our latest *TI-WRITER Tutorial* will help you record your lunar impressions with snappy word processing techniques to speed up and simplify the literary process.

Back on Earth, a symphony of applications awaits us. In *Porsches and Other Pipedreams*, you will find new ways to save money for that dream purchase—a Steinway, or a super stereo . . .

Making music isn't the only way to use notes and measures — menu planners use them too. In *Moveable Feasts*, you can plan a menu that makes a banquet of ordinary leftovers.

Apple devotees can get right to the core of the matter with three pieces written especially for them. *Biting Into Your Apple* is a theme and variations approach to home computing. Elaborate graphics programming is made simple in *Easy As Apple Pie*, and *3D-IIe* plots and rotates three-dimensional graphics on your screen.

Those who believe in a more hands-on approach to graphics will enjoy *Art at Your Fingertips*—our triple-treat review (with programs) of a powerful graphics system.

Next, strike up the band with a parade of tutorials. We lead off with *66 Keys to Graphics Success*, which explains how your first steps with your Commodore 64 or VIC-20 can create some instantly impressive graphics.

For those with a more spritely disposition, there are two special features to perk up your programs. *Don't Be a SlowPOKE* speeds up sprite routines, and *Down Memory Lane* will help you maximize character design with a minimal use of BASIC memory. And bringing up the rear, *Simon's Basic* and the *Super Expander 64* team up in *Bigger Better BASIC* to give you a wider range of commands than ever before.

The IBM PCjr is featured in its own 12-page solo section. *A Detailed Look Inside the Peanut's Shell* provides extensive, never-before-published reference information on this long-awaited micro.

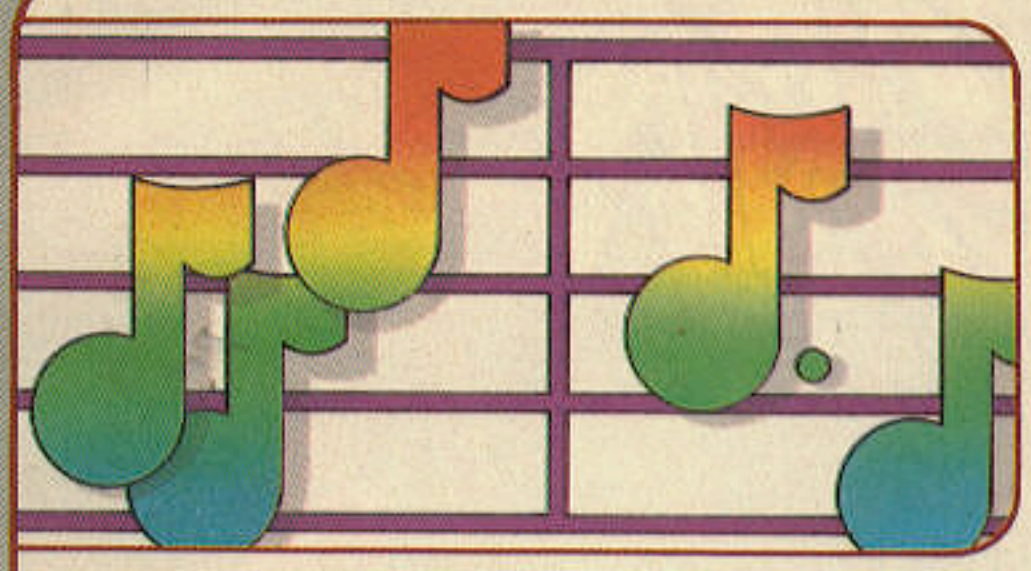
Our orchestral finale comes in a flourish of games for all instruments. TI users can try their luck at *Slots* or save the world in *Meltdown*. And for game lovers of all machine persuasions, *Flak Attack* and *Tower of Hanoi* are certain to be challenges.

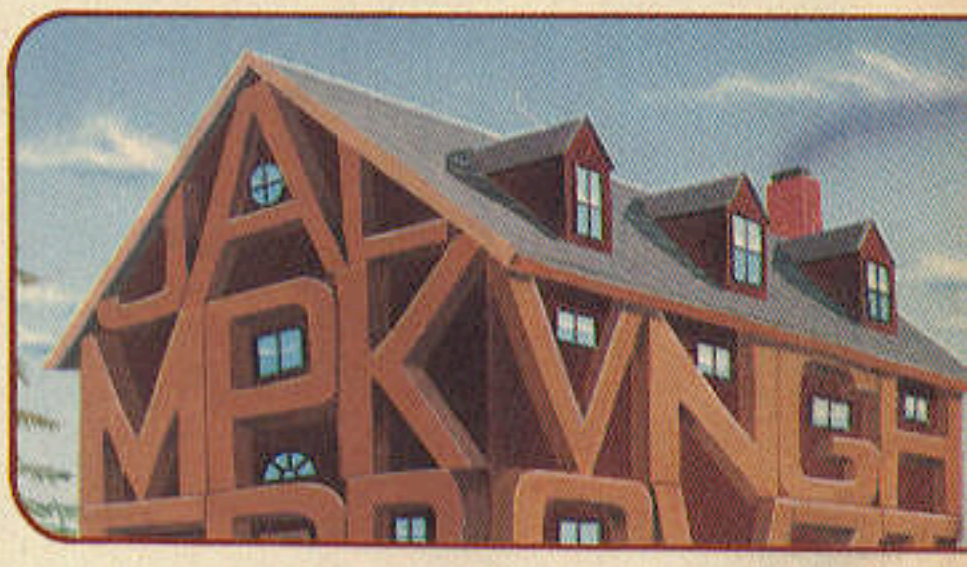
So strike the keys and join the chorus! With *Home Computer Magazine* as your conductor, you'll find the melody lingers on and on and on.

Until next month, have fun reading, learning and RUNing HCM

CONTENTS

VOLUME 4 NUMBER 1

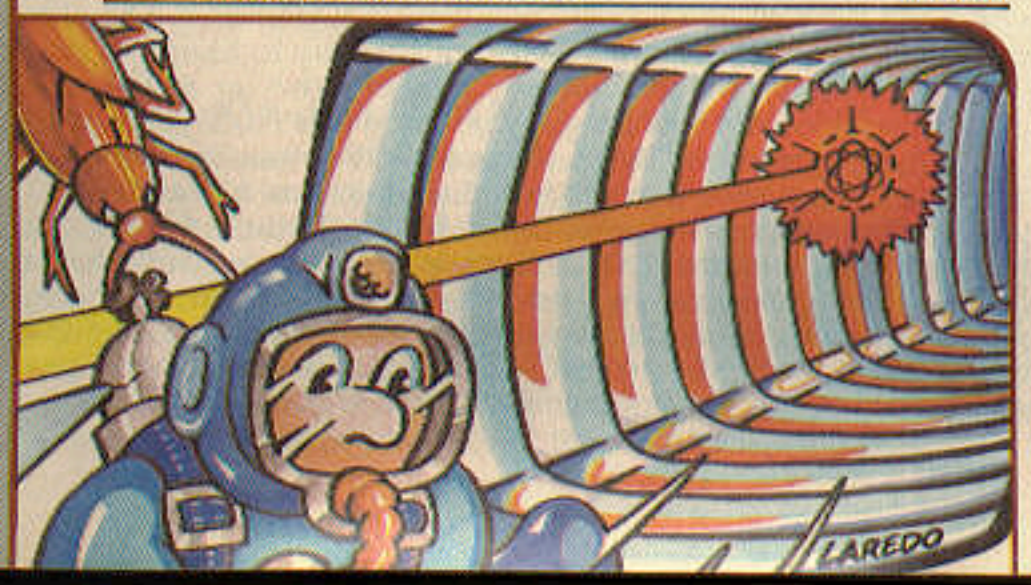




PRODUCT REVIEWS

GAMEWARE BUFFET

DEPARTMENTS



HOME COMPUTER™ DIGEST
News and Happenings in the Home Computer World

Bound in between Pages 98 & 99

by J. Larry Schott
and the HCM Staff

There's an inheritance I received from a favorite relative that I'd like to share with you through the Home Computer. When I was a kid in Kentucky, we always looked forward to family gatherings with Great-Uncle Larry. Wearing a white starched shirt, this ancient and weathered farmer would ceremoniously tune up his old fiddle, and standing ramrod straight on our shady porch, he'd play and play. As I followed the melodies flowing from his coarse hands, veined as the tobacco he tended, I picked up those traditional tunes. Twenty years later, thinking about long-gone Uncle Larry late one night in the glow of my monitor, it dawned on me what I could write with the programming skills I had learned. I set about transcribing those songs he left me, feeling that the tough part of this program was behind me. For me, figuring out *how* to program is easy compared to deciding *what* to program.

You'll recognize these melodies from animated cartoons, old Civil War movies, even commercials (if not folk festivals). Though they were passed to me on a fiddle, you can pick them out on any instrument. Mandolins are tuned exactly the same way as fiddles, flutes have the same range as fiddles, and even guitarists like David Bromberg perform fiddle tunes. The computer sounds a lot like a clarinet when it plays them. This software is flexible, so you can use it as you wish. *Larry's Ten Fiddle Tunes* can be slowed down for instruction on any instrument; it can be your partner while playing a duet; or it can simply be run solo for some entertaining examples of computer music. You can also borrow a selection for your own programs; these songs are so time-worn that they are public domain.

How exactly did I cram ten traditional fiddle tunes into a single Home Computer program in less than 16K? Read on.

Two things about all the music programs I had typed on my TI computer frustrated me. One was the endless repetition of typing CALL SOUND on every line. The other was that after all that time at the keyboard, the reward of a few seconds of music seemed too small. This program addresses both of those problems.

The first solution grew out of a hint that a friend of mine, Dave Seymour, gave me. Dave has been involved with computers since he helped solder together one of the first ones, the ENIAC, in Philadelphia in the 1940's. His golden advice: "Anything you repeat a lot, you can probably put into a loop. Let the computer do the hack work. That's what they're good for." Thus developed my plan to use only one CALL SOUND statement containing variables and continually read DATA into it. That insight was probably reinventing the wheel, as they say, but as a neophyte computer user, I was pleased with myself for thinking of it. I believe this is the first mention of such a coding system in *HCM*.

The second problem was solved by the same strategy. Since DATA takes up less room than interminable commands, the loop opened up room for more music. Six, then eight, and finally ten tunes fit into a single program. That seemed too good to be possible. However, there is a corollary to Murphy's Law: Once you solve one problem, another shows up. By filling the available RAM with program, you eliminate much of the room for processing the DATA. This causes the music to "hiccup"—i.e., pause while the computer does its internal housekeeping—throwing the tune slightly out of rhythm. [Because the program will play in either BASIC or Extended BASIC, it can face different system configurations, and it reacts differently on those configurations. It hiccups *most* when running in BASIC with the disk system attached; entering CALL FILES (1) before running the program deletes some of the disk buffers and gives the system more workspace. In BASIC with the cassette system attached, or in

Extended BASIC with the disk system attached, it runs better. The programs run best in Extended BASIC with memory expansion attached.—Ed.]

Despite this almost human imperfection (which actually sounds a lot like the way Uncle Larry played), you can play along with the computer. For those who want to accompany the fiddling TI on rhythm guitar or whatever, the musical key and the starting note for each song are printed at the bottom of the screen. All the fiddle tunes are easy to follow with two or three chords once you have an idea where to start.

Special Program Features

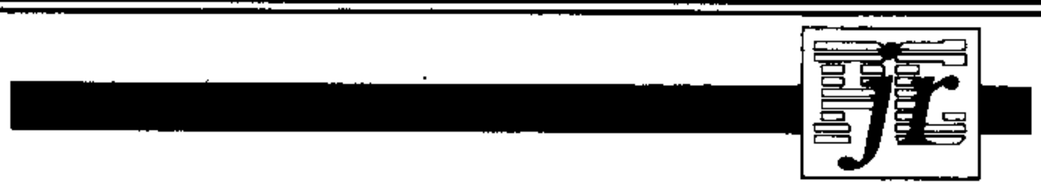
As I mentioned, only one CALL SOUND statement—line 2280—plays every bit of the music. Here's how the play routine works. Line 2220 checks for the end of the melody line flag (99), and if it finds the flag, the program returns to the tune's subroutine. The next line checks to see if the variable D it just read is really a duration (less than 110) or is instead a frequency (necessarily over 110). If it is a duration, the program reads the next DATA as the frequency—line 2240—and goes right to the CALL SOUND to play the note. If not, the frequency is set equal to that D, the duration is assumed to be 1, and *then* the note is played. This process is used because the overwhelming number of notes are duration 1. One little routine saves typing the duration for all those notes. The 1 can be thought of as a sixteenth note, 2 is a quarter note, and so on. The decimal .7 roughly approximates a note of a triplet. It's actually a little long, but it sounds right.

The actual speed is determined by multiplying the duration by the variable SP. You can permanently set the value of SP in line 200 at the beginning of the program, or you can adjust it with menu choice (12). The suggested lower parameter of 50 is not locked into the program, and you can go as low as 2 without blowing it up. However, because of the time needed to read and run through the routine, all values below 50 sound pretty much alike—rather like the mechanical rock group DEVO. Hiccups show up less at slower speeds, which are also better for learning the tunes.

All other possible input errors are locked out of the program at lines 560 and 1590, making the program very user-friendly while running. I also designed the program to be user-friendly during the long, boring, mistake-prone process of typing in DATA. First of all, each DATA section of melody is clearly marked by a REM with the initials of the song. Each second melody line is additionally identified by a 2. For example, SJ is the first line on "Soldier's Joy"; SJ2 is the second line. This should make it easy to find the mistyped DATA if something sounds wrong or the program stops in the play routine.

To make it less likely that you have a typo to begin with, each line of DATA contains *sixteen* numbers. If there is an odd amount of DATA, there will be a few numbers right before the next REM statement. Even when the remainder was just one number, the end of section flag 99, I gave the flag its own line to maintain consistency and make typing in the DATA easy. As additional insurance against accidentally changing something in the DATA (I suspect it has happened), only the following numbers appear in these songs: durations .5,.6,1.5,2,3; frequencies from 220 to 1175 (all are numbers from the note chart in

Continued on p. 14



The one factor that sets the IBM PCjr version of this program apart from the others is the way in which we create the sound. The PCjr uses the Microsoft BASIC PLAY command, but has the ability to "PLAY" with three voices. The PLAY command allows you to compose music with notes and scales, rather than having to figure out the frequencies. The music is composed by building a string of subcommands for the PLAY statement to execute. You can designate a note with the letters A through F, and even designate sharps with the # sign. The L command is used to set the length of the note. For example L8 would make the music play eighth notes, while L4 would play quarter notes. The T command sets the tempo of the music in beats per minute. For example T120 would play 120 beats per minute which is about normal, while T40 would play at 40 beats per minute and be three times slower.

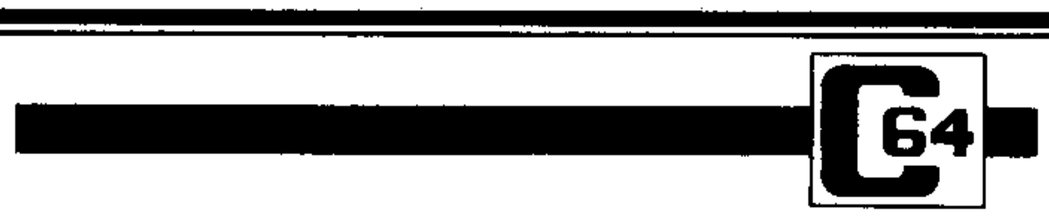
There are eight octaves available in the range of the sound chip. [This is the same Texas Instruments sound chip found in the TI-99/4A—Ed.] There are two ways you can specify the octave you are using. The first way should always be one of the first commands in your PLAY command string. The O command, followed by a number from 0 to 7, will select an octave for all of the following notes to be played at until the octave is changed. You should always place this command before any notes are played in each string so the computer will know which octave to use.

The second way to change the octave is with the < (less than) and > (greater than) signs. The < sign lowers the octave and the > raises it. The following string could be interpreted like this:

"O2 BAB>BAB<BAB>>DEF#"

The octave is set to 2 and the notes B, A and B are played. The octave is then raised to 3 and the notes are repeated. The octave is then lowered back to 2 and the same notes are played again. The octave is then raised to 4 with the >> and the notes D, E, and F# (F sharp) are played.
The last command used in this program is P. This command causes a pause of one beat. —*W. K. Balthrop*

Continued on p. 16



One of the subroutines in *Larry's Ten Fiddle Tunes* can be adapted for your own programs when you want them to play music.

In order to play a note on the Commodore 64 we must first look up the two corresponding values to be POKEd (or moved) into two registers of the sound chip to generate a particular note. These values can be found in the table of musical notes in the Appendix of the *Commodore 64 User's Manual*. The two registers are actually combined to form one sixteen-bit number that corresponds to the frequency of the note. If we want a succession of different notes (as in a melody), we need a way to input these POKE values to play the tune. We could put all the necessary POKE values directly into DATA statements, read them in and POKE them into the sound chip, but there is a better way.

It is often more practical to deal with musical notes by relating each note to its corresponding audio frequency rather than to a high or low POKE value that is meaningless outside the world of the Commodore 64. This will also shorten the DATA statements since we will have only one number instead of two for each note. *Larry's Ten Fiddle Tunes* uses a method based on these practical considerations to generate the notes for each of the ten tunes. If, for example, you want to play the A above middle C, first find its corresponding audio frequency (440 hertz) and put this into a DATA statement along with frequencies for the other notes to be played in sequence by the Play Note subroutine.

The Play Note subroutine will do several chores for us. First it calculates the corresponding high and low POKE values; then it sets up all the other parameters associated with playing a note on the Commodore 64. Next it turns the sound chip on and goes into a FOR–NEXT loop for a designated interval (equivalent to the duration of the note). Then it turns the sound chip off and returns to the calling program. The Play Note subroutine starts on line 8600 and returns on line 8660. Before using this routine we must set the variable G equal to 54272 (the beginning address of the sound chip registers). It is also a good practice to clear all of the sound chip registers whenever a new type of tone is used. Lines 270 and 280 perform these two steps.

Continued on p. 15

Uncle Larry

the instruction manual); flag 99, which signals the end of a musical phrase; and the frequency 22222, used as a musical rest since it is too high to hear and easy to type.

Another innovation to help spot problems during debugging is the spread of DATA lines throughout the subroutines as remarks. These DATA lines are used to print the menu each time, in lines 290-500. However, their placement in lines 630-1440 indicates which subroutine controls which song. This was done to keep from entering redundant REM statements. I wanted to document the program thoroughly but I hated to waste RAM.

The second part of each of these DATA lines (everything after the comma) is stored in the array A$. This information is printed as M$ after the menu selection is made, displaying the musical key and the first note and string number. To help you understand the rest of the program, the variables are obvious: **D**uration, **F**requency, and **SP**eed in the CALL SOUND; **R**ow and **C**olumn (think of **C** as a TAB) in the PRINT M$ (any message); **N**ote (the ASCII of G, D, A or E for tuning), **T$** to pull title information out of array A$, and the ubiquitous all-purpose counter **X**. The **X** counts characters printed during the PRINT routine and keeps track of where the program is during the play routine. Since it never prints and plays at the same time, the same counter variable can be reused for each section.

Extended BASIC

The program was originally written in Extended BASIC, but I broke it down to plain vanilla TI BASIC to reach the widest possible audience. Some of you will want to change it back. In Extended BASIC, the menu was written very differently, taking advantage of DISPLAY AT and ACCEPT AT. Though that code was much shorter, the display was more static without the drama of scrolling the menu each time. You might want to make cosmetic changes if you have Extended BASIC. Stacked statements in Extended make the whole program much neater; for instance:

```
650 RESTORE 2340 : : GOSUB 2210 : : IF X<2 THEN 650
680 RESTORE 2410 : : GOSUB 2210 : : IF X<4 THEN 680 :
    :RETURN
```

Two stacked lines replace seven BASIC lines. This is a good place to see how each subroutine to play the tunes works. The format for each tune is traditionally the same—AABB—where the first line is played twice, followed by the second section twice. The counter X, incremented in line 2300, keeps track of where we are in that formula. The play routine can be dramatically condensed from eleven lines to three, as shown in lines 2210 to 2300:

***LARRY'S TEN FIDDLE TUNES* (TI)**
Explanation of the Program

Line Nos.	
100-180	Program header.
190-250	Initialization.
260-280	Change color of lower case letters.
290-500	Print menu.
510-620	Select from menu; print pertinent information from array to bottom of screen.
630-1430	Ten subroutines for the ten songs.
1440	Print a blank line in menu for readability.
1450-1570	Draw head of instrument and strings for tuning utility.
1580-1610	Choose string to tune.
1620	Leave tuning section when 0 is pressed.
1630-1770	Note, frequency and row placement information for tuning peg.
1780	CALL SOUND for tuning (negative duration allows interrupt).
1790-1930	Draw peg to turn.
1940	Display letter of note on peg.
1950	Loop to stay in tuning routine.
1960-2030	Change playing speed.
2040-2140	End message.
2150-2190	Print message subroutine.
2200-2310	Play music subroutine.
2320-3340	DATA for music.

```
2210 READ D : : IF D=99 THEN 2300 ELSE IF D<110 THEN
     READ F ELSE F=D : : D=1
2280 CALL SOUND (INT(SP*D),F,1) : : GOTO 2210
2300 X=X+1 : : RETURN
```

The information in the tuning subroutine can be similarly compacted, and DISPLAY ATs can be used to draw the pegs and name the notes. You can also replace the whole PRINT M$ routine (lines 2150-2190) with a single DISPLAY AT statement. Of course, the most important and lengthy part of the program, the mass of DATA, is unchanged in the switch from BASIC to Extended BASIC. Although you could cram more numbers on each line, I believe the increased possibility of typing errors outweighs the savings.

Play it Again, Uncle Larry

So now you can put into your computer a part of my uncle and hundreds of other folk musicians stretching back through centuries. I'm not sure what my taciturn tutor would have said about my mechanical method of teaching you his tunes. All I know is that he always smiled when he played. These songs made him happy, and if passing them along in this new-fangled way makes any of you smile, then I guess he'd be in favor of it.

Now hit the electricity and rosin up the bow!

TI-99/4A

```
100 REM  *********************
110 REM  *    LARRY'S TEN     *
120 REM  *     FIDDLE TUNES   *
130 REM  *********************
140 REM  BY LARRY SCHOTT
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.1.1
170 REM  TI BASIC
180 REM
190 REM  INITIALIZATION
200 SP=175
210 CALL SCREEN(8)
220 CALL CHAR(33,"E7E7E7E7E7E7E7E7")
230 CALL CHAR(44,"FFFFFFFFFFFFFFFF")
240 CALL CHAR(95,"0808080808681F0600")
250 DIM A$(14)
260 FOR X=9 TO 11
270 CALL COLOR(X,5,1)
```

Continued on p. 30

Uncle Larry . . . from p. 13

Now let's analyze the subroutine itself. The central segment of the program, lines 8600 to 8660, plays the notes for *Larry's Ten Fiddle Tunes*. Line 8600 will derive the high and low Poke values necessary to drive the sound chip in the correct tone:

8600 FD=INT(F/.06097) :HF=(FD/256) :LF=FD-(256*HF)

HF equals the high frequency; LF equals the low frequency. FD is an interim variable that is used only in the calculation. F equals the note frequency in hertz.

8610 POKE G+5,184:POKE G+6,169

sets the attack/decay and the sustain/release values to simulate the tone of a fiddle.

8620 POKE G+24,15

sets the volume level to 15 (maximum).

8630 POKE G+1,HF:POKE G,LF

POKEs the values derived by line 8600.

8640 POKE G+4,33

does two things: It generates a sawtooth wave form for voice 1, and it initiates the attack/decay cycle.

8650 FOR ZX=1 TO INT(SD*D): NEXT

determines the length of time the note will play. The two control variables are SD and D. The value of SD is constant throughout the program unless changed by Option 12 of the menu (Change Speed). The variable D is used to simulate note values, such as eighth notes, quarter notes, and so on.

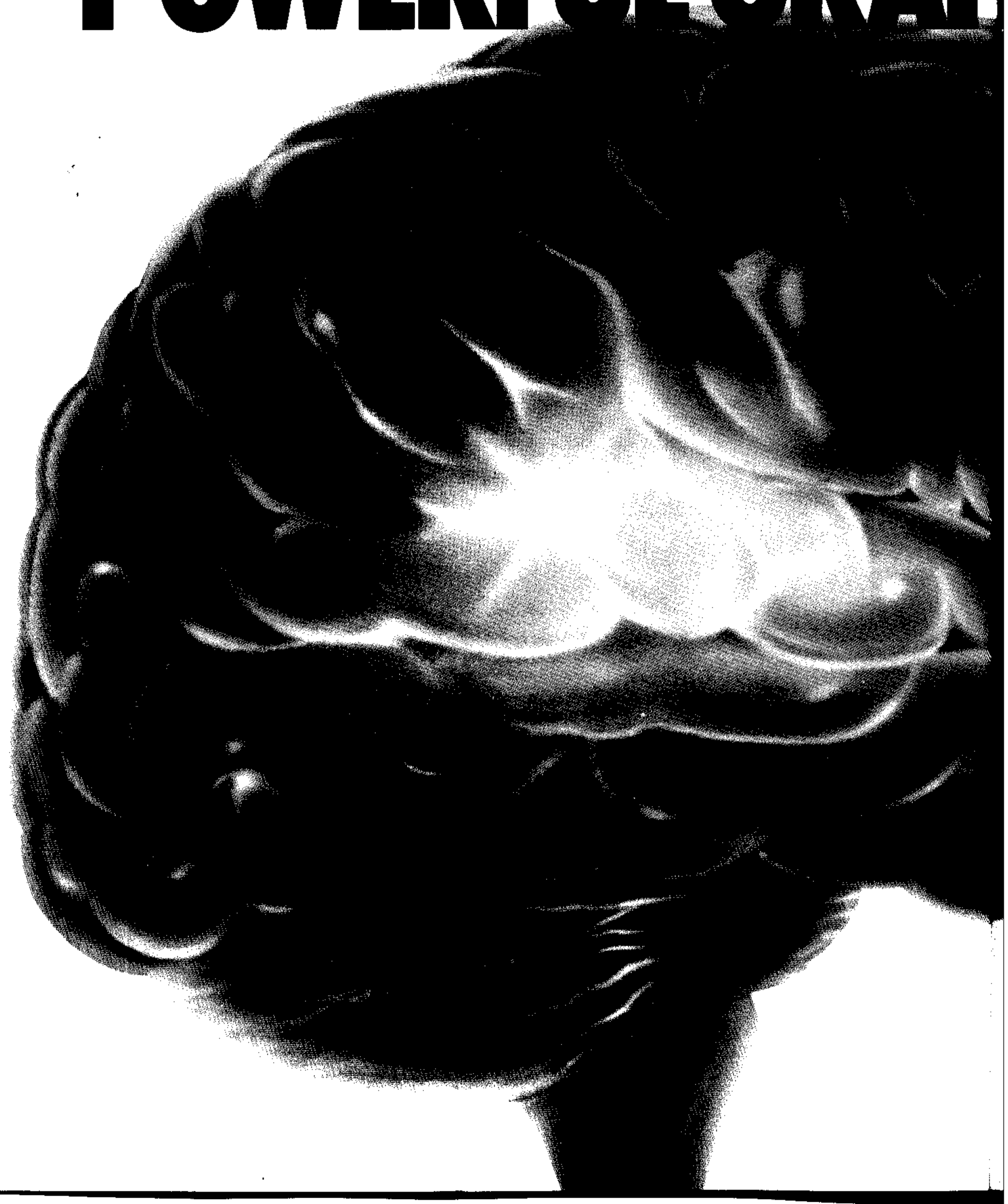
8660 POKE G+4,32:RETURN

initiates the release cycle and effectively turns the sound chip off at a speed determined by the value poked into G+6 in line 8610. We also return from the subroutine here.

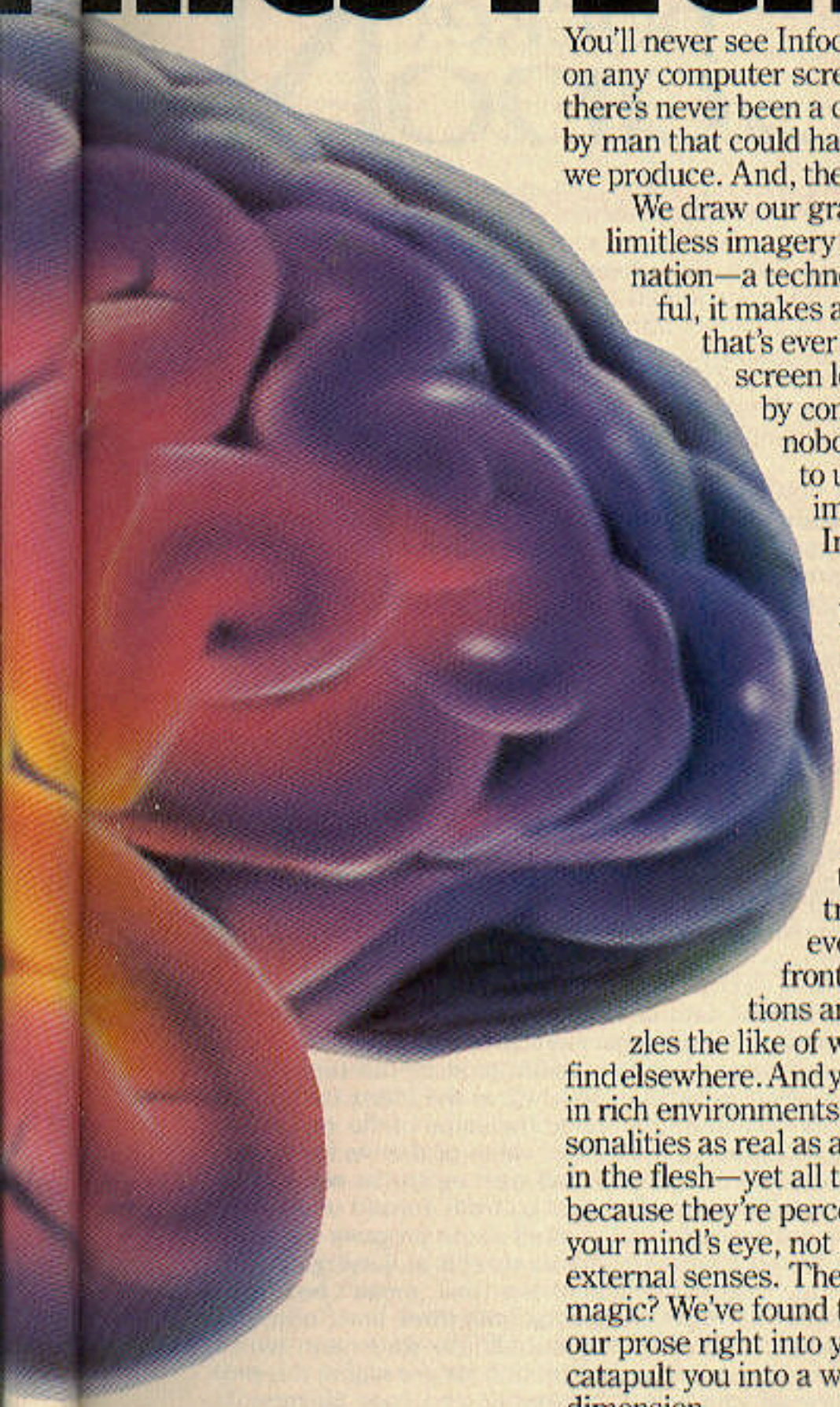
—John Thrasher

Continued on p. 174

WE UNLEASH TH
POWERFUL GRAP



HE WORLD'S MOST PHICS TECHNOLOGY.



You'll never see Infocom's graphics on any computer screen. Because there's never been a computer built by man that could handle the images we produce. And, there never will be.

We draw our graphics from the limitless imagery of your imagination—a technology so powerful, it makes any picture that's ever come out of a screen look like graffiti by comparison. And nobody knows how to unleash your imagination like Infocom.

Through our prose, your imagination makes you part of our stories, in control of what you do and where you go—yet unable to predict or control the course of events. You're confronted with situations and logical puzzles the like of which you won't find elsewhere. And you're immersed in rich environments alive with personalities as real as any you'll meet in the flesh—yet all the more vivid because they're perceived directly by your mind's eye, not through your external senses. The method to this magic? We've found the way to plug our prose right into your psyche, and catapult you into a whole new dimension.

Take some tough critics' words about our words. SOFTALK, for example, called ZORK® III's prose "far more graphic than any depiction yet achieved by an adventure with graphics." And the NEW YORK TIMES saw fit to print that our DEADLINE™ is "an amazing feat of programming." Even a journal as video-oriented as ELECTRONIC GAMES found Infocom prose to be such an eye-opener, they named one of our games their Best Adventure of 1983.

Better still, bring an Infocom game home with you. Discover firsthand why thousands upon thousands of discriminating game players keep turning everything we write into instantaneous bestsellers.

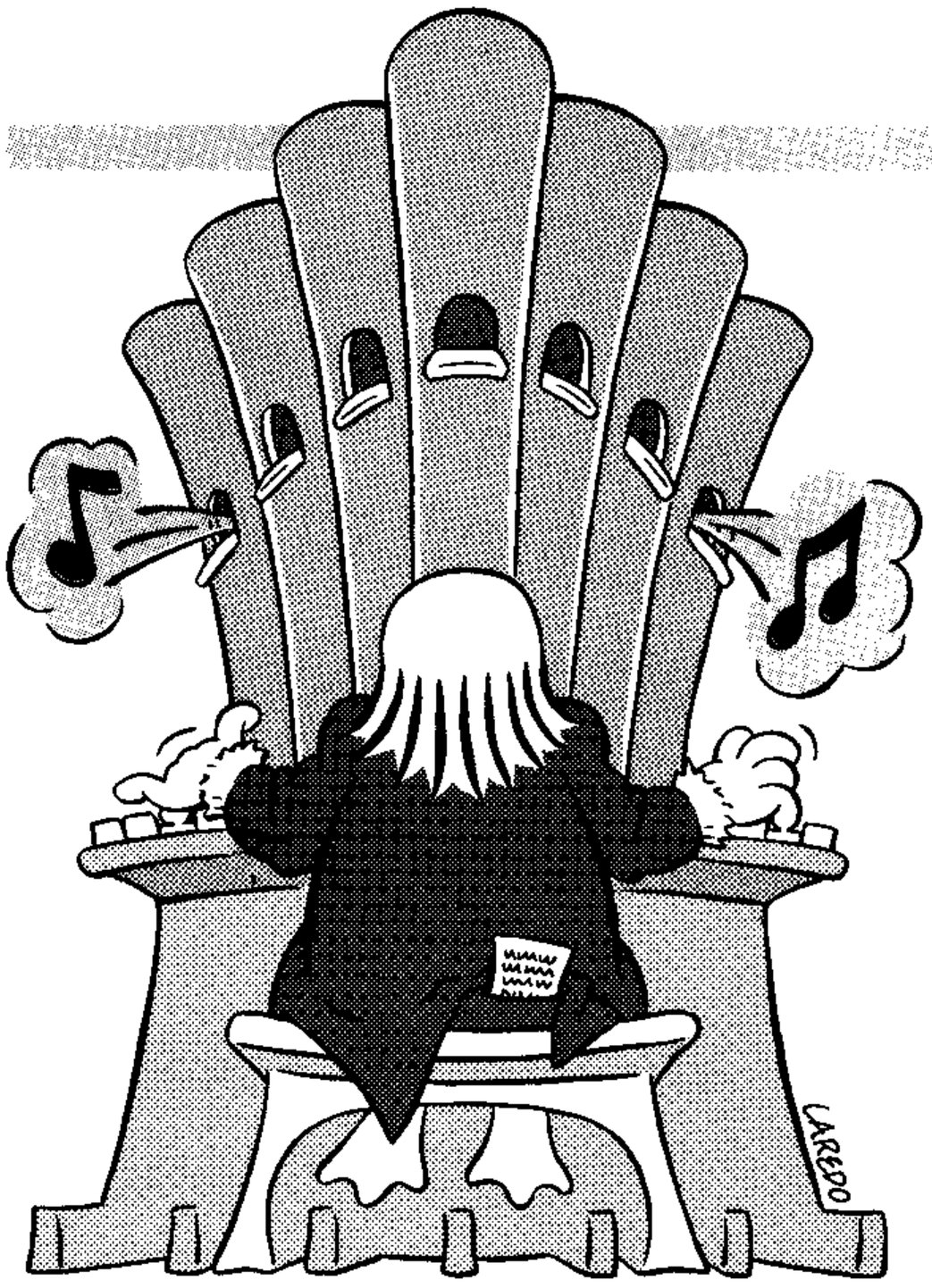
Step up to Infocom. All words. No graffiti. The secret reaches of your mind are beckoning. A whole new dimension is in there waiting for you.

(For more information on Infocom games contact: Infocom, Inc., P.O. Box 855, Garden City, NY 11530.)

INFOCOM™

The next dimension.

For your: Apple II, Atari, Commodore 64, CP/M 8," DEC Rainbow, DEC RT-11, IBM, MS-DOS 2.0, NEC APC, NEC PC-8000, Osborne, TI Professional, TI 99/4A, TRS-80 Model I, TRS-80 Model III.



PROFESSOR HOLL'S pocket CanoN

by S. T. Holl

Be seated at the keyboard, Maestro,and play this little composition scored for the TI-99/4A computer:

```
100 REM  *********************
110 REM  *   POCKET CANON    *
120 REM  *   BY  PACHELBEL   *
130 REM  *********************
140 REM  BY S. T. HOLL
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.1.1
170 REM  TI BASIC
180 REM
190 DIM F(7)
200 DATA 0,0,262,196,220,165,175,131,175,196
210 READ N,V,F(0),F(1),F(2),F(3),F(4),F(5),F(6),F(7)
220 DEF M(X)=INT(X)-8*INT(X/8)
230 CALL SOUND(500,F(M(N/4)),3,1.5*F(M(N/2)),30+29*(V>31),2*F(M(N)),30+30*(V>63))
240 N=N+1+32*(N>31)
250 V=V+1
260 GO TO 230
```

If this doesn't work the first time you run it, and the error message is INCORRECT STATEMENT IN 230, then check line 230 carefully. Did you notice that there are two closing parentheses at the end of the line? How about that single period—not comma—in 1.5? If the error message is DATA ERROR IN 210, be sure to check line 200 too—it contains ten entries separated by commas; because some of these entries are duplicates of others, it is not difficult to lose your place when typing them in.

Now that you have the program running, let us discuss what it is you are hearing. A *canon** is a musical form in which the melody of the leading voice is imitated in some fashion by the other voice or voices. The imitating voices can be at the same tempo and pitch as the leader, or they can vary either or both. When you started this program, you heard the theme alone, played very slowly by the first voice; then the second voice joined the first, playing the theme twice as fast and a perfect fifth higher; and finally the third voice chimed in four times as fast and a full octave higher than the lead. A *round* like "Frere Jacques" or "Row, Row, Row Your Boat" is a simple canon in which the imitating voices use the same tempo and pitch as the lead but start at different times.

Now, how did we pack all that into eight program lines? The melody itself is stored in the third through tenth entries in the data statement. You could substitute your own eight-note theme by replacing those numbers with new ones you have looked up in the musical frequencies table on page III-7 of your *User's Reference Guide*. This particular theme comes from a celebrated canon of Johann Pachelbel. His harmonization is different—more complex, and much more beautiful—from our childhood rounds. Pachelbel's Canon in D is the perfect accompaniment for golden sunrises, but well worth the hearing any time.

RCA Vector

Line 210 initiates two counters we'll need later, and then reads the frequencies of the notes into the vector F. You may think of a vector (or array) as a series of numbers all using the same name. To specify the first one we use F(0), the second one under that alias is F(1), and so forth. CAUTION: When using vectors, always declare them first with a DIM (for DIMension) statement which tells the machine things it needs to know to manage them properly.

The advantage of using a vector to store our tune is that we can use an expression like N/2 as the index to the vector(e.g., F(N/2)),and then when the value of the expression changes, it refers to a different value of the vector F.

Canons play themes over and over again. So we want to make an index for the F vector go from zero to seven over and over again in a natural manner as our program executes, and we would like to be able to do this at varying speeds.

The mechanisms which make this music box play Pachelbel's theme are compacted into three lines: numbers 220,230, and 240. Line 220 is a DEFine statement, which establishes a base 8 modulo function for use within this program. We all routinely use base 12 and base 60 modulo arithmetic when we do problems involving time. For example, to find out what time it will be 19 hours from now, we add 19 to the present time and then subtract twelves until the hour is 12 or under. Since there are 8 notes indexed 0 through 7 in this theme, we need a transformation which will turn any number into its modulo 8 equivalent, then round that down to an integer. Line 220 defines such a function, M( ). Try a few different values of X; you will see that this is true. Incidentally, this definition of M( ) works for negative numbers as well as positive ones. We'll not make use of that generality here, but it comes free, and if you pursue program-

*Ordnance enthusiasts who have managed to read this far without remarking about the single n in the title should turn immediately to the Gameware Buffet, where they will find the usual fine laser and photon torpedo smorgasbord.

ming, you're certain to need it again somewhere else. So jot down the function (or maybe this magazine issue and page) on a 3x5 card and file it under Modulo Arithmetic.

The purpose of line 240 is to increment by one the counter N each time it is executed until you reach 32. At this point N is reset to zero and the incrementing continues. The expression in parentheses is relational; during execution that expression is replaced with zero when it is false, with minus one when true.

Line 230 is the one actually making the tones. The first of its seven arguments is the duration of the sound. Five hundred milliseconds is about as short as the notes can be without producing gaps between them while the computer finishes the other things it needs to do in each iteration of the program. The remaining six arguments, separated by commas, are the frequencies and volumes of the three voices, in pairs. Volume can range from loud (zero) down to inaudible (30). The volume for the first voice, which is the one playing the slowest tune, is set at three. The volumes for voices two and three start at 30 but contain relational expressions which "turn up the volume" when the counter V (initially zero and incremented by one every iteration) reaches 32 and 64, respectively. This corresponds to one and two complete playings of the theme by the first voice.

Perfecting the Pitch

Now let us look at how the three frequency entries work. In line 230 they are:

first voice: F(M(N/4))
second voice: 1.5*F(M(N/2))
third voice: 2*F(M(N))

The array is loaded in line 210 with the frequencies of the notes of the theme in the DATA statement in line 200. Voice 1 uses those values of F directly. Voice 2 uses the same frequencies increased by 50 percent. If you try this for several of the frequencies listed on page III-7 of your manual, you will see that adding half again to the frequency will make it five whole notes higher. (There will, perhaps, be a difference of 1 in the last place due to roundoff in the table.) Voice 3 uses twice the frequencies in F. Can you tell from page III-7 what the effect of this is?

"Pachelbel's Canon in D is the perfect accompaniment for golden sunrises, but well worth the hearing anytime."

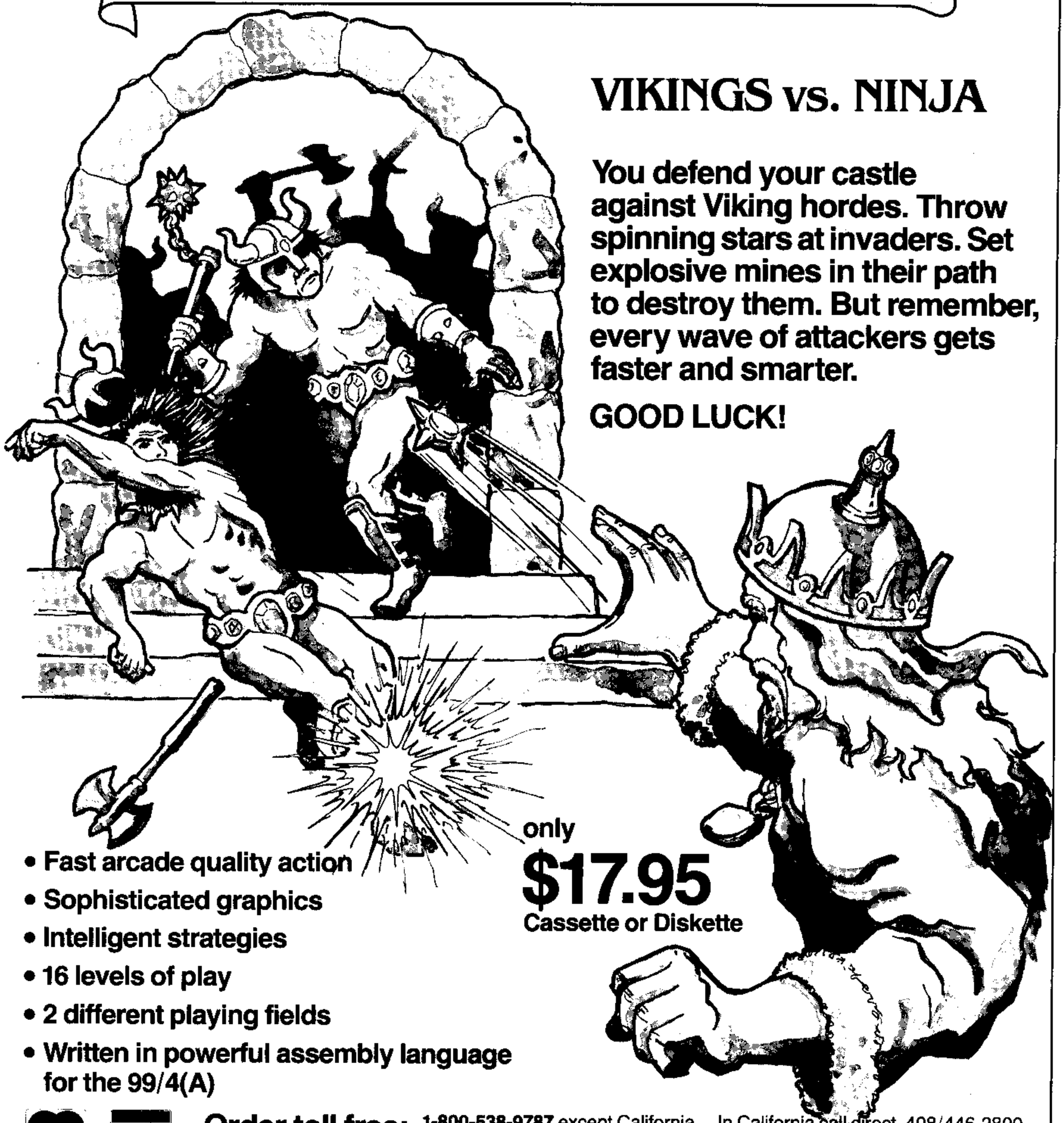
The expressions M (N/4), M (N/2) and M(N) determine which note from the vector F is to be played during a given invocation of CALL SOUND. We already know that N goes from zero up to 31 in steps of one and then starts over. We also know that the function M( ) returns the modulo 8 equivalent of its argument. In the case of the first voice, this means that as N goes from 0 to 31, the value N/4 increases by 1/4's: 0, 1/4, 1/2, 3/4, 1, 1¼, 1½ . . . up to 7¾, and M(N/4) takes the values 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, etc., since M( ) rounds down to whole numbers modulo 8. In the case of the second voice, we have M(N/2) running 0, 0, 1, 1, . . . up to 7, 7 and then starting over at 0 again as N passes its halfway point. The third voice goes 0, 1, 2, . . . up to 7 four times as N goes from 0 to 31.

Now that you know where the theme is hidden, how the relative pitches of the three voices are set, and how the relative speeds of the three voices are determined, you should be able to write 8-note canons of your own. Themes of different length will require a bit more modification (to the modulo function, for example), but not much more. Now go ahead and blast your canon!

[Editor's note: Send in your best canons and fugues and we will publish as many as space in our letters column permits. Would anyone like to write an article and program to play Ragtime?]

HCM

KING of the CASTLE



VIKINGS vs. NINJA

You defend your castle against Viking hordes. Throw spinning stars at invaders. Set explosive mines in their path to destroy them. But remember, every wave of attackers gets faster and smarter.

GOOD LUCK!

- **Fast arcade quality action**
- **Sophisticated graphics**
- **Intelligent strategies**
- **16 levels of play**
- **2 different playing fields**
- **Written in powerful assembly language for the 99/4(A)**

only

$17.95

Cassette or Diskette

MasterCard VISA **Order toll free:** **1-800-538-9787** except California. In California call direct 408/446-2800 and deduct $2.00 from the purchase price. Operators standing by 24 HRS.

KING of the CASTLE $17.95

Specify: ☐ Cassette ☐ Diskette

☐ Check or M.O. enclosed

Charge to VISA ☐ MASTERCARD ☐

Expiration date _____ Account Number _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

CYDEX SOFTWARE
P.O. Box 14471
Portland, Oregon 97214

NO RISK GUARANTEE

If you are not totally satisfied with the excellent quality of this software, return it within 15 days for a full refund.

Letters to the Editor

Slick Solutions

In your September, 1983 issue, Donald Beck presented an interesting problem. He challenged readers to write a BASIC program which found all the four-digit numbers that equal their original value after you divide them by two, add together the halves, then square the sum . His answer is inefficient, however. Here's a BASIC program that does the job over a hundred times faster:

```
100 CALL CLEAR
110 XHI=INT(SQR(9999))
120 XLO=INT(SQR(1000))+1
130 FOR X=XLO TO XHI
140 Y=X*X
150 Z=INT(Y/100)
160 IF (Y-Z*99)<>X THEN 180
170 PRINT Y
180 NEXT X
190 END
```

Woodrow Wilson
Olivenhain, CA

That's a very slick and fast solution to Mr. Beck's problem. Soon we'll have more problems for readers to work on, as you may have seen from the note in the November issue.

Memory Not Jarred

I wish to comment on two items mentioned in your September issue.

The first is a comment on the Foundation memory card. I got the 128K version and upon installation found that the card is about a sixteenth of an inch higher than the TI cards. This pressed it very firmly into the top pads of the PES box and precluded any movement, even when some varlet dropped the box on its end. This card not only survived this incident, but it has survived some very rugged trips to my North Carolina summer home in either the back of a Toyota Land Cruiser or in a little baggage trailer that I use. I do recommend that it be installed between the PES card and the RS232 card. My PES box was in a special carrier that purposely has little padding but is constructed of thick plywood. I am testing one system for possible use in Central America for portability under rural transportation conditions.

The second has to do with the letter by Clifford Parms on page 49 of the September issue concerning two drives in one. I ordered a pair in March from Western Micro Systems and received them in late June. What they sent me were two TEAC disk drives in the exact configuration that I ordered. I have not had any problems with the power supply in the PES but do recommend the added power supply that they sell as an extra. They, too, survived the summer trips.

This is my experience with these two items for whatever it may be worth.

I do like the magazine very much.

Jasper Pierce
Sumter, SC

It sounds like your equipment has taken as rough a beating as any is likely to get, and it's encouraging to hear that everything held up.

Generation Gap

I have been using the IBM PC for quite a while at the office and have amassed a great deal of software for it. I'm currently interested in purchasing the IBM PCjr for my home and am curious as to whether or not my PC software will work on the PCjr. Or will I have to buy all new software for my PCjr at home? I understand that the languages the two systems use are different as well.

Arlene Delmonico
Chicago, IL

Many of the programs written for the IBM PC will also run on the PCjr, because IBM has made an attempt at keeping the two systems compatible. A problem you may run into will be with software which requires more than the maximum 128K of memory in the PCjr including the screen memory—or uses special machine language routines that are not part of the regular BIOS calls. [See related article in this issue]. Also, because the PCjr has only one disk drive, software which requires two drives will not work unless it can be adapted to a single drive system. To get a comprehensive list of programs which are compatible with both systems, have your local IBM dealer show you the PCjr software compatibility chart.

As far as languages go, the PC's Advanced BASIC (BASICA) will not run on the IBM PCjr because it occupies the same area of memory used by Cartridge BASIC. Cartridge BASIC is located in ROM (Read Only Memory) at the cartridge port. But don't despair. Any program written in BASICA will run under Cartridge BASIC because Cartridge BASIC is a superset of BASICA. Programs written in Cartridge BASIC on the PCjr may not run on the PC under BASICA, however, because they may have Cartridge BASIC commands that aren't in BASICA.

A Difference in Assembly

I would like to commend you on a publication that I find useful and far more informative than any other in the small computer field. Your subject matter is well treated and the range of coverage that you provide on the Texas Instruments brand of computers is unmatched by any other computer magazine that I know of today. However, I do have a bone to pick with you.

Most Assembly Language programs that you publish are written for the Mini Memory Module. Many of us would like to convert these to run with the Editor/Assembler package but are unsure of just exactly how to modify them. As they now stand, they definitely do not run with the Editor/Assembler and 32K card; I know, I've tried.

David L. Ramsey
Woodbridge, VA

There are three major differences between programs for the Mini Memory and the Editor-Assembler cartridges:

Location of utility routines. They reside at different locations in the two systems. Wherever a Mini Memory program refers to a utility routine

HCM Review Criteria

Each month, *HCM* reviews software packages for the IBM PC and PCjr, Apple II, II+ and IIe, TI- 99/4A, and Commodore 64 and VIC-20 computers. These reviews take a detailed look at the quality of commercially available third-party software for these home computers.

At the beginning of each review, a review-at-a-glance box provides the user with an instant assessment of the program. Each software item will be evaluated, where relevant, with the criteria below.

- **Performance**—how well the activity responds to the player's commands; how well the sound effects, music, or speech are integrated with the software.
- **Documentation**—the quality of the printed matter that comes with the software: whether the instructions are clear and comprehensive; whether the machine configuration requirements are spelled out. Information such as how to load the program, use the keyboard, and restart the activity contributes to the documentation rating, as do tips on performance peculiarities.
- **Engrossment**—whether the game or activity has that intangible quality that holds the player on the edge of his seat while the hours tick by unnoticed.
- **Ease of Use**—the degree to which a user can interact with the software without outside help; the ease and effectiveness of error-handling features; whether the actual reading level of the activity is appropriate for the suggested audience.

Education-Specific Criteria

Educational software may also be evaluated in the following areas:

- **Concept Presentation**—whether the concepts are presented clearly, in logical order, and in enough depth for the learner to be able to apply the learning to other situations.
- **Rewards**—whether the audio-visual rewards are motivating and whether they are appropriate to the activity.
- **Graphics**—rates the quality of the graphics and whether they enhance or detract from the educational purposes of the activity.

Letters to the Editor

(by address unless you *EQUate that address to a label), you should substitute an Editor/Assembler mnemonic for that routine.*

Entries in the REF/DEF table. In Mini Memory, you must insert the information necessary—program title and entry point—into the proper memory locations, and then change the starting address of the table. The Editor/Assembler does this automatically with the REF and DEF assembler directives at the start of the program.

Memory use. The space occupied by the Command Cartridge is not available for use with the Editor/Assembler. Thus any references to addresses between >7000 and >7FFF must be changed. The best way is to change all those address references to label references, which can be relocated.

As far as editorial balance is concerned, we want to cover as many aspects of the world of the Home Computer as we can, while still providing material that the greatest number of our readers can use. Those two requirements sometimes conflict; we try to strike a happy medium.

C-64 Keyboard Buffer

I am having a problem with the keyboard buffer on my Commodore 64. I have to be very careful that I don't accidentally hit any keys while running my program, because if I do, these keys get entered into the keyboard buffer, and when the program starts executing code that accesses the buffer, it reads these garbage inputs as input data. What can I do about this?

Jeff Strong
Canton, OH

There are a couple of ways to correct this problem. One method would be to clear the buffer just before accessing it. The code would look like this:

```
100 GET A$:IF A$<>"" THEN 100
```

The next line of code would then read:

```
110 GET A$:IF A$="" THEN 110
```

You can correct this another way by not accessing the keyboard buffer at all. There is a byte, maintained by the keyboard scan routine, that contains the unique code of the key currently being pressed. This byte can be read by PEEKing address 197. The code that will be returned for each key pressed is not part of any standard coding, e.g., ASCII, EBCDIC, etc. If you want to develop a table that matches codes to key presses, try inputting the following BASIC line in the command mode:

```
FOR X=1 TO 3:Z=PEEK(197):PRINT Z:X=1:NEXT
```

To find the code for each key, simply press the desired key and watch the code scroll up the screen. You can use this method in a program by inserting the following BASIC statement:

```
100 Z=PEEK(197):IF Z=64 THEN 100
```

64 is the code that will be placed in address 197 when no key is being pressed. Good luck, Jeff.

Memory Pages

I have been trying to learn more about the inner workings of my Commodore, because I want to start some machine language programming. But explanations of machine language programming are hard for a beginner to understand, partly because everyone uses terms without defining them. One term that everyone seems to use without explanation is "page." What is a "page" of memory?

Henry J. Feingold
Roanoke, VA

If you were to write a story, you would think of the work as a whole, but if the story were printed, the physical structure of the book would break the story up into pages. Similarly, we would like to deal with our computer's memory as a whole, but the hardware breaks the memory up into pages. *In BASIC programming, for instance, Commodore handles virtually all the difficulties of going from page to page for you.*

When you write assembly language programs, the situation is a bit different, and in that case, the zero page *will be important. A page is 256 memory locations, each one byte long, and each of those 256 bytes has an associated memory address. These memory addresses run from 0 to 65535. Most of these addresses (addresses,* not *memory locations) are two bytes long. The addresses on the zero page (the first 256 bytes of memory) are only one byte long. The Commodore operating system uses the zero page for many operations where speed is important because it evaluates these one-byte addresses more quickly than the addresses on other pages.*

Apple Text Files

When I write programs that create text files on my Apple IIe, I sometimes have problems with them. If I have a program that opens and writes to the same file with the same filename each time it runs, sometimes I get garbage at the end of that file, and I can't figure out why. Sometimes it looks like text from previous runs of the program. I hope you can help me find a solution to this problem.

Horst Wiener
Pompano Beach, FL

This is a common problem that all Apple programmers have to cope with. If a new text file is shorter than an old one with the same name, it will have part of the old file on the end. (See page 66 of Apple's DOS Programmer's Manual.*) To avoid this problem, always use the following statements when a program routinely creates and reuses a text file:*

```
200 PRINT CHR$(4);"OPEN TEXTFILE"
210 PRINT CHR$(4);"DELETE TEXTFILE"
220 PRINT CHR$(4);"OPEN TEXTFILE"
```

This first opens the old file, deletes it, and then opens it again. You can't just delete it because this will return an error—and stop your program—if you're using a diskette on which that file hasn't already been created.

This is a bit cumbersome, and we've heard that Apple's new operating system (ProDOS, to be released later this year) will address this problem, as well as some of the other Apple idiosyncracies

Consistency Problems

After a lot of soul searching, I finally managed to justify the purchase of an APPLE IIe. Since this purchase, I've spent many enjoyable hours gaming and hacking. Since the editing facilities on the APPLE IIe don't have total screen control, one of my first purchases was an editing utility—the *Global Program Line Editor*. While using this utility, I've noticed that my programs don't produce consistent results. Is there something I don't know?

Annette Weidler
Greenville, OH

The APPLE IIe is a very flexible computer. To really understand this flexibility, you must know how your software is affecting various memory locations. Some utilities, such as the Global Program Line Editor, *modify locations in the hardware I/O address area. For instance, if your program reads an input from the keyboard, APPLESOFT will add 128 to the ASCII code before placing this value in location −16384. However, certain utilities (such as the editor you use) interfere with this process. The result is that memory location −16384 contains the ASCII code for the keyboard input without the addition of 128. Try running your program immediately after booting APPLE DOS. If the problem was caused by an editing program, your program will operate as it should.*

Temporary Amnesia

One of your readers in Texas notified me of a problem with my program, *Cash Flow* (August, 1983), of which we were both unaware: The program will not run without the memory expansion card. Unless the card is inserted, an error message (MEMORY FULL IN 150) is generated.

The program was composed on my machine (which has a memory card) and as it was not a long program, the issue never arose. I suppose that the machine on which it was edited was also so equipped.

I have attempted getting around the problem through conserving memory (CALL FILES(1); omitting REMs; shortening variable names; allowing only 10 entries instead of 16) all to no avail.

I think, therefore, that you should mention that the article requires memory expansion.

Happily, I have heard this comment from only one reader (the rest called or wrote to express their delight). Hopefully the universe of those with Extended BASIC and disk drives but without memory expansion is small.

Thanks for an excellent job of editing and a typo-less printing job on the program. Should I find the time to do a third article, I will be in touch.

Joel S. Moskowitz
Rancho Cordova, CA

Thanks for your letter, Joel. You're right—we inadvertently edited your program on a machine with a memory expansion card, and the fact that memory expansion was required *escaped us. We hope this didn't inconvenience our readers unduly. At the moment, however, we are investigating the possibility of program compaction as well, so that perhaps more TI users can take advantage of your excellent program.*

HCM

CP/M®

PROCESSOR FOR THE TEXAS INSTRUMENTS 99/4A

NOW YOUR TI-99/4A REALLY MEANS BUSINESS!

Morning Star Software introduces the CP/M Processor for the TI-99/4A. Now the advantages of the most accepted business programs are yours. With the CP/M Processor, your TI-99/4A is ready to enter the business world.

The Morning Star Software CP/M Processor fits into your peripheral expansion system like any expansion card. It's as simple as that. Once the CP/M Processor is activated, you have all the advantages of two processors: The CP/M Processor handles all the program computations, while the TI Processor channels the information to the screen, disks and printers. This gives you the flexibility you've always wanted.

CP/M is the industry standard business format for computers today. CP/M offers you the advantage of working with programs that were designed to meet the needs of the professional business world. Powerful commands, time saving functions and proven methods are the result.

The CP/M Processor by Morning Star is built to take advantage of both the TI computer and CP/M technology. The CP/M Processor contains a 5MHz 8085 CPU which, with the 16-bit TI-9900 CPU, translates to tremendous processing power. Your TI-99/4A combined with the CP/M Processor becomes a true business tool at a fraction of the expense of a new computer system.

- ***Needs no modifications to the TI-99/4A system.***
- ***64K RAM, 8K ROM.***
- ***Makes additional use of the TI 32K memory expansion.***
- ***Supplemental installation and usage manual included.***
- ***Contains sufficient memory for virtually all programs.***
- ***Compatible with TI-99/4A storage diskettes.***
- ***Provides 60K transient program area.***
- ***Fully burned-in and test cycled to ensure reliability.***
- ***Fully shielded and protective metal enclosure.***
- ***Ultra-reliable gold connector assembly.***
- ***No chip sockets for better reliability.***
- ***90-day warranty.***
- ***CP/M-80 2.2 disk-based operating system included.***
- ***Digital Research™ CP/M user's manual included.***
- ***Access to more than 10,000 commercially available software packages.***

THE MORNING STAR CP/M PROCESSOR

$595

TO ORDER: Check, Money Order, Master Card, VISA, and C.O.D. orders accepted. Add $5.00 shipping and handling. Dealer inquiries welcomed.

COMPATIBLE CP/M® SOFTWARE

As Morning Star reviews CP/M packages, we will continue to update our software selection. The programs we offer are the products which demonstrate outstanding values. They have met our criteria of being extremely user friendly, comprehensive in content, plus make the best use of the CP/M card's capabilities. They also represent the tasks most needed to be covered by a business computer.

SUPERWRITER™ by Sorcim

This is the word processing program you've been waiting for. Designed to be easily understood in ten minutes with straightforward language and screen prompts. Written specifically for the business person with word processing requirements. It provides all key features of a professional text editor. It incorporates form letters, report generation, list maintenance and even a spelling checker, the award winning SpellGuard™. Similar features cost as much as $900. SuperWriter™ only **$295**

PERSONAL PEARL™ Data-Base Manager

Now easily create data systems for business, accounting, marketing, office management, education, and more. A complete user's manual, tutorial disk and screen menu guides even novices through the most sophisticated data base manager available. Unlimited files, forms and reports can be created. ***Info World*** *reported "Personal Pearl offers you the strongest, easiest to use data-base manager I have seen at any price".* **$295**

THE RANDOM HOUSE PROOF READER

20,000 word Random House dictionary. Displays spelling errors in context and can make spelling suggestions. You edit the word. Corrections are automatically checked and substituted into the document. File new words into the dictionary with a keystroke. **$50**

CBASIC® INTERPRETER

The language of choice for many CP/M compatible software packages. This brings you a variety of established and time tested applications and allows you to write your own CP/M compatible programs **$150**

BUSINESS MASTER® PLUS

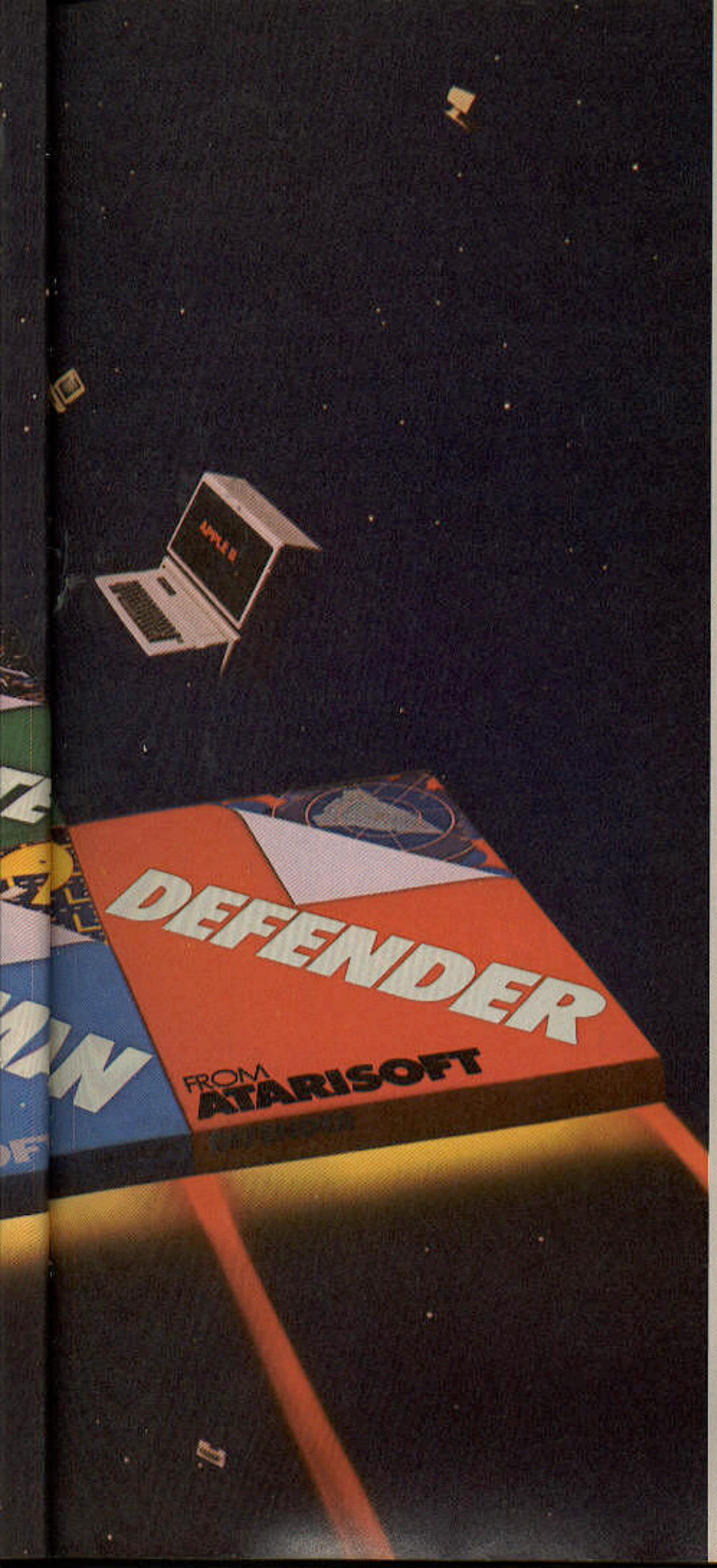
A comprehensive business package designed to meet the needs of leading businesses. The package of 10 diskettes includes General Ledger, Payroll, Fixed Assets Accounting, Accounts Receivable, Accounts Payable, Inventory, and Mailing List programs. Requires CBASIC Interpreter. **$289**

MORNING STAR SOFTWARE

4325 S.W. 109th Ave., Beaverton, Oregon 97005. Toll-Free 1-800-824-2412, in Oregon 503/646-4695, TEX NET NO. TI3416

Peripheral Requirements: Peripheral Expansion System, disk, disk controller 32K RAM card.

NOW YOUR COMPUTER FITS THE ORIGINAL ARCADE HITS.



Now the excitement of original arcade graphics and sound effects comes home to your computer.

Introducing ATARISOFT.™ A new source for computer software.

If you own a Commodore VIC 20 or 64, a Texas Instruments 99/4A, an IBM or an Apple II, you can play the original arcade hits.

DONKEY KONG by Nintendo, CENTIPEDE,™ PAC-MAN, DEFENDER, ROBOTRON: 2084, STARGATE and DIG DUG. (On the TI 99/4A you can also play Protector II, Shamus, Picnic Paranoia and Super Storm.)

So, start playing the original hits on your computer.

Only from ATARISOFT.

Some games also available on ColecoVision and Intellivision.

ATARISOFT™

Now your computer fits the arcade hits.

Complete this coupon and we'll keep you up to date on the newest hits from ATARISOFT.™

Name _____

Address _____

City _____ State _____ Zip _____

Telephone _____

PRODUCT OWNED: (Check one)

1 ☐ TI-99/4A
2 ☐ IBM PC
3 ☐ Commodore 64
4 ☐ ColecoVision
5 ☐ Commodore Vic 20
6 ☐ Intellivision
7 ☐ Apple II
8 ☐ _____

Mail to:
Atari, Inc., P.O. Box 2943,
So. San Francisco, CA 94080.

ASM 35

TI-99/4A

THE EVERYTHING BOOK

For The TEXAS INSTRUMENTS 99/4A HOME COMPUTER

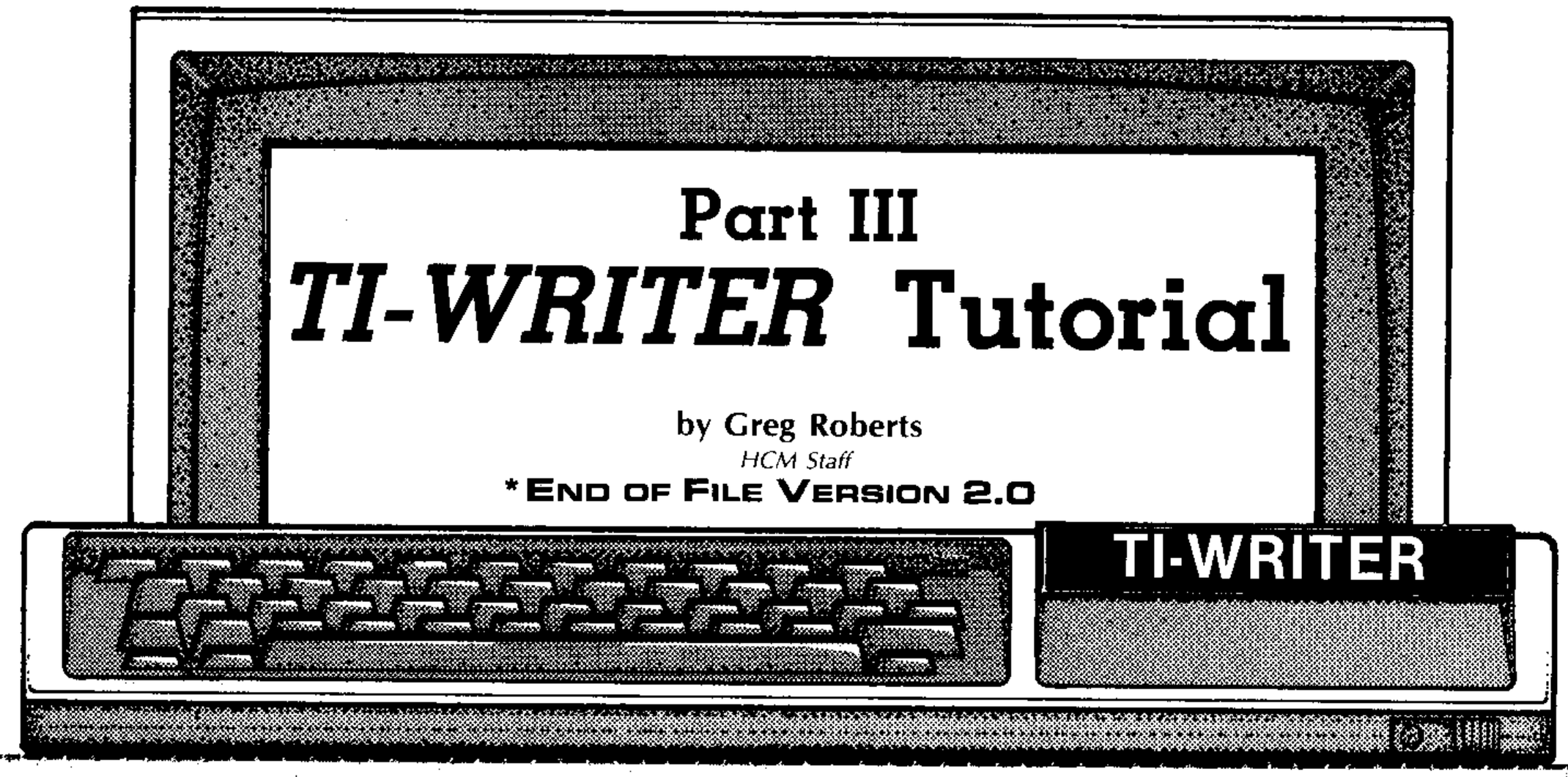
- *HOW-TO Information on expanding your system*
- *GAMES - The newest, the best, the most*
- *GIANT Software Directory over 500 descriptions*
- *Peripherals and Accessories Hundreds of items: printers, joysticks, disk drives, plotters, modems, monitors & much more!*
- *DISCOUNTS! And Easy Order Info*

GET IT FREE!

Just Call Us Toll-Free

1-800-348-2778

Or send your name and address to:
TENEX, Box 6578, South Bend, IN 46660
(219) 277-7726



Part III
TI-WRITER Tutorial

by Greg Roberts
HCM Staff

***END OF FILE VERSION 2.0**

This month we cover the remaining "frills" in the *TI-WRITER* word processing program. These features may not be essential to your writing, but they certainly can speed it along.

You may already have some of your writings on diskette. So let's start out by loading a file. Press [FCTN] [9], type LF and [ENTER]. Note that you do not have to select 'Files' from the command line. Just go directly to LF and enter DSK1. plus the name of the file. You don't have an old file to play with? *Then just* put a few lines of gibberish on the screen so that we can get started.

Home Cursor [CTRL] [L] puts the cursor in the upper left-hand corner of the screen, saving you the trouble of moving it with the arrow keys. An even more specific operation, [CTRL] [V], moves the cursor to the beginning of the line you are working on. A similar time-saver is Last Paragraph or [CTRL] [H]. It shoots the cursor back to the first letter of the previous paragraph.

More minor tactics: Duplicate Line lets you reproduce the line appearing above your cursor. Press [CTRL] [5]. Delete Line, [FCTN] [3] or [CTRL] [N] , takes out an entire line and closes up the text. You can even Delete End of Line by using [CTRL][K] to erase the character under your cursor, and everything to the right of it.

By this time you may be questioning the value of learning such restricted plays. After all, when would you ever have to copy the exact line you've just written? It is, admittedly, an obscure device belonging on the same shelf with electric carving knives and patio bug zappers. The only place I can think of using Duplicate Line would be in song writing. I suppose the Beatles might have found a use for it in setting down that immortal line

"I get by with a little help from my friends."

That is, they repeat the line with just a minor variation. *Duplicate Line* would have saved them the trouble of typing it twice. Some of the other operations will, I hope, more readily show their value. At first some of these may not seem as convenient as using the arrow keys, but eventually they will prove their worth. If you practice the commands, you will soon be using all of them without having to refer to the strip at the top of the keyboard.

Crazyquilt Commands

Some of the difficulty in learning these operations comes from their haphazard arrangement on the keyboard. They may be triggered by the Control key, the Function key, with letters or numbers. Many operations can be activated by two combinations of keys, adding to the confusion. It is unfortunate that some of the operations aren't tied to keys that would help in memorizing them—for example, [CTRL] [H] for Home Cursor, or [CTRL] [O] for Oops! But such is not the case with the present version of *TI-WRITER*. Nevertheless, nearly all the operations described in the manual can be time-savers, once you get them under your thumb.

Several of these operations are so close to the Delete Line feature ([FCTN][3]) that they can easily cause you to blank out a line of text. But you don't necessarily have to retype the lost line. If you press [CTRL] [1] or [CTRL] [Z], otherwise known as Oops!, you regain the line. You must, however, press Oops! immediately after the mistake occurs. If you have already started another line, the program cannot recover the lost line.

"You are forced to retype DSK1. and the name for a deletion, to give you time to think before you plunge that masterwork into oblivion."

Now on to large-scale operations. DeleteFile certainly is not in the frill category. This command lets you clear your diskette of unwanted programs. When you press [FCTN] [9], then DF, note that the program does not automatically display the filename. You are forced to retype DSK1. and the name for a deletion, to give you time to think before you plunge that masterwork into oblivion.

The Purge command is like the Delete command, but without its finality. Use [FCTN] [9], then P, to clear the contents of the screen after you have saved the material on diskette. You can then conveniently start your next file. This one also carries a safeguard: The program asks PURGE FILE ARE YOU SURE (YES OR NO)? It gives you time to make sure you have entered SaveFile before you clear your work from the screen. Beyond this safeguard, you get yet another chance to recoup from having accidentally purged a file. RecoverEdit is a command that comes from pressing [FCTN] [9] followed by RE. You must access this command as soon as possible after accidentally purging your file, or the program will not be able to recover it.

But even with these safety features, it is easy to destroy your work and not realize it until much later. For example, you purge a file, start a new one, then go to save the second file with the SF command. Here it is very easy to unwittingly hit

Continued on p. 30

TI-Writer Tutorial

SF without changing the file name that automatically appears on the screen—the file name of the previous work already purged. The disk drive spins, and it's *adios Carlos*, as a two-line mailing label takes the place of a 20,000 word thesis on *Frustration-Induced Trauma Among Computer Operators*. This problem can be avoided only by working cautiously with Purge and Save commands, and probably by suffering a few devastating losses of data.

With all this Deleting, Purging, and RecoverEditing, you may not always be sure of the contents of a file. To find out exactly what you have on a diskette, push [FCTN] [9] for the command line, SD for ShowDirectory, then [ENTER]. The program will ask you for the disk number (a simple numeral is all that's required) and then list all files in alphabetical order, in addition to telling you how much space is left on your diskette (0-358 sectors). Make sure your *TI-WRITER* Command Cartridge is in the cartridge port when you ShowDirectory, or your program will crash.

Wait, you say. What an absurdly obvious remark. Of course you have to keep your Command Cartridge in the machine in order to use the program. Surprise! Such is not the case. Once you set up your word processing program to the point of entering TEXT EDITOR from the main menu, you can pull out the cartridge, and *TI-WRITER* just keeps on breathing. You can then lend it to someone across the office, classroom or street. With just your user diskette you can load, save, and print files. Do not, however, try to ShowDirectory or use RecoverEdit without the cartridge.

Once you learn these refinements, you will be in control of all the editing power of the *TI-WRITER* word processor. But that is just the beginning. Next time we will explore some of the printing and formating options available in this program.

HCM

Uncle Larry . . . from p. 17

IBM PCjr

```
980 DATA "O3 T120 AF#DDEF# T60 G T120 EAF#DDEF#GF#EAF#DDEF# T60 G T120 EF#ED T60 E T120"
990 DATA "T60 O2 F#>D T120 F#DEF#GE T60 D T120 F#DEF#GEDF#A>DC#<BAGF#GABAGF#E T60 D T120 F#DEF#GE T60 D T120 F#DEF#GEDF#A>DC#<BAGF#GEF# T60 D"
1000 DATA "T120 O2 AGF#A>D<AF#A>D<AGB>D<BGB>D<BAB>D<BAB>C#DEF#GEF#GEF#DC#<BAF#A>D<AF#A>D<AGB>D<BGB>D<BAB>C#DEF#GEF#GEF# T40 D"
1010 DATA "T240 O3 DC T120<BGGDGGBGB>DC<B>C<AADAA>C<A>CEDC<BGGDGGBGB>DC<B>C<B>C<A>DC<BGG T60 G"
1020 DATA "T240 O3 GA T120 BGGDGGBGBBAGAF#F#DF#F#F#DF#AGF#C#GGDGGC<BA>DC<BGG T60 G"
1030 DATA "T120 O2 B T40 >D T120 ED <B>G<B>C<B>G<B>A<AAB T40 >D T120 ED<B>G<BA>C<BABGGBB>DDED<B>G<BA>C<BABGG"
1040 DATA "O2 B>DGBGAGBGDGBGAGEGDGBGAGBGDEDC<BGGEDGBGAGBGDGBGAGEGBAGF#GF#EDEABGG"
1050 DATA "T171 O2 DEF# T60 G T120 BG>C<GBG>C<GBGADEF# T60 G T120 BF>C<GBGA>DC<AG"
1060 DATA "O3 T171 DEF# T60 G T171 AGF#G T120 D<B>D T60 G T171 AGF# T120 GABA T60 G T171 AGF# T120 GD<B>DEGF#A T60 G"
1070 END
```

HCM

Uncle Larry . . . from p. 15

TI-99/4A

```
280 NEXT X
290 REM  PRINT MENU
300 FOR X=1 TO 17 STEP 8
310 CALL HCHAR(X,1,95,256)
320 NEXT X
330 RESTORE
340 CALL HCHAR(24,1,32,32)
350 FOR X=1 TO 3
360 IF X<>2 THEN 390
370 PRINT TAB(5);"LARRY'S FIDDLE TUNES"
380 GOTO 400
390 PRINT
400 CALL HCHAR(23,1,95,4)
410 CALL HCHAR(23,29,95,4)
420 NEXT X
430 FOR X=1 TO 14
440 READ T$,M$
450 A$(X)=M$
460 PRINT TAB(3);T$
470 CALL HCHAR(23,1,95,4)
480 CALL HCHAR(23,29,95,4)
490 NEXT X
500 PRINT ::
510 REM  SELECT FROM MENU
520 M$="SELECTION:     KEY:   1ST NOTE:"
530 C=2
540 GOSUB 2150
550 INPUT "    ":S
560 IF (S<1)+(S>13)THEN 550
570 M$=A$(S)
580 C=16
590 GOSUB 2150
600 X=0
610 ON S GOSUB 640,720,800,880,960,1040,1120,1200,1280,1360,1460,1970,2050
620 GOTO 290
630 REM  SUBROUTINES FOR THE 10 TUNES
640 DATA "  (1) SOLDIER'S JOY",D      F#ON 3
650 RESTORE 2340
660 GOSUB 2210
670 IF X<2 THEN 650
680 RESTORE 2410
690 GOSUB 2210
700 IF X<4 THEN 680
710 RETURN
720 DATA "  (2) ARKANSAS TRAVELLER",DA ON 4
730 RESTORE 2470
740 GOSUB 2210
750 IF X<2 THEN 730
760 RESTORE 2520
770 GOSUB 2210
780 IF X<4 THEN 760
790 RETURN
800 DATA "  (3) CINCINNATI HORNPIPE",D D/OPEN 3
810 RESTORE 2570
820 GOSUB 2210
830 IF X<2 THEN 810
840 RESTORE 2630
850 GOSUB 2210
860 IF X<4 THEN 840
870 RETURN
880 DATA "  (4) POP GOES THE WEASEL",G  G ON 3
890 RESTORE 2680
900 GOSUB 2210
```

TI-99/4A

```
910 IF X<2 THEN 890
920 RESTORE 2720
930 GOSUB 2210
940 IF X<4 THEN 920
950 RETURN
960 DATA " (5) GARRY OWEN",G     G ON1
970 RESTORE 2760
980 GOSUB 2210
990 IF X<2 THEN 970
1000 RESTORE 2810
1010 GOSUB 2210
1020 IF X<4 THEN 1000
1030 RETURN
1040 DATA " (6) COCK AND HEN",G     B ON 2
1050 RESTORE 2860
1060 GOSUB 2210
1070 IF X<2 THEN 1050
1080 RESTORE 2900
1090 GOSUB 2210
1100 IF X<4 THEN 1080
1110 RETURN
1120 DATA " (7) TOM & JERRY REEL",D     A/OPEN 2
1130 RESTORE 2940
1140 GOSUB 2210
1150 IF X<2 THEN 1130
1160 RESTORE 3000
1170 GOSUB 2210
1180 IF X<4 THEN 1160
1190 RETURN
1200 DATA " (8) IRISH WASHERWOMAN",G     D ON 2
1210 RESTORE 3060
1220 GOSUB 2210
1230 IF X<2 THEN 1210
1240 RESTORE 3110
1250 GOSUB 2210
1260 IF X<4 THEN 1240
1270 RETURN
1280 DATA " (9) MC DONALD'S",G     B ON 2
1290 RESTORE 3160
1300 GOSUB 2210
1310 IF X<2 THEN 1290
1320 RESTORE 3210
1330 GOSUB 2210
1340 IF X<4 THEN 1320
1350 RETURN
1360 DATA "(10) TWO FORTY REEL",G     D/OPEN 3
1370 RESTORE 3250
1380 GOSUB 2210
1390 IF X<2 THEN 1370
1400 RESTORE 3290
1410 GOSUB 2210
1420 IF X<4 THEN 1400
1430 RETURN
1440 DATA "",""
1450 REM
1460 DATA (11) tune the fiddle,""
1470 REM
1480 CALL CLEAR
1490 PRINT " PRESS NUMBER"::" OF STRING TO"::" BE TUNED."::::
1500 PRINT " PRESS (0)"::" FOR MENU."::::
1510 CALL VCHAR(1,22,33,96)
1520 CALL VCHAR(1,22,44,10)
1530 CALL VCHAR(1,23,44,6)
1540 CALL VCHAR(1,24,44,4)
1550 CALL VCHAR(1,25,44,8)
1560 CALL HCHAR(14,22,44,4)
1570 PRINT TAB(20);"4321"
1580 CALL KEY(0,K,S)
1590 IF (S=0)+(K<48)+(K>52)THEN 1580
1600 ON (K-47)GOSUB 1620,1630,1670,1710,1750
1610 GOTO 1580
1620 GOTO 300
1630 N=69
1640 F=659
1650 R=7
1660 GOTO 1780
1670 N=65
1680 F=440
1690 R=3
1700 GOTO 1780
1710 N=68
1720 F=294
1730 R=5
1740 GOTO 1780
1750 N=71
1760 F=196
1770 R=9
1780 CALL SOUND(-SP*20,F,1)
1790 IF K>50 THEN 1840
1800 C=27
1810 GOSUB 1880
1820 CALL HCHAR(R-2,C-1,44)
1830 GOTO 1940
1840 C=18
1850 GOSUB 1880
1860 CALL HCHAR(R-2,C+3,44)
1870 GOTO 1940
1880 REM DISPLAY PEG
1890 FOR X=1 TO 3
```

Continued on p. 32

PROGRAMS . . .

$2 EACH ! !

WHY PAY MORE FOR QUALITY PROGRAMS?

WE ARE THE LOWEST COST ORGANIZATION

SPECIALIZING IN SOFTWARE FOR THE 99/4A . . .

IN THE WORLD !

The 99/4A Program Exchange, Inc. is an international user's group specializing in the distribution of owner/user written software. Our purpose is to bring a tremendous range of quality software written by owners/users to 99/4A owners at the lowest possible cost.

JOIN OUR USERS GROUP !

★ ONE-TIME MEMBERSHIP FEE

The 99/4A Program Exchange has a low ONE-TIME membership fee of only $10.00. What's more, we'll mail out membership materials the very same day that we receive your membership fee!

★INCREDIBLY LOW PRICES

Members may purchase ANY PROGRAM from our owner/user written software catalog for ONLY 2^{00} per program—no tricks, no gimmicks. Programs available on CASSETTE or DISKETTE.

★QUALITY AND DIVERSITY

Our software library currently has over 1000 programs for 99/4A users! Programs are categorized as follows: GAMES, DEMONSTRATIONS, EDUCATION, BUSINESS, HOME, APPLICATION and UTILITY.

★LITERATURE

The 99/4A Program Exchange publishes highly informative newsletters containing hints, program reviews, problem solvers and more.

★SOMETHING FOR EVERYONE

There is no programming experience required to join. However, if you do program and want to submit a program to our software library, we have a generous 5 for 1 program exchange! We even accept MASTERCARD and VISA.

THE 99/4A PROGRAM EXCHANGE INTERNATIONAL USERS GROUP

WE ARE HERE TO STAY !

I understand that I will receive The 99/4A Program Exchange, Inc. software catalog containing quality programs for $2 per program, a current newsletter and an official software order-form.

Send your $10.00 Membership Fee.

☐ CHECK OR MONEY ORDER ☐ VISA or MASTERCARD

Card # _____ Exp: _____

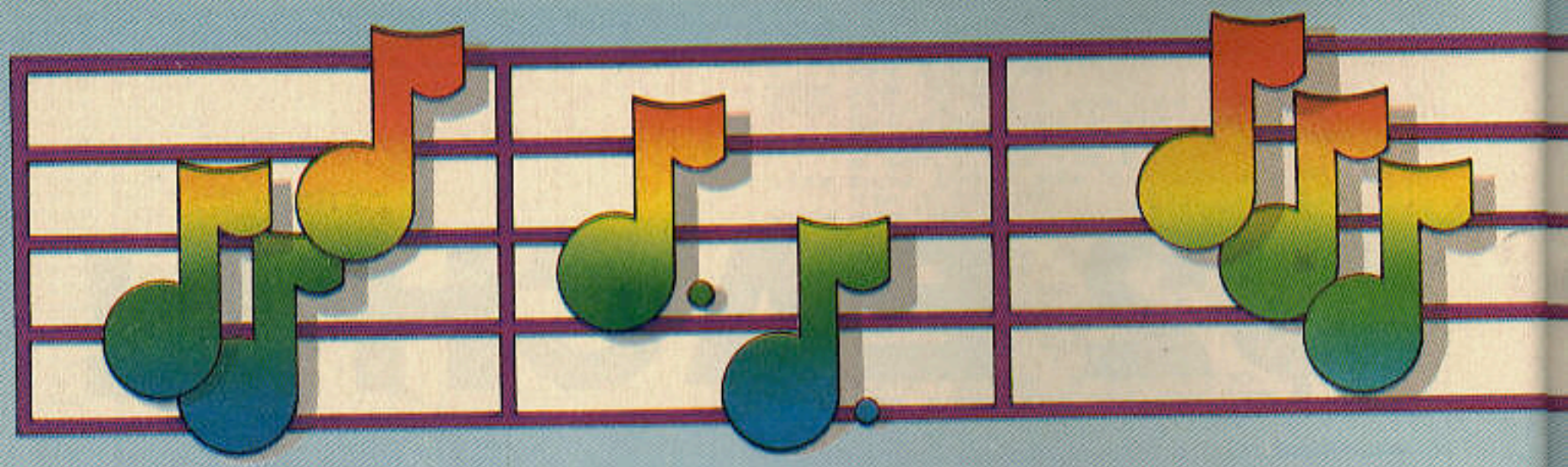
Name: _____

Address: _____

City _____

State _____ Zip _____

THE 99/4A PROGRAM EXCHANGE
P.O. BOX 3242, TORRANCE, CA 90510



I Write the Songs:

Electronic Sheet Music

Program by Carol Burris
and the HCM Staff

Like many Home Computer users, I was amazed to discover the 99/4A's extensive sound, color and graphics capabilities. After finding a formula to generate a musical scale in Herbert Peckham's book, *Programming BASIC*,* I decided to try some musical programming. I wanted a program to not only play music, but to display the notes on a staff as well. The *magic* in the title of my resulting program refers to the fact that in writing it, I really learned BASIC.

Music Magic is an easy-to-use Extended BASIC program which lets you play, display and save music. Songs can be composed of notes or chords within a two-octave range (A below middle C to high G# [no Ab] above high C) and can be up to 43 notes or chords in length.

Getting Started

After a short introduction, the first menu appears. You may choose either 1) Recorded Song—which will load and play a song previously saved on tape or diskette—or 2) New Song. Naturally, the first time you use *Music Magic* you will press 2. A second menu then appears which asks you to select either Single Note Entry, Two-Part Harmony, or Three-Part Harmony. After you make this choice, a third menu asks if the key you're writing in has any sharps. If you type N (No), the next menu will appear, asking if your piece has any flats. If it does, after you type Y (Yes), the next screen will ask, HOW MANY? You need to tell the computer the letter name of each flat (or sharp). If we were writing in the key of F we would tell the computer that we have one flat, B flat. In this program you will not be able to cancel the flat and play a B natural, because there is no provision for accidentals (sharps, flats, and natural signs that appear next to the note and not in the key signature).

When your choices are made, you can begin entering your song. One staff of music appears, ready to receive your notes, and the message NOTE 1 prompts your first entry. After each note is entered, it is played and displayed on the staff.

If you decide that the note (or chord) is not what you wanted, press "R", and the computer will allow you to enter it again. Throughout the program, whether you have just selected the key signature, or have selected Recorded Song instead of New Song, entering "R" will let you backtrack.

The staff displays approximately 13 notes, and then a new staff will begin.

Our Song

In order to get a feel for what this program can do for you, let's arrange the popular carol, "Joy to the World" for the 99/4A using *Music Magic* so you can pick up music writing hints along the way. We will write our version in the key of C—with no sharps and no flats.

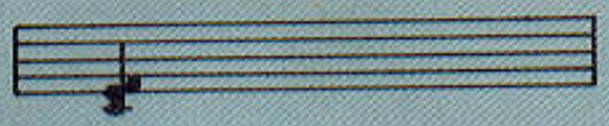
Entering single notes (without harmony) is a simple matter of typing in the letter name of the note. In the two- or three-part option, things get a little tricky. If you want a single note to appear on the staff you will need to enter the same note two or three times in succession. You must also remember to enter the lowest note of the dyad or triad first.

For example, to enter the first dyad in our song properly, you should type the E first and then the C. To indicate that this is the C above Middle C, however, you must also type [FCTN][O] for the single quote. Your second entry will be displayed as C' on the screen, and the note will be printed in the third space from the bottom.

This takes care of the first note in our song; next comes the prompt to enter the duration of the note. An eighth note (♪) is the shortest note, and the whole note (𝅝) is the longest note available. Durations are entered by typing in a number according to the note you want from the chart below.

NOTE VALUE:		TYPE IN:
Eighth	♪	1
Quarter	♩	2
Half	𝅗𝅥	4
Whole	𝅝	8

Before you put the rest of "Joy to the World" on the staff, here are some more hints about entering notes. If you put in a three-part combination like this:



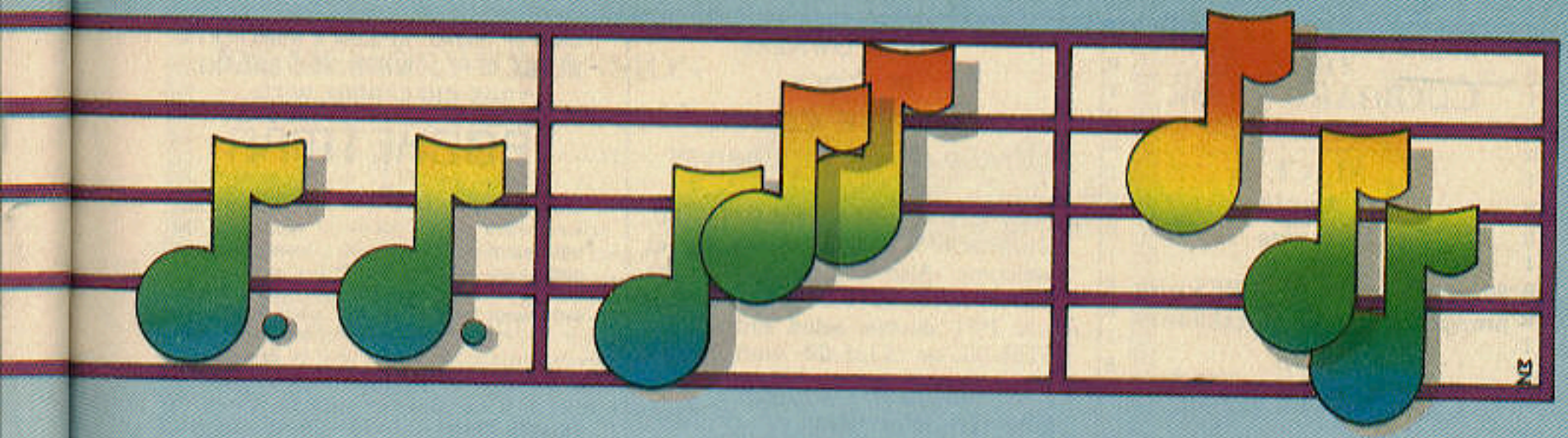
it is counted as two notes, not one. The note that looks like it is on the wrong side of the stem (but isn't!) counts as one extra note space. Therefore, one less space is available for other notes in your song. In other words, instead of having room for 43 notes you will, after entering this combination as your first note, have 41 note spaces left. If you have chosen 2- or 3-part harmony and wish to play a single note, simply enter the same note for all of the note prompts for that space. To enter a 2-note combination in 3-part mode, simply type in the higher note of the two, twice. For example, E, C', C' results in E and C' being printed and played.

You can keep track of song length with the note counter located in the lower left corner of the screen. If you accidentally exceed 43 notes, the entry stops, and your song thus far is played. To end a song of less than 43 notes, enter Q after the NOTE 1 prompt.

The next menu screen will offer the self-explanatory choices of: 1) PLAY 2) RECORD 3) NEW SONG or 4) EXIT. Now, choose the New Song option and enter all the notes for the first seven measures

Continued on p. 36

*Texas Instruments and McGraw Hill, 1979.



Just Assemble Melody:

Music in Mini Memory

by Cleon Chapen

For the past year or so I've been writing Assembly Language programs for my TI, and after tackling such sticky subjects as bit-mode graphics, sprites, and data files, the time has come to face the music.

As a musician by profession and a compulsive TI devotee by inclination, one would think I could dive right in and make beautiful music. But after taking a long hard look at the Editor/Assembler manual's chapter on sound, I was, quite frankly, intimidated. The detailed bit-picking needed to format correct sound lists seemed like an awful lot of work. When you create sound data, you are writing machine language programs for the sound processor to execute. The natural thing to do, then, is to write a program that will remember all those details for you. After all, isn't that what computers are for?

This program does for the sound processor what the *Line-by-Line Assembler* does for the CPU: translates a symbolic assembly code (in this case, very much like TI BASIC CALL SOUND statements) into a machine-executable form. In addition, it will (if you choose) load and run a short machine language subprogram called PLAY which executes the sound list through TI BASIC.

If you haven't already typed the program in, I'll wait here for a few minutes while you do so . . . Be extra careful in typing the DATA statements marked MACHINE ROUTINE. A code in the *Music Assembler* checks this data, but some errors may not be detected.

Before you run the program, type CALL INIT to clear the contents of the Mini Memory so that plenty of memory is available for the sound lists. The program will begin putting in the lists at location >701C (the FFM pointer). If there is not enough room for the sound list and the PLAY subprogram to live together, the *Music Assembler* will issue a ****MEMORY FULL**** error and stop.

Type RUN, and *Music Assembler* will initialize while the title screen is displayed. An initial prompt will be displayed, telling you how to stop the assembler. Enter sound data after each "?" prompt. To assemble your first sound list, consider the following TI BASIC sound statement:

Example 1

```
CALL SOUND(1000,440,15,554,15,659,15,-5,20)
```

Notice that the code between parentheses consists of nine fields separated by commas, each of which has a certain meaning depending on its position. *Music Assembler's* version of this statement looks like this:

Example 2

```
1000.440.15 .554.15.659.15. -5.20.
```

The meaning of each number remains the same, but the statement is slightly more compact. Since TI BASIC has an aversion to the comma in INPUT statements, we use the period instead. The only other difference is that each statement must end with a period. If you forget, the program will make you do it over again. Politely.

Here is another version of the same statement which will produce the same machine code:

Sound List 1

```
(1) 1000.A1.15.C#1.15.E2.15. -5.20.
```

You may specify notes by name as well as frequency. There are several advantages to this. The most obvious is that you don't have to look up frequencies any more. In addition, the frequencies that the program uses are slightly more accurate than whole numbers, with improved intonation for the ultra-fastidious ear (like mine).

The note names that *Music Assembler* recognizes correspond to those in the Editor/Assembler manual (section 20.3), minus the generator number. They extend from A0 (110) to F6(5587.65). To simplify the search routine, only sharps are used. To specify Bb3, for instance, use A#3 instead. Of course, you may specify notes by the range of frequencies allowable in TI BASIC.

Type in Sound List 1 above, then press [ENTER]. *Music Assembler* will assemble the line, checking to be sure all the values are within the proper ranges, then load the data into memory. It will then ask for your next line with the "?" prompt. If you discover that you mistyped something (3000 instead of 1000, for example) just type REDO and press [ENTER]. The *Music Assembler* will back up one line, forget what you just typed, and ask you to re-enter the line. If you type in the wrong designation for a note—FI instead of F1, for instance—it will prompt you to reenter the *note* designation only—F1—not the whole sound list. Since your first sound list should be a little more interesting, try entering these lines:

Sound List 1 (continued)

```
(2) 1000.A1.15.D2.15.F#2.15.
(3) 1000.A1.15.D2.15.G2.15.
(4) 1000.D2.15.F#2.15.A2.15.
```

When the last line is finished assembling, stop the process by pressing [ENTER]. The *Music Assembler* will add some code to the end of your list to make the sound processor shut off when it finishes the list, and reset the First Free Memory pointer to the next word boundary (even-numbered address) after the sound list. This is where the next list will be loaded (assuming you do not issue a CALL INIT command) the next time you run the program. This stacking of lists can be very useful if you are assembling a number of them. You can locate them anywhere you want by presetting >701C via the EASY BUG program.

If you want to hear it played (an understandable desire), type Y in answer to the question PLAY LIST? (Y/N). The *Music Assembler* will then load the PLAY subprogram and execute it.

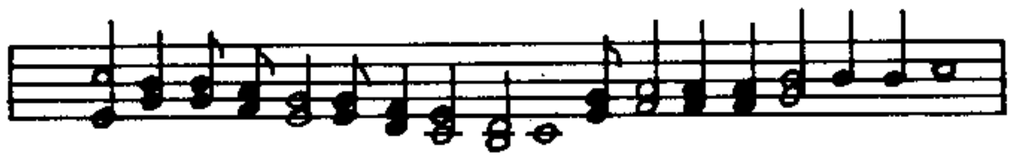
After the music is over, press [ENTER], and *Music Assembler* will tell you (in decimal and hexadecimal notation) where the sound

Continued on p. 37

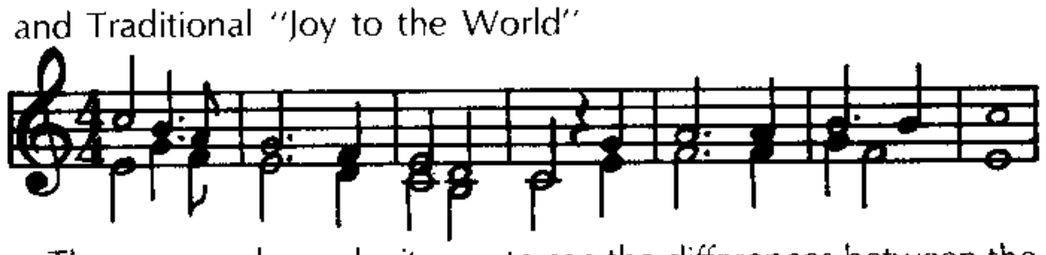
I Write the Songs . . . from p. 34

of "Joy to the World". We can compare the electronic sheet music version with the way it looks on traditional sheet music.

Music Magic "Joy to the World"



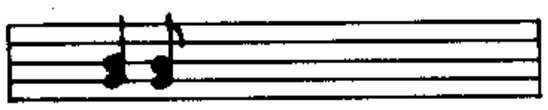
and Traditional "Joy to the World"



These examples make it easy to see the differences between the two notations. In the *Music Magic* version:

1. There are no measure bars.
2. The notes are not spaced on the staff according to their durations.
3. All of the stems go up and are on the right-hand side of the notes.
4. There are no dotted notes or rests.

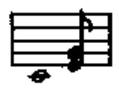
The first three points are merely cosmetic differences, but not having dotted notes or rests can be a serious musical limitation. Actually, you can compensate for their absence by using some simple values in music. All of the notes have a relative duration. That is, an eighth note always lasts half as long as a quarter note, which lasts half as long as a half note, and so on. With 4/4 time, in which there are four beats to a measure, a whole note (𝅝) is 4 counts, a half note (𝅗𝅥) is 2 counts, a quarter note (♩) is 1 count, and an eighth note (♪) is a half count. When a dot appears next to a note, it increases that note's duration by one half. A dotted quarter note, then, would last for 1½ counts (in 4/4 time), or as long as a quarter note and an eighth note combined. When you see a dotted note, such as the second one in "Joy to the World" (traditional version), it would "translate" into this:



Rests are handled in a similar way. Each rest has a value that corresponds to a note of equal duration.

An Eighth rest (𝄾) = ♪
A Quarter rest (𝄽) = ♩
A Half rest (𝄼) = 𝅗𝅥
A Whole rest (𝄻) = 𝅝

Our solution to the rest problem was to make the note value before the rest *longer* and the note following it *shorter* than they would be in the traditional version.

So,  becomes 

You will, no doubt, make some interesting innovations of your own as you finish this song and create new ones.

The Record option allows you to put a melody you've written on either cassette or diskette. To get a full 43-note melody on cassette, it will have to be longer than five minutes. If you choose to record on diskette, be sure your diskette is in disk drive one.

Expanding the Program

This kind of program lends itself to personalizing, adapting and modifying. As amazing as the 99/4A is, it is still a microcomputer with the BASIC interpreter. If you ask it to do too much during play and display, your sound will drag.

If you have memory expansion (the program, as is, only leaves about 200 bytes free if the disk system is connected), you could add some other editing features. You could, for example, devise a way to write tunes longer than 43 notes, or a way to handle accidentals. You could even add a treble clef graphic to make your *Music Magic* versions look more authentic. If you are successful with any of these innovations let us all know in a Letter-to-the-Editor! *Music Magic* is another example of what can happen when you lend your inventive mind to your computer—the two of you can make beautiful music together.

MUSIC MAGIC
Explanation of the Program

Line Nos.	
100-170	Program header.
180-400	Redefine characters and set up arrays.
410-430	New or saved song menu.
440-460	Select cassette or disk option to retrieve a song.
470-480	Code for inputting song.
790-910	Song playback and note display.
920-1020	Option menu.
1030-1060	Cassette or disk option for recording a song.
1070-1100	Draw staff subroutine.
1110-1220	Save song subroutine.
1230-1320	Play recorded song subroutine.
1330-1340	Error recovery.
1350-1410	Menu display subroutine.
1420-1570	Sharps and flats subroutine.
1580-1800	Note printing subroutine.
1810-1850	Adjust notes for sharp or flat.
1860	Eliminate deleted note from the screen.

Continued on p. 54

Just Assemble Melody . . . from p. 35

Using the Machine Language Routine

This program was intended to facilitate the construction of sound lists for Assembly Language applications, but it may also be used in conjunction with TI BASIC programs if some care is exercised. Because the sound list will play in its entirety, even while the BASIC interpreter continues, your program can go on to other things while the music plays. However, because TI BASIC has no idea what is going on, there may be unpredictable side effects, especially when BASIC is manipulating strings or displaying graphics on the screen. [See the Editor/Assembler manual, page 312, for more information.—Ed.] Any CALL SOUND statement will stop the execution of a sound list. Here is a sample program which will play a sound list over and over until a key is pressed, then interrupt it (much like the techniques used in TI's *Tombstone City*):

```
100 CALL LINK("PLAY")
110 CALL PEEK(-31794,N)          Is it done?
120 IF N=0 THEN 100              Yes, play it again.
130 CALL KEY(0,K,S)              Key pressed?
140 IF S=0 THEN 110              No, check sound again.
150 CALL SOUND(-1,-1,30)         Stop the sound list.
 .
 . program continues
 .
```

The PEEK statement (line 110) reads a byte at >83CE. This will be zero when the interrupt routine has finished the sound list.

You can save and reload the PLAY subprogram and any sound list you generate with EASYBUG. Be sure to save everything from >7000 to >7FFF. This ensures that you have the entry point to the PLAY subprogram, and can play the sound list using the short program segment above.

Here, in the form used with the *Line-by-Line Assembler*, is an Assembly Language listing of the "PLAY" subprogram that the *Music Assembler* loads:

```
ST    EQU   >837C
      AORG  >7FBE
TA    DATA  0              Any value poked into this address
                           will be used as the address of the
                           sound list.
H1    BYTE  >01
      EVEN

PL    MOV   @TA,R1         Get table length pointer.
      MOV   R1+,R2         R1 is now start of sound list.
      MOV   R2,@>830C      Copy length to GPL parameter
                           address.

      BLWP  @>6018         Get string space routine.
      DATA  >0038          Get address of allocated space.
      MOV   @>831C,R0      Move sound list to VDP RAM.
      BLWP  @>6028

      LIMI  0              Play list
      MOV   R0,@>83CC
      SOCB  @H1,@>83FD
      MOVB  @H1,@>83CE
      LIMI  2
      CLR   @ST            Return to BASIC.
      RT
      TEXT  'PLAY
      DATA  PL
      AORG  >701C
      DATA  >7FF8, >7FF8
      END
```

It would be possible to put the program title in the REF/DEF table in this way because the program begins at >7FBE and uses up all but the last eight bytes of memory. The last two lines set *both* the First *and* Last Free Addresses in Medium Memory to >7FF8. The Last Free Address indicates the start of the REF/DEF table.

list now resides in memory. The first word of the sound list tells its length. This is used by the PLAY subprogram when allocating memory in VDP RAM for the list. It is a good idea to write these numbers down for future reference. To hear the music again, type CALL LINK("PLAY").

If you take a closer look at the sound statements above, you may notice that fields 8 and 9 (the noise specification and volume) are missing from lines 2-4, but that the noise continues. This is because the sound, once started, continues until it is explicitly stopped. The *Music Assembler* assumes that you want the sound of channel X to continue unless you tell it otherwise. TI BASIC, you will remember, works slightly differently, turning off all sound as it finishes each CALL SOUND statement. So a program like this:

```
100 FOR X=1 TO 10
110 CALL SOUND(500,440,2)
120 NEXT X
```

makes a slight ticking sound instead of a continuous tone.

Consider the following revision of Sound List 1:

Sound List 2

(1) 1000.A1.15.C#2.15.E2.25.-5.20.
(2) 1000. . . D2.15.F#2.15.-5.30.
(3) 1000. . . . . G2.15.
(4) 1000.D2.15.F#2.15.A3.15.

Line 1 is unchanged. If you look at line 2 in both lists, you will see two differences. First of all, in Sound List 2 the noise processor is turned off by the specification of the maximum attenuation (30). Since the noise is never again specified in this list, the *Music Assembler* assumes that it should stay off. Secondly, instead of fields 2 and 3 repeating A1.15., as in line 1, the extra periods tell channel 1 to continue what it was doing. The positions of the periods indicate which channels to continue.

Look now at line 3. Which channels change, and which ones stay the same? Remember, two periods are needed to specify a blank field. At line 4, every note changes, so every field is filled except

for the noise fields. Here is another example, this time in two-part counterpoint:

Sound List 3

(1) 2000.C1.8.
(2) 1000. . .A1.8.
(3) 1000.F1.8.
(4) 1000. . .G1.8.
(5) 1000.E1.8.
(6) 2000.F1.8.A1.8.C2.8

Because the top voice (on channel 2) does not enter until line 2, it is not specified. The program ensures that it continues what it was doing before (i.e., nothing). In lines 2 and 4 the lower voice (channel 1) holds through, indicated by the two empty fields. Notice that the upper voice will continue in line 5.

Unlike TI BASIC, Assembly Language does not accept negative frequency values as a means of interrupting a sound list. Instead, you may turn all three channels (plus noise) on and off at will, each independently of the others. For example, Sound List 4 creates a syncopated effect in the middle voice by turning it on and off while the other voices continue.

Sound List 4

(1) 1000.A#3.0.D3.0.F3.0.
(2) 250. . .D3.30.
(3) 250. . .D3.0.
(4) 250. . .D3.30.
(5) 250. . .D#3.0.
(6) 1000.A3.0.C3.0.

The *Music Assembler* can detect most errors, but not all. If a note is specified for a duration of zero, the interrupt routine in the console will take this as the end of your sound list. Any notes currently sounding will continue to sound. Another problem can arise when the attenuation field is blank:

200.C#2. .A#5.4.

This line may assemble, but perhaps not the way you intended.

After a little experimentation with various values in the fields, you will know how to cover the whole range of sound available in TI BASIC. But you will then be able to use the TI-99/4A's sound generator somewhat differently from the way it's used in TI BASIC (see box). This will add yet another dimension to the possibilities your Home Computer offers.

NOTE DESIGNATIONS

Note	Frequency	Note	Frequency	Note	Frequency
A0	110.00	G#2	415.30	G4	1567.98
A#0	116.54	A2	440.00	G#4	1661.22
B0	123.47	A#2	466.16	A4	1760.00
C1	130.81	B2	493.88	A#4	1864.66
C#1	138.59	C3	523.25	B4	1975.53
D1	146.83	C#3	554.37	C5	2093.00
D#1	155.56	D3	587.33	C#5	2217.46
E1	164.81	D3	622.25	D5	2349.32
F1	174.61	E3	659.26	D#5	2489.02
F#1	185.00	F3	698.46	E5	2637.02
G1	196.00	F#3	739.99	F5	2793.83
G#1	207.65	G3	783.99	F#5	2959.96
A1	220.00	G#3	830.61	G5	3135.96
A#1	233.08	A3	880.00	G#5	3322.44
B1	246.94	A#3	932.33	A5	3520.00
C2	261.63	B3	987.77	A#5	3729.31
C#2	277.18	C4	1046.50	B5	3951.07
D2	293.66	C#4	1108.73	C6	4186.01
D#2	311.13	D4	1174.66	C#6	4434.92
E2	329.63	D#4	1244.51	D6	4698.64
F2	349.23	E4	1318.51	D#6	4978.03
F#2	369.99	F4	1396.91	E6	5274.04
G2	392.00	F#4	1479.98	F6	5587.65

Middle C = 262.

Music Assembler
Explanation of the Program

Line nos.	
100-160	Program header.
170-220	Define functions.
230-260	Define arrays.
270-650	Data to define notes.
660-730	Data which defines machine language routine.
740-890	Set up program.
900-1310	Main program loop.
1320-1830	Make sound list.
1840-1970	Reset memory address pointers.
1980-2230	Routine to call machine language routine and play sound list.
2240-3010	Subroutines.
2240-2360	Find note input.
2370-2530	Accept sound list entry from keyboard.
2540-2640	Generate a hexadecimal address.
2650-2840	Check for values out of range.
2850-2890	Out of memory message.
2900-3020	Check machine language routine data.

TI-99/4A

```
100 REM *******************
110 REM * MUSIC ASSEMBLER *
120 REM *******************
130 REM BY CLEON CHAPEN
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.1.1
160 REM TI BASIC, TI MINI MEMORY
170 REM * MINI MEMORY
180 REM * FUNCTIONS
190 REM *
200 DEF H(N)=INT(N/16)
210 DEF L(N)=N-H(N)*16
220 DEF CODE(N)=INT(111860.8/N+.5)
230 REM *
240 REM * ARRAYS
250 REM *
260 DIM SD(8),NO$(68),FRE(68)
270 REM *
280 REM * DATA BLOCK
290 REM *
300 REM * PRE-DEFINED NOTES
310 REM *
320 DATA A#0,116.54,A0,110
330 DATA B0,123.47,C#1,138.59
340 DATA C1,130.81,D#1,155.56
350 DATA D1,146.83,E1,164.81
360 DATA F#1,185,F1,174.61
370 DATA G#1,207.65,G1,196
380 DATA A#1,233.08,A1,220
390 DATA B1,246.94,C#2,277.18
400 DATA C2,261.63,D#2,311.13
410 DATA D2,293.66,E2,329.63
420 DATA F#2,369.99,F2,349.23
430 DATA G#2,415.30,G2,392
440 DATA A#2,466.16,A2,440
450 DATA B2,493.88,C#3,554.37
460 DATA C3,523.25,D#3,622.25
470 DATA D3,587.33,E3,659.26
480 DATA F#3,739.99,F3,698.46
490 DATA G#3,830.61,G3,783.99
500 DATA A#3,932.33,A3,880
510 DATA B3,987.77,C#4,1108.73
520 DATA C4,1046.5,D#4,1244.51
530 DATA D4,1174.66,E4,1318.51
540 DATA F#4,1479.98,F4,1396.91
550 DATA G#4,1661.22,G4,1567.98
560 DATA A#4,1864.66,A4,1760
570 DATA B4,1975.53,C#5,2217.46
```

TI-99/4A

```
580 DATA C5,2093,D#5,2489.02
590 DATA D5,2349.32,E5,2637.02
600 DATA F#5,2959.96,F5,2793.83
610 DATA G#5,3322.44,G5,3135.96
620 DATA A#5,3729.31,A5,3520
630 DATA B5,3951.07,C#6,4434.92
640 DATA C6,4186.01,D#6,4978.03
650 DATA D6,4698.64,E6,5274.04,F6,5587.
65
660 REM *
670 REM * MACHINE ROUTINE
680 REM *
690 DATA 1,0,192,96,127,190,192,177
700 DATA 200,2,131,12,4,32,96,24,0,56,1
92,32,131,28,4,32,96,40,3,0,0,0,200
710 DATA 0,131,204,248,32,127,192,131,2
53,216,32,127,192,131,206
720 DATA 3,0,0,2,4,224,131,124,4,91
730 DATA 80,76,65,89,32,32,127,194
740 REM *
750 REM * SETUP
760 REM *
770 CALL CLEAR
780 PRINT TAB(7);"MUSIC ASSEMBLER":::::
::::
790 FOR I=0 TO 68
800 READ NO$(I),FRE(I)
810 NEXT I
820 GOSUB 2890
830 CALL CLEAR
840 D$="."
850 H$="0123456789ACBDEF"
860 CALL PEEK(28700,H1,L1)
870 FFM=H1*256+L1
880 LC=FFM+2
890 CALL CLEAR
900 REM *
910 REM * MAIN LOOP
920 REM *
930 GOSUB 2400
940 IF FL=99 THEN 1870
950 OPTR=0
960 FOR I=0 TO 8
970 SD(I)=0
980 NEXT I
990 FOR I=0 TO 8
1000 NPTR=POS(I$,D$,OPTR+1)
1010 IF NPTR=0 THEN 1050
```

```
1020 IF OPTR+1=NPTR THEN 1290
1030 S$=SEG$(I$,OPTR+1,NPTR-OPTR-1)
1040 GOTO 1070
1050 S$=SEG$(I$,OPTR+1,5)
1060 IF (S$="")+(S$="     ")THEN 1370
1070 IF I=7 THEN 1130
1080 IF (I=1)+(I=3)+(I=5)THEN 1090 ELSE
1190
1090 IF (ASC(S$)>64)*(ASC(S$)<72)THEN 11
00 ELSE 1120
1100 GOSUB 2290
1110 GOTO 1300
1120 IF (ASC(S$)>47)*(ASC(S$)<58)THEN 11
90
1130 IF SEG$(S$,1,1)<>"-" THEN 1240
1140 FOR J=2 TO LEN(S$)
1150 IF (ASC(SEG$(S$,J,1))>47)*(ASC(SEG$
(S$,J,1))<58)THEN 1170
1160 GOTO 1240
1170 NEXT J
1180 IF VAL(S$)>-9 THEN 1260 ELSE 1240
1190 FOR J=1 TO LEN(S$)
1200 IF (ASC(SEG$(S$,J,1))>47)*(ASC(SEG$
(S$,J,1))<58)THEN 1220
1210 GOTO 1240
1220 NEXT J
1230 GOTO 1260
1240 PRINT :"* BAD VALUE: ";S$:"  TRY AG
AIN..":
1250 GOTO 930
1260 SD(I)=VAL(S$)
1270 GOSUB 2680
1280 GOTO 1300
1290 SD(I)=0
1300 OPTR=NPTR
1310 NEXT I
1320 REM *
1330 REM * MAKE SOUND LIST
1340 REM *
1350 REM * DURATION
1360 REM *
1370 DU=INT(SD(0)*255/4250)
1380 DU$=CHR$(DU-(DU=0))
1390 REM *
1400 REM * SOUND GENERATORS
1410 REM *
1420 G1$=""
1430 G2$=""
1440 G3$=""
1450 NC$=""
1460 NA$=""
1470 IF SD(1)=0 THEN 1510
1480 V=CODE(SD(1))
1490 G1$=CHR$(128+L(V))&CHR$(H(V))
1500 G1$=G1$&CHR$(144+INT(SD(2)/2))
1510 IF SD(3)=0 THEN 1550
1520 V=CODE(SD(3))
1530 G2$=CHR$(160+L(V))&CHR$(H(V))
1540 G2$=G2$&CHR$(176+INT(SD(4)/2))
1550 IF SD(5)=0 THEN 1600
1560 V=CODE(SD(5))
1570 G3$=CHR$(192+L(V))&CHR$(H(V))
1580 G3$=G3$&CHR$(208+INT(SD(6)/2))
1590 REM *
1600 REM * NOISE CONTROL
1610 REM *
1620 IF SD(7)=0 THEN 1730
1630 IF SD(7)<-4 THEN 1660
1640 TY=0
1650 GOTO 1680
1660 TY=4
1670 SD(7)=SD(7)+4
1680 NC$=CHR$(224+TY+ABS(SD(7))-1)
1690 NA$=CHR$(240+INT(SD(8)/2))
1700 REM *
1710 REM * FINISH AND LOAD
1720 REM *
1730 SOUND$=G1$&G2$&G3$&NC$&NA$&DU$
1740 L$=CHR$(LEN(SOUND$)-1)
1750 SOUND$=L$&SOUND$
1760 LF=LF+LEN(SOUND$)
1770 SAVELC=LC
1780 FOR I=1 TO LEN(SOUND$)
1790 CALL LOAD(LC,ASC(SEG$(SOUND$,I,1)))
1800 LC=LC+1
1810 IF LC>32701 THEN 2850
1820 NEXT I
1830 GOTO 930
1840 REM *
1850 REM * RESET POINTERS
1860 REM *
1870 CALL LOAD(LC,4,159,191,223,255,0)
1880 LF=LF+6
1890 LC=LC+6
1900 IF LC/2=INT(LC/2)THEN 1920
1910 LC=LC+1
1920 FH=INT(LF/256)
1930 FL=LF-FH*256
1940 CALL LOAD(FFM,FH,FL)
1950 LCH=INT(LC/256)
1960 LCL=LC-LCH*256
1970 CALL LOAD(28700,LCH,LCL)
1980 REM *
1990 REM * PLAY LIST?
2000 REM *
2010 CALL CLEAR
2020 PRINT "* ASSEMBLY COMPLETED"
2030 INPUT "* PLAY LIST? (Y/N) ":I$
```

Continued on p. 40

Just Assemble Melody

TI-99/4A

```
2040 IF ASC(I$)=89 THEN 2070
2050 IF ASC(I$)=78 THEN 2210
2060 GOTO 2010
2070 RESTORE 690
2080 LOC=32704
2090 CALL LOAD(LOC-2,H1,L1)
2100 PRINT "* LOADING MACHINE ROUTINE"
2110 FOR I=0 TO 63
2120 READ V
2130 CALL LOAD(LOC+I,V)
2140 NEXT I
2150 PRINT "* PLAYING"
2160 CALL LOAD(28702,127,248)
2170 CALL LINK("PLAY")
2180 PRINT :"* PRESS ANY KEY TO PROCEED"
2190 CALL KEY(0,K,S)
2200 IF S=0 THEN 2190
2210 PRINT :"* COMPLETED SOUND TABLE
         LOCATED AT ADDRESS";FFM
2220 GOSUB 2570
2230 STOP
2240 REM *
2250 REM * SUBROUTINES
2260 REM *
2270 REM * SEARCH FOR NOTE
2280 REM *
2290 FOR J=0 TO 68
2300 IF NO$(J)=S$ THEN 2350
2310 NEXT J
2320 PRINT :"* UNKNOWN NOTE NAME: ";S$::
2330 INPUT " RETYPE NOTE: ":S$
2340 GOTO 2290
2350 SD(I)=FRE(J)
2360 RETURN
2370 REM *
2380 REM * KEYBOARD INPUT
2390 REM *
2400 IF OP THEN 2430
2410 PRINT "* ENTER SOUND LISTS":"* ENTE
     R NULL STRING TO END"::
2420 OP=-1
2430 INPUT I$
2440 IF I$="" THEN 2450 ELSE 2470
2450 FL=99
2460 RETURN
2470 IF I$="REDO" THEN 2480 ELSE 2500
2480 LC=SAVELC
2490 GOTO 2430
2500 IF SEG$(I$,LEN(I$),1)="." THEN 2530
2510 PRINT :"* FINAL PERIOD MISSING.
         TRY AGAIN...":
2520 GOTO 2400
2530 RETURN
2540 REM *
2550 REM * MAKE HEX ADDRESS
2560 REM *
2570 A$=""
2580 FOR D=12 TO 0 STEP -4
2590 N=INT(FFM/(2^D))
2600 FFM=FFM-N*(2^D)
2610 A$=A$&SEG$(H$,N+1,1)
2620 NEXT D
2630 PRINT " (HEX >";A$;")"
2640 RETURN
2650 REM *
2660 REM * CHECK RANGES
2670 REM *
2680 ON I+1 GOTO 2690,2720,2750,2720,275
     0,2720,2750,2780,2750
2690 MIN=1
2700 MAX=4250
2710 GOTO 2800
2720 MIN=110
2730 MAX=44733
2740 GOTO 2800
2750 MIN=0
2760 MAX=30
2770 GOTO 2800
2780 MIN=-8
2790 MAX=-1
2800 IF (SD(I)<=MAX)*(SD(I)>=MIN)THEN 28
     10 ELSE 2820
2810 RETURN
2820 PRINT :"* BAD VALUE: ";SD(I):" ";S
     TR$(MIN);" TO ";STR$(MAX):" IS THE
       VALID RANGE"::
2830 INPUT " RETYPE BAD NUMBER: ":SD(I)
2840 GOTO 2800
2850 REM *
2860 REM * OUT OF MEMORY
2870 REM *
2880 PRINT ::::"**** MEMORY FULL ****"::
     :::
2890 REM *
2890 END
2900 REM *
2910 REM * CHECK MACHINE
2920 REM * DATA - DELETE
2930 REM * WHEN DATA IS
2940 REM * VERIFIED
2950 REM *
2960 FOR I=0 TO 63
2970 READ V
2980 CHECK=CHECK+V
2990 NEXT I
3000 IF CHECK=5790 THEN 3010 ELSE 3020
3010 RETURN
3020 PRINT ::"*** ERROR IN MACHINE ROUTI
     NE"::
```

HCM

Fig. 8 — This listing demonstrates some of the new or modified graphic screen commands available in Cartridge BASIC using the 64KB Memory and Display Expansion card. The 32K of memory reserved by Line 140 is necessary for the high-resolution modes (SCREENs 5 and 6) called for in lines 180 and 260.

```
100 REM NEW SCREEN MODES FOR THE PCjr
110 REM
120 REM RESERVE 32K OF MEMORY FOR THE D
    ISPLAY
130 REM
140 CLEAR ,,,32768!
150 REM
160 REM SELECT MEDIUM RESOLUTION (320x2
    00) WITH 16 COLORS
170 REM
180 SCREEN 5
190 REM
200 REM SELECT A RED FOREGROUND (4), AN
    D WHITE BACKGROUND (15)
210 REM
220 COLOR 4,15
230 REM
240 REM SELECT HIGH RESOLUTION (640x200
    ) WITH 4 COLORS
250 REM
260 SCREEN 6
270 REM
280 REM CHANGE PALETTE COLOR #2 TO RED
    (4)
290 REM
300 PALETTE 2,4
310 END
```

with 4 colors. To see this last high-resolution mode properly (Screen Mode 6) requires a compatible RGBI monitor [see Fig. 3A]. Because most of the mountain of forthcoming third-party educational, entertainment and productivity software will undoubtedly be requiring the versatility of either 16-color medium resolution or 80-column text displays, the 128K-byte Enhanced Model will be the standard in the PCjr world.

The PCjr offers its programmers a rich development environment—especially in 16-color medium-resolution mode (Screen Mode 5). The 320 pixels of horizontal resolution by 200 pixels in the vertical direction allow well-defined graphic images. There is, however, a small price to pay—you must allocate 32K-bytes of your total 128K-bytes of system RAM to the video display. You do this in Cartridge BASIC with the statement:

CLEAR ,,,32768

A Sound Comparison Between the PCjr and TI-99/4A

Although the TI-99/4A and the IBM PCjr use the same Texas Instruments sound chip, there are major differences in the way each system creates sound or music. TI BASIC uses a single command—CALL SOUND—to create both sound effects and music. IBM PCjr Cartridge BASIC has three different commands—SOUND, NOISE, and PLAY—that control the sound chip.

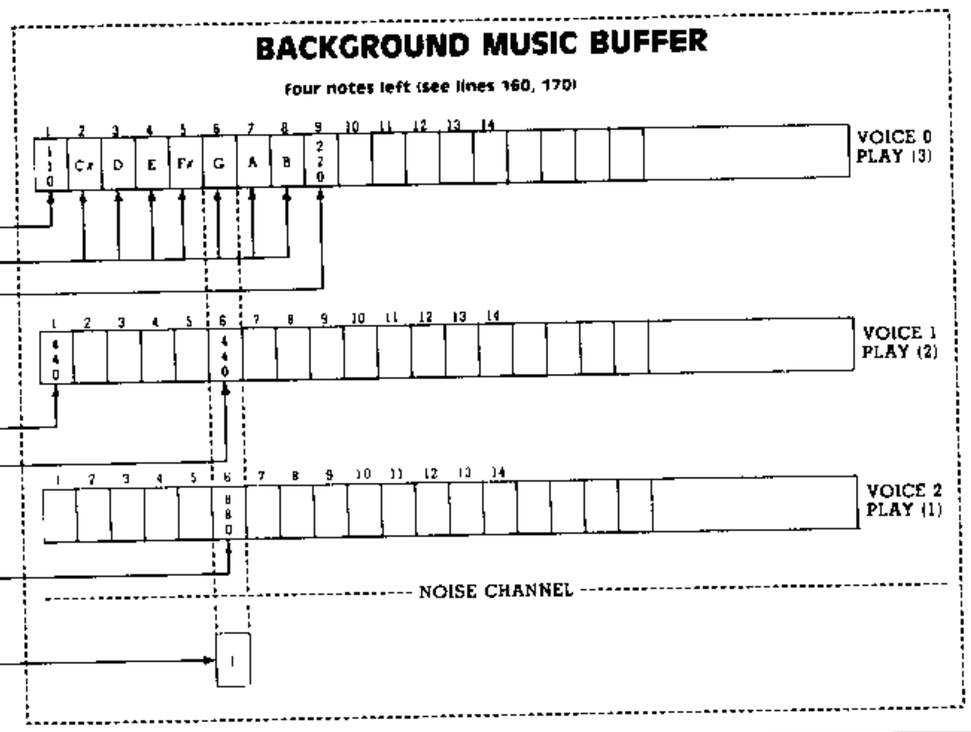
The PCjr's SOUND command is similar to the TI command CALL SOUND. Both let you specify the frequency, duration, and volume of a tone. The difference is that with the TI command you can specify all three voices plus the noise channel within a single command. All three tones and the noise that are created will have the *same* duration. On the PCjr, *each* of the three voices is activated with a separate SOUND command, and the noise channel is activated with NOISE. This may seem awkward at first, but it has its advantages: Because the individual voices are set up with their own SOUND commands, each voice can have a *different* duration.

PCjr Cartridge BASIC has a third command—PLAY—which is unlike anything found in TI BASIC. The PLAY command lets you set up a string of sub-commands to compose music with the actual notes (instead of by frequency). A program can play three of these command strings at the same time, with each string controlling one of the three voices. The noise channel can't be directly inserted into the PLAY command, but if the music is playing in the background mode (e.g., the program continues while the music buffer plays), then the NOISE command can be used to mix noise with the three voices. The placement of the noise within the music chain of commands can be calculated by testing one of the three voices to determine the number of notes left in the buffer. When the program is creating sounds or playing music in the foreground mode, the statement that is creating the sound must finish executing before the program proceeds to the next statement.

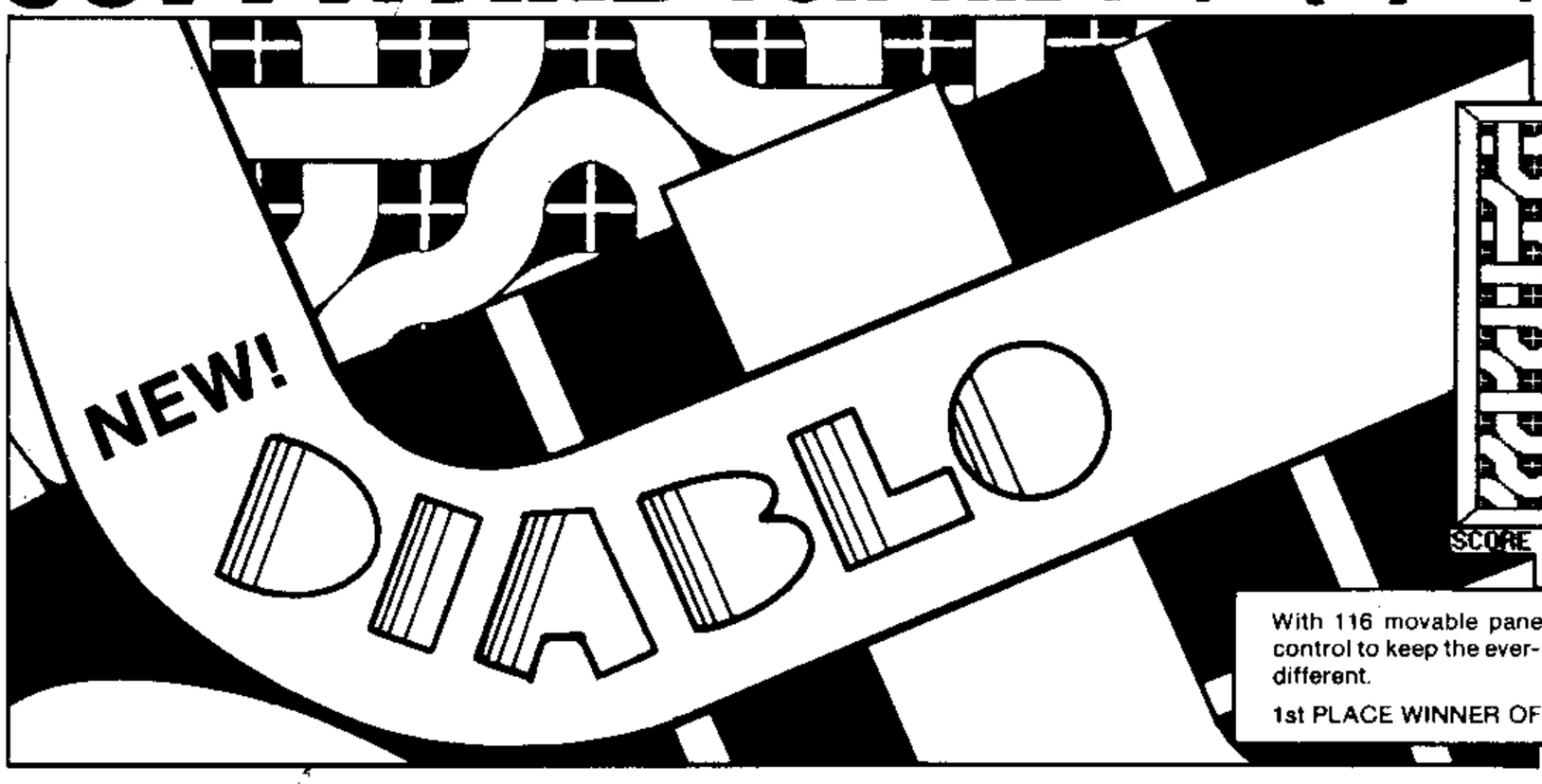
Explanation of the Sample Program

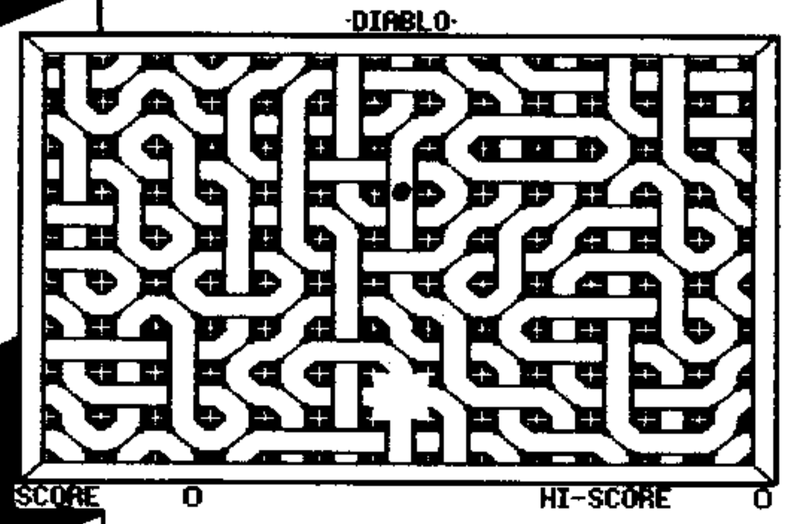
Line No.	Purpose
100	Turns on the sound chip and turns off the internal piezoelectric speaker.
110	Places the sound in background mode. All sounds created with the SOUND command and PLAY command will be placed in the music buffer.
120	Places a tone in the music buffer with a frequency of 110Hz, a duration of 20 clock ticks (each clock tick is 18.5 ms.), and a volume of 15. The output is to voice #0.
130	Places the string of music commands into the default music buffer (voice #0). These notes follow the tone created in line 120 because they are on the same voice channel.
140	Places a tone of 220Hz in the music buffer for voice #0 following the PLAY command string from line 130.
150	Places a tone of 440Hz into the first position of the music buffer for voice #1. This tone will start at the same time that the tone from line 120 starts because they are in the same position in the music buffer.
160	Tests the buffer for the number of remaining unplayed notes.
170	When the number of notes left reaches four, the program branches to line 190. Subroutine returns; program halts.
180	Continues checking the buffer.
190	Places a tone of 440Hz in the music buffer for voice #1. This note will start to play when there are four notes left in the buffer for voice #0.
200	Places a tone of 880Hz into the buffer for voice #2. This tone will play at the same time as the tone from line 190 and the note G from voice #0.
210	Turns on the noise generator at the same time as the tones in lines 190, 200, and the note G in voice #0.
220	Exits the subroutine.

```
100 BEEP OFF: SOUND ON
110 PLAY "MB"
120 SOUND 110, 20, 15, 0
130 PLAY "C#DEF#GAB"
140 SOUND 220, 20, 15, 0
150 SOUND 440, 20, 15, 1
160 A = PLAY (3)
170 IF A = 4 THEN GOSUB 190: END
180 GOTO 160
190 SOUND 440, 20, 15, 1
200 SOUND 880, 20, 15, 2
210 NOISE 1, 15, 20
220 RETURN
```



SOFTWARE FOR THE 99/4(A) FROM EXTENDED SOFTWARE COMPANY

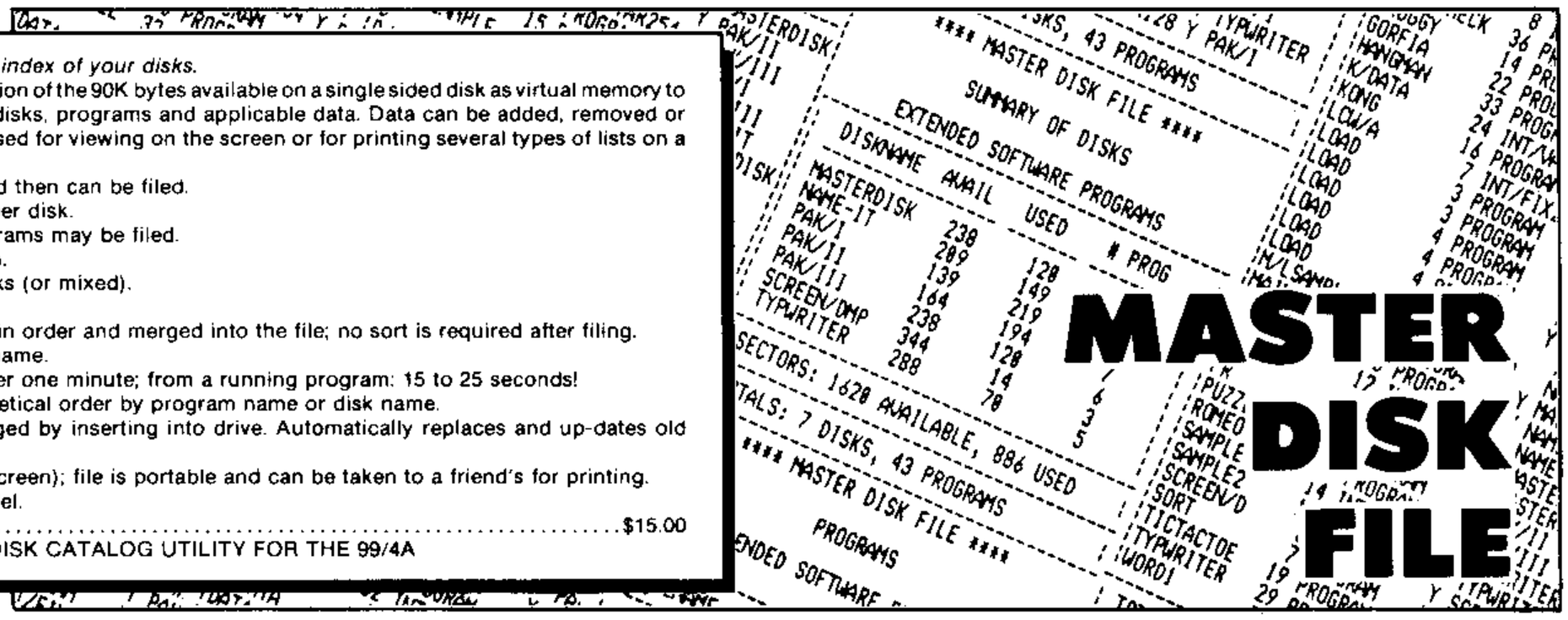




With 116 movable panels, this game demands strategy, planning & decisive control to keep the ever-advancing ball from rolling off the board. Every board is different.

1st PLACE WINNER OF THE T.I.S.H.U.G. AUSTRALIA CONTEST

MASTER DISK FILE — *A master index of your disks.*
MASTER DISK FILE uses a large portion of the 90K bytes available on a single sided disk as virtual memory to create and store a perpetual file of disks, programs and applicable data. Data can be added, removed or updated. The file may then be accessed for viewing on the screen or for printing several types of lists on a printer.
Insert disk into drive, it is catalogued then can be filed.
Reads up to 95 programs and files per disk.
Maximum of 120 disks or 1100 programs may be filed.
Does not require memory expansion.
Supports single or double sided disks (or mixed).
Supports single or multiple drives.
Programs are catalogued from disk in order and merged into the file; no sort is required after filing.
SEARCH for disks or programs by name.
Look-up time from a cold start: under one minute; from a running program: 15 to 25 seconds!
List on screen or a printer in alphabetical order by program name or disk name.
Up-date disks that have been changed by inserting into drive. Automatically replaces and up-dates old information.
May be used without a printer (on screen); file is portable and can be taken to a friend's for printing.
Supports any printer: serial or parallel.
Diskette (only) $15.00
GUARANTEED TO BE THE BEST DISK CATALOG UTILITY FOR THE 99/4A



GAMES PAK/III

KONG - must fight his way past barrels, ladders & trap doors to save Roxanne from Igor's bomb. Six different screens. Joystick req. $15.00

BOUNCER — bounds from one trampoline to another, scoring points while avoiding the arrows which will burst him. 6 screens. Joysticks req. $15.00

ROMEO — must traverse a desert, stream (alligators & sharks) & cavern to obtain a fitting reward. Joysticks req. $15.00

GAMES PAK/II

ARTILLERY — The opposing force must be destroyed by determining angle and force of each shot. An ever-changing wind complicates matters. 1 or 2 players $9.95

DE-CYPHER — An encrypted message is displayed and guesses change all corresponding letters to the guess. Comes with 50 messages which can be changed $9.95

PUZZLE 15 — Slide alphabetic squares into order. Multiple squares may be moved. Computer keeps track of moves & has replay feature $9.95

FLIP CHECKERS — Sandwich opponent's piece to flip it to your color. 1 or 2 players. Real-time clock. Fast computer decisons. Joysticks req. $9.95

GAMES PAK/I

FROGGY — Jump FROGGY across 10 lanes of traffic, then across 6 logs. Keyboard or joysticks $9.95

EXTENDED BASEBALL - Joystick control of the pitcher & batter, and individual batting averages. Multi-base & multi-runner plays. Joysticks req. $9.95

GORFIA PESTULITIS - Joystick control of a laser sight or inertia influenced space mines to shoot down the invading Gorfians. Joysticks req. $9.95

EXTENDED HANGMAN - Quick graphics, music, color, speech (optional) and sound are added to keep the players entertained. Includes 580 words of 4 to 9 letters in length in easy, medium and difficult groups $9.95

TIC-TAC-TOE — Quick set-up and quick decision making at four levels of difficulty. The levels avoid the frustration of the novice never having a chance to win $9.95

TYPWRITER
A 16K WORD PROCESSOR on Cassette or Disk
Any Input/Output storage of text — disk, cassette, cassette input/disk output, or vice versa.
Selectable *right justify.*
Complete text *Editing* — by cursor control; including insert & delete lines, partial print, printer halt or abort without text loss, page FWD & BKWD, and more.
Complete *Software Control* of Printer (depending upon its capabilities) — for enhanced print, underlining, formatting, 28 to 254 characters per print line, etc.
No Special Equipment — monitor, console, Extended Basic module, C or D, printer.
Comes with a 20 page instruction booklet.
Cassette $32.00 Diskette $35.00

SCREEN/DUMP — Print the screen to a dot-addressable, 8-bit printer. Does not require extra memory! Disk version is simple to use. Cassette version requires mild programming knowledge $12.00

NAME-IT
DATA BASE for: *Mail Lists, Labels, Files*
Records: 250 records per diskette consisting of up to nine 28-character items per record.
Prompts: user designated prompts.
Complete *File Sort*: 250 records in *100 Seconds.*
Search; Pre-set; print labels & lists.
Includes a FORM LETTER program that uses NAME-IT data in TYPWRITER generated form letters
Cassette version differs from disk version.
Cassette $32.00 Diskette $35.00

NOTE: Should you decide to up-grade to the TI-WRITER module, TYPWRITER and NAME-IT data can be converted for use by that module. NAME-IT alone, will generate 250 TI-WRITER form letter records.
TI-WRITER is copyrighted software of Texas Instr.

A detailed catalog is available free. Circle "FREE" on the order form or send a letter or postcard.

ORDER FORM

EXTENDED BASIC MODULE REQUIRED FOR ALL PROGRAMS

CATALOG	FREE		$ FREE
GAMES PAK/I (Froggy, Extended Baseball, Gorfia Pestulitis, Extended Hangman, Tic-Tac-Toe)	$26.95	(C or D)	$
GAMES PAK/II (Artillery, De-Cypher, Puzzle 15, Flip Checkers)	$26.95	(C or D)	$
GAMES PAK/III (Kong, Bouncer, Romeo)	$26.95	(C or D)	$
TYPWRITER (word processor)	$32.00	(C price)	$
	$35.00	(D price)	$
NAME-IT (data base/mail list)	$32.00	(C price)	$
	$35.00	(D price)	$
SCREEN/DUMP (printer required)	$12.00	(C or D)	$
MASTER DISK FILE	$15.00	(D only)	$
INDIVIDUAL GAMES:			
DIABLO	$19.95	(C or D)	$

☐ Froggy ☐ Extended Baseball ☐ Gorfia Pestulitis ☐ Extended Hangman ☐ Bouncer ☐ Kong
☐ Tic-Tac-Toe ☐ Artillery ☐ De-Cypher ☐ Puzzle 15 ☐ Romeo ☐ Flip Checkers

Total Individual Games: (C or D) $ _____
Shipping & Handling (1st Class Mail): Included
Sales Tax: Included
Add $2.00 if C.O.D. (U.S. Mail Only): $ _____
Check or money order or C.O.D. Total: $ _____
PLEASE SPECIFY CASSETTE OR DISK

Send this form or a substitute with check or money order to:
Extended Software Company
11987 Cedarcreek Drive
Cincinnati, Ohio 45240
(513) 825-6645

SATISFACTION GUARANTEED
IF YOU ARE NOT COMPLETELY SATISFIELD, YOU MAY RETURN THE PROGRAMS (and instructions) WITHIN 15 DAYS FOR A FULL REFUND OF YOUR PURCHASE PRICE.

Dealer and programmer inquiries welcome.

Cut Here (may be copied or substituted)

HEN PECKED

A Review by Greg Roberts
99'er HCM Staff

Name:	Henpecked
Program Type:	Barnyard Affair
Machine:	TI-99/4A
Distributor:	Navarone Industries 501 Vandell Way Campbell, CA. 95008
Price:	$37.50, cartridge

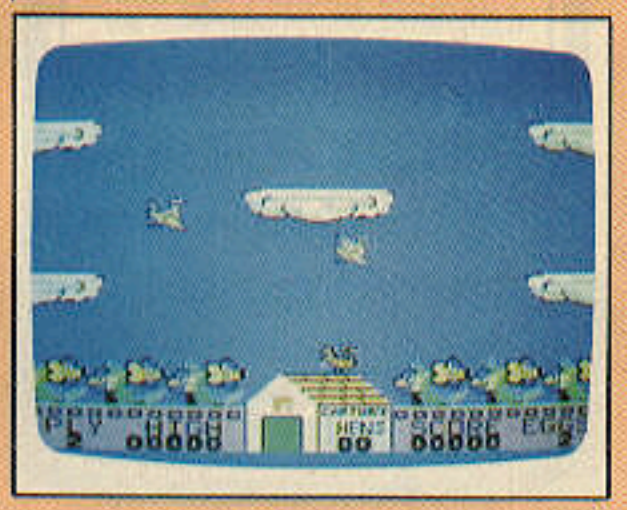
System requirements:
Assembly Language, Joysticks optional

	Poor	Fair	Good	Excellent
Performance:			■	
Engrossment:			■	
Documentation:		■		

A computer chicken's life is not easy. Crossing the road, of course, is a day-to-day struggle for some chickens, while others are stuck in computer games with wolves and other predators. In *Henpecked* the problems are domestic: The birds are caught up in a courtship ritual that is not all love and kisses; in fact, it's closer to the domestic turmoil of a nest of black widows.

The prime mover in this drama is a red rooster whose only aim in life is to harass the hens that flutter and fly about the chicken house in the center foreground of your screen. To play this game, use a joystick or arrow keys to keep the rooster hovering, and to help him pull off a successful "landing," at which point the hen lays an egg and disappears. You may then chase the rolling egg and descend on it to rack up even more points—a move you can repeat time and again over the same egg. Even the flying hens can lay, and their eggs always fall gently to the ground where they hatch into more hens.

But these hens are definitely not the passive kind. If a she-bird lands on the

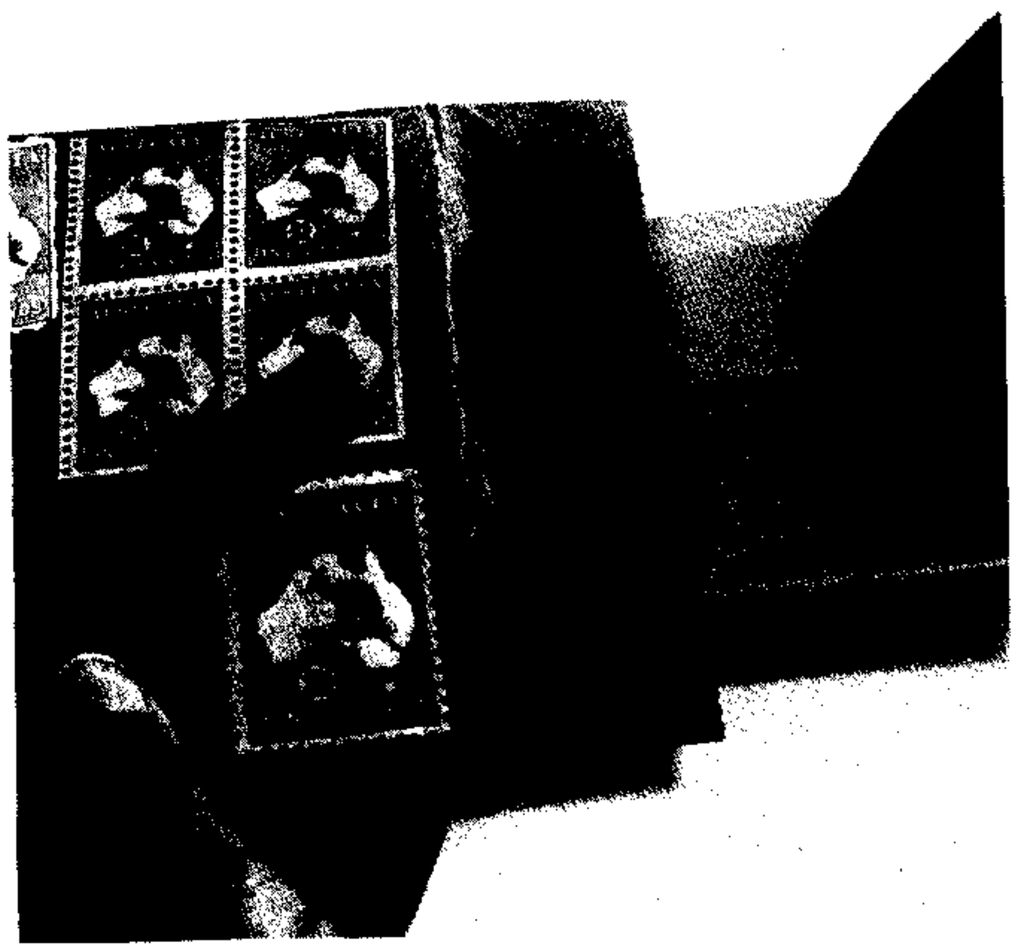


rooster, he's dead. And there is just one rooster against a whole flock of hens, a flock which gets bigger and more aggressive at the higher levels of play. There are ways, however, to protect your rooster: station him just under a cloud or at the top of the screen—either of which will act as a shield from the Amazon chickens attacking from above. Thus stationed, you can make short, not very daring attacks on the hens, and build up a score that would even make Casanova grunt his approval. This way, however, the game gets too easy, unless you establish a rule against cloud-hovering and hire a referee to enforce it—someone who understands rabbit punches, steroids, and sticky bats.

The game's scenario may not be an absolute delight to everyone—in fact, it is easy to see why half the population might be rubbed the wrong way by these birds' mating ritual. On the other side of the coop, game defenders might argue that the hens do have the opportunity to clobber the rooster—but still there is only one way to score.That's with the rooster in control. This game is less concerned with equality than it is with showing traditional barnyard scenes.

Uptown Barnyard Graphics

As for barnyard realism, this game's birds are expertly drawn and animated. They strut, flutter and fly with an intricacy of detail unsurpassed and seldom equalled in computer games. Yes, for all you nit-pickers plucking at details, chickens can fly to some extent; the



HOPPER

A Review by
Greg Roberts
HCM Staff

Name:	**Hopper**
Program Type:	**Arcade game**
Authors:	**Michael D. Archuleta and John M. Phillips.**
Machine:	**TI-99/4A**
Distributor:	**Texas Instruments, Inc. P. O. Box 53 Lubbock, TX 79408**
Price:	**$39.95, cartridge**

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

Long ago, when the TI Home Computer was very young, a small band of users wandered over the outback looking for games. Tied to their bleak terrain, these aboriginal arcadians would pick up anything that looked loadable, digging with their joysticks the most primitive software.

Over the past few seasons, however, the desert has bloomed. Games are plentiful. Tonight the tribe sits well-satisfied round a billabong overflowing with fat, colorful tapes and diskettes, and each game hunter's CPU is warm and full.

In times of plenty the games people like to tell stories: "The rarest beast of all is the plug-in cartridge from the dry plains of the south," an elder tribesman explains to the young men who squat around a gray-backed monitor. "Such games come so seldom, each one brings the whole tribe out to see it—and once in a pink moon there comes a trophy that makes the whole clan celebrate."

But a cartridge of such a calibre has not shown its prongs for many seasons, and the games people have been craving to try their remote controllerangs on something new. Memories have grown dim of the last notable game from that remote place, and the tribe's chief—a user so positively ancient that he still remembers rubbing his fingers against the Chiclet keys of a 99/4—tries to recall the legendary coming of the last great cartridge. "Its name was Parsec," he explains, "and I seem to remember a great rejoicing in the desert, but it has been so long—and my mind is scrambled from staring endlessly at the computer-generated rocks and gullies." Suddenly the withered one cuts short his story, and all eyes stare out into the

birds of this flock usually move quite realistically as they fight and flap through the nuptial dance. As for the chickens bombing through the clouds, eggs falling from the sky, and sudden death by hen-smothering, we have no choice but to switch gears and bear in mind that this is a computer game, not a natural history text. In any case, the program is a good example of the stunning graphics that can be achieved with a self-contained ROM cartridge which requires no additional memory packages, disk drives, or other expensive peripherals.

It is unfortunate that these superb graphics are without a game plan of equal worth. Regardless of expertly-drawn characters, a half-hour at the keyboard will reveal that there isn't enough going on here. Like professional actors in a weak play, the birds are all dressed up with no place to go. And as they float around on the screen, they sometimes fail to respond to the joystick, even when it's right on target.

Unwritten Rules

Many buyers of this cartridge may have to take some time to figure out the object of the game. The cartridge comes without documentation of any kind except for vague remarks on the carton such as "You as the rooster have a choice: to rule the roost or become totally henpecked." These loosely-stated rules of the game point to a program with loose ends here and there—idiosyncrasies that may annoy some players. Others will appreciate the game just for its graphics. **HCM**

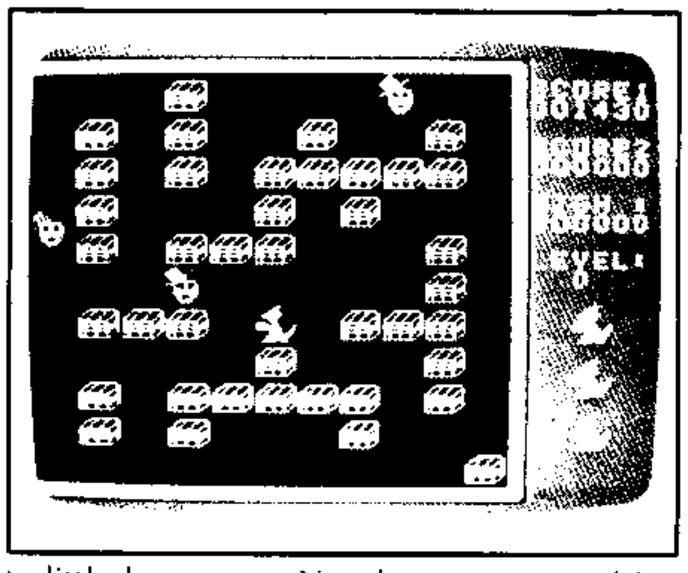
blackness to see a wonderful beast lope into the light of the campfire . . .

We who are stationed at that remote outpost called the *HCM* review desk also have full tucker bags of fine gameware these days. And as we load game after game, it is easy to feel a bit jaded, even fire-opaled, sometimes, from so much rich fare. When a game comes up on the screen, our first reaction is to pigeonhole it as to place of origin—whether it be The Land of Kong, Froggerania, or Pac-Man-Du. But this time we are unable to peg the program. Yes, believe it or not, there's something new under the joystick, a new TI Command Cartridge called *Hopper*.

Speeding through a maze of packing crates, three evil circus trainers are trying to grab Chadly, the pink kangaroo. You can make Chadly kick the crates to form barricades, or let him drop the cargo on his enemies. He can smash crates too, in order to make a trap for the trainers—permanently fencing them in. With this movable maze, the strategies are unlimited, giving this game near immunity to boredom.

This is a fast-paced game. You must think and move quickly to protect your little pal joey as you devise ways to trap the trainers. By some bizarre quirk of fortune, I managed to capture all three trainers together in one box where I could put the squeeze on them—Pit and the Pendulum style—and come up with a thrilling score of 5,870 on my first of the three kangaroos. As I bragged that this was the highest score possible for a first inning, someone more perceptive than I observed that, in fact, it would be easy to rack up a higher score based on an oddity of the game: You get ten points for kicking a crate, no matter what. So, once you've captured at least one trainer and squashed the others, you can sit there kicking crates until they are all lodged in the corners of the square. Then you can smash the crates for an additional sixty points each. These easy points can be taken as an extra reward for capturing the trainers, or as an annoyance. Fortunately, they are not significant enough to greatly alter a high score.

Hopper's graphics are good, but not extraordinary—except perhaps for the feisty little kangaroo. You have to move him right, left, forward, or back in order to take action on the crate of your choice. These precise joystick maneuvers add to the challenge.



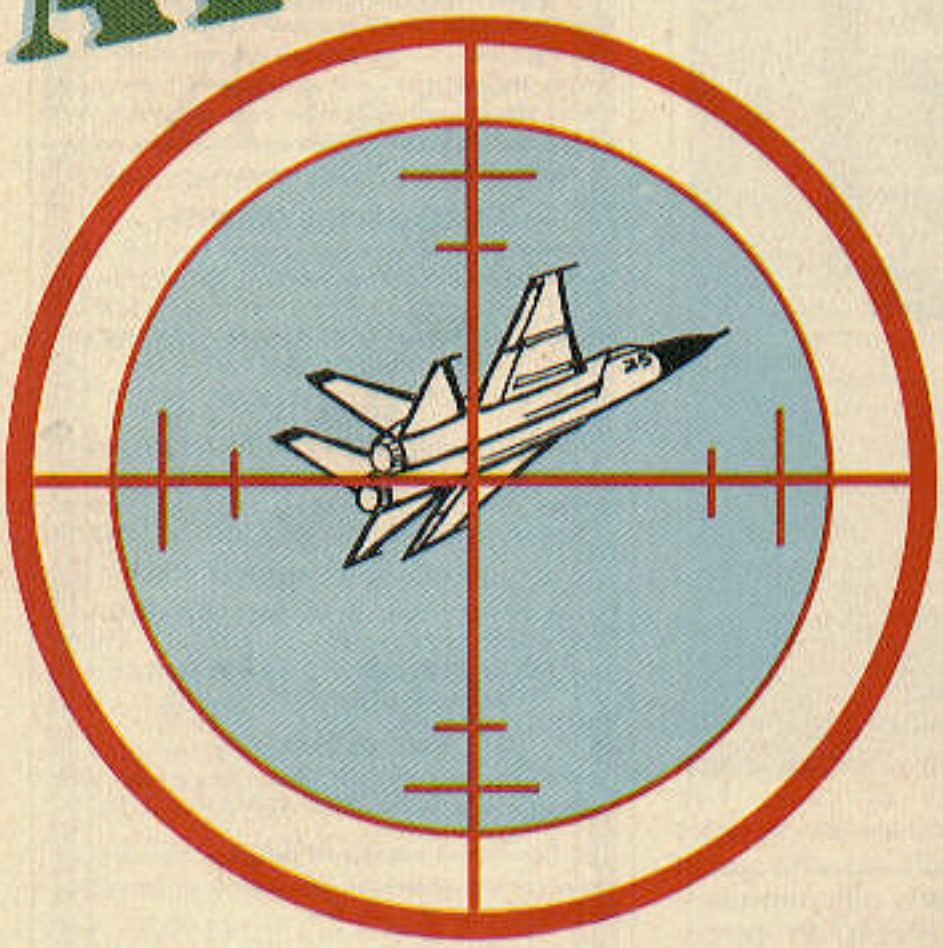
The game's sound effects are logical and simple. The high-pitched beeping might get annoying after a while, but the volume control is always handy.

The documentation (which at press time was in manuscript state, but adequate) mentions nine levels of play. The progressively-higher levels are simply faster and more difficult, with no change in scenery except to offer a different crate configuration. Only at Level 10 do you encounter a new and very difficult challenge. This is a "surprise" screen, so all we can tell you is to keep your eyes open for some flash of recognition, or it will be "lights out" for that round and back to Level 1.

Designers Phillips and Archuleta came up with *Hopper* as part of a TI employee-incentive program in which they designed the product at home, on their own time, with off-the-shelf equipment. No special TI-proprietary development systems were used on this one. That's right, lads, this particular big game was not bagged with the high-powered weapons of the Lubbock laboratories, but with the slings and arrows of that outrageous little device called the Editor/Assembler package. The program takes up only about 6K-bytes of memory.

This game is just the sort of program cartridge that makes the 99/4A such a good buy—even if you have nothing but a console and monitor. The best choice, mate—definitely the dinkum oil. **HCM**

FLAK ATTACK



by Mark Moseley

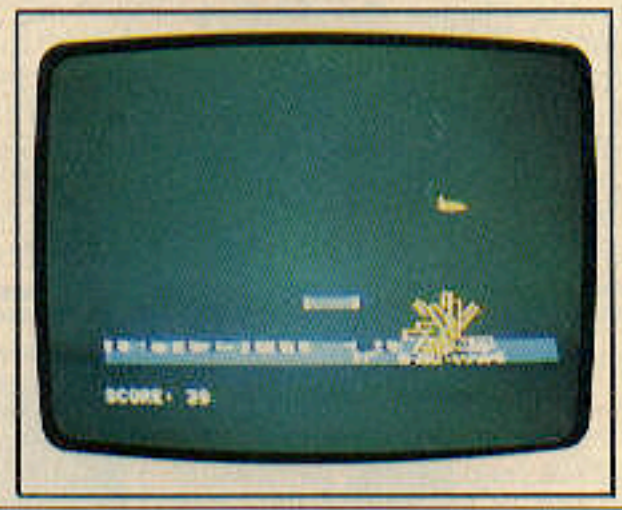
and the HCM staff

The best defense is a good offense. Or is it the other way around? In *Flak Attack,* either maxim could apply. Your job is to shoot down an attacking plane with your missile launcher before it blasts you three times. The plane attacks from left and right at random altitudes. The enemy's speed and frequency of fire vary with the skill level you choose. Your missile launcher can move back and forth and hide behind a barrier. Your launcher should be kept moving because the plane "remembers" its last position and is likely to fire at that spot.

But don't expect the barrier to last forever—each time the pilot hits the barrier, he makes a hole that exposes you. Likewise, your missile launcher can fire through the holes blasted by the plane. Three keys—the S, D, and E keys—control your moves. The S key moves the missile launcher to the left, and the D key moves it right. The E key fires the missile. The fate of your government depends on your ballistic skills. Good luck, Captain.

***Note: TI readers who have been with us from the beginning will recognize* Flak Attack *as a new, improved version of* Anti-Aircraft Gun. *It originally appeared in Volume 1, Number 1 (May/June 1981), of* 99'er Magazine.

The moving plane is created with the DRAW command on the Apple IIe. DRAWing the plane at its first location in the same color as the background erases the plane; then reDRAWing it at a new location in the plane's color gives the plane the illusion of motion. The DRAW command also creates the exploding tank at the bottom of the screen. Rotating and expanding the shape with the ROT = and SCALE = commands creates the illusion of an explosion.



The IBM PCjr uses the DRAW command to create the plane and the exploding tank. The explosion effect is created by passing a series of new values to the DRAW string. These new values change the rotation angle and the scale size of the object.

Rather than drawing high-resolution lines, the Commodore C-64 and VIC-20 and the TI-99/4A create the effect of an explosion by making a series of alterations in the character definitions displayed at that location. The Commodore machines use POKEs; the TI uses CALL HCHAR.

The Apple version of *Flak Attack* uses shape tables and the DRAW command to produce animated graphics. The DRAW command lets you create shapes and then expand or rotate them. To use this command, you must first generate the code for the shape. These codes are listed in the Applesoft BASIC Programmers Reference Manual.

You will also need to create a shape table index so that the computer knows where you have stored the shape data in memory. The shape table index must include the number of shapes being defined. In addition, each entry needs an offset from the beginning of the index to the beginning of each shape table. The index created in line 1490 contains six shapes, with the first shape starting 15 bytes after the beginning of the index. The second shape starts 31 bytes after the start of the index, directly after the first shape. When the program says DRAW 1 AT 10,10, it will draw shape 1 (the cannon) at screen location 10,10.

in effect, using commands within a command. Graphics representations are easy to implement using commands such as L for left, U7 for up seven, and so on. You can also change the scale of the object you are drawing right in the middle of a string with S*x* (*x* being the scale factor from 0 to 255). A scale of 4 is normal size, so 1 would be one-quarter size, and 8 would be double-size.

The shape can also be rotated to any angle permitted by the screen's resolution. The angle of rotation is set with TA*x* (for Turn Angle), where *x* is any number from −360 to 360. This function is demonstrated by the demolition of the cannon in line 1170. This line is a FOR-NEXT loop running from −360 to 360, expanding the shape as it rotates. The shape isn't erased before redrawing, so the effect is an expanding explosion of color.

If you don't erase your animated graphics before redrawing, they will leave a trail, and parts of the new shape that lie over the old shapes might be erased. You erase simply by changing the color with the C*x* command in the DRAW command string, where *x* is any number representing a legal color for the resolution being used. The cannonball in line 900 uses this process. First the coordinate is chosen using the PSET statement, and the old shape is erased. Then the coordinate is moved four pixels up the screen and the ball is redrawn. A designation of C0 after the DRAW command sets the color to black to match the background. The C2 specifies the color red, which is one of the four colors (green, red, black and brown) currently available in this mode.

One of the toughest bugs in this program showed up in the keyboard buffer. If a key is held down too long, the cannon will take off and not stop until the buffer of sixteen characters has been used. This problem is taken care of in line 710, which erases the keyboard buffer. If used every time the keyboard is read with INKEY$; as in line 700, the buffer will be cleared. This method lets you hold down a key for as long as you like, so that you can stop the cannon and fire the instant you release the key.

Continued on p. 72

The TI version of *Flak Attack* takes advantage of TI BASIC's graphics and color. Because there are no sprites in TI's console BASIC, it is not possible to get the smooth action of Extended BASIC. However, this does not hamper the game significantly. The scenario is the same as in the other versions, with a fighter plane firing at your surface guns. You have a barrier to hide behind (until it gets destroyed). If the barrier gets hit, part of it disappears, and the next shot at the same spot will go all the way through. But this hole in the barrier will also allow you to fire through it. You have only three surface guns, and once all three are destroyed the game ends. The S and D keys are used to move the cannon back and forth, and the E key is used to fire.

Flak Attack (TI-99/4A)
Explanation of the Program

Line nos	
100-160	Program header.
170-230	Title screen.
240-290	Initialize variables.
300-350	Input level of difficulty.
360	Stop the program if option 4 is selected.
370-430	Set up graphics characters and initialize game.
440-510	Set up graphics colors and display the playing screen.
520-610	Move plane and missile.
620-660	Read keyboard and branch to subroutines.
670-720	Move gun left.
730-780	Move gun right.
790-860	Initial firing of the cannon.
870-980	Initialize altitude and direction of plane.

Continued on p. 80

Flak Attack . . . from p. 61

Flak Attack (Apple) Explanation of the Program

Line nos.	
100-170	Program header.
180-290	Initialize game; display options page.
300-370	Display playing screen.
380-450	Initialize plane and place on screen.
460-570	Player's move.
580-650	Move plane and check laser fire.
660-720	Initialize cannonball and place on screen.
730-810	Cannonball routine; check for hit.
820-930	Plane explosion routine.
940-1040	Fire laser.
1050-1140	Hit tank.
1150-1230	End of game options.
1240-1270	Read and display data.
1280-1320	Key scan routine. Reads keyboard but doesn't wait for key press. Scans once, returns to game.
1330-1410	Data for title screen.
1420-1740	Data for shape tables.
1490	Shape table index.
1530	Cannon.
1570	Cannonball.
1610-1620	Plane/eastbound.
1660	Plane/westbound.
1700	Plane explosion.
1740	Ground explosion.
1750-1800	Data for skill-level screen.
1810	End of program.

APPLE II Series

```
100 REM ******************
110 REM *   FLAK ATTACK    *
120 REM ******************
130 REM BY M. MOSELEY AND
140 REM THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.1.1
170 REM   APPLE II SERIES APPLESOFT
```

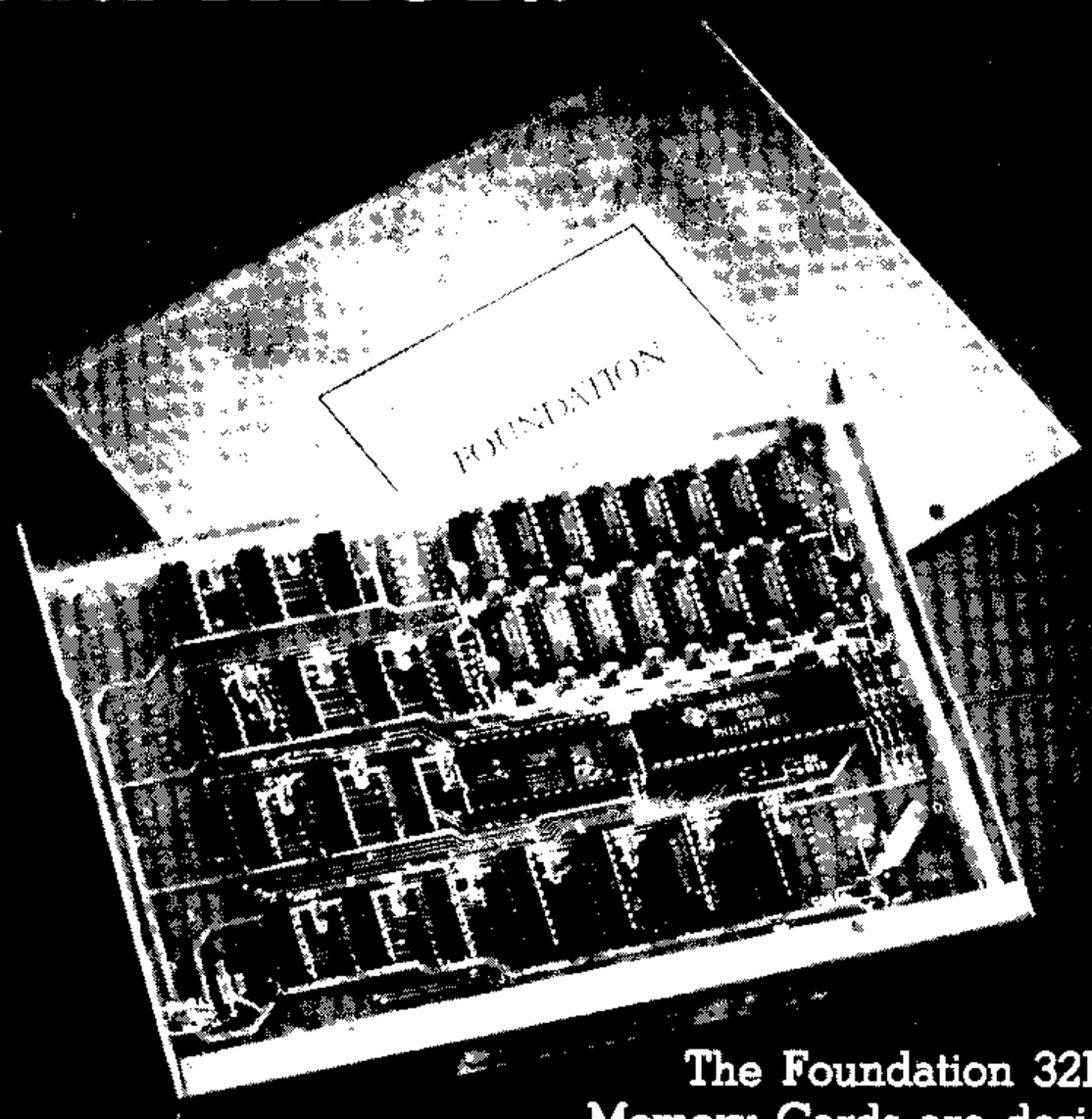
APPLE II Series

```
180 REM **********
190 REM INITIALIZE
200 REM **********
210 TEXT
220 YP = 141: ROT= 0:PS = 3:PT = 0: SCALE= 1
230 HO(0) = 1:HO(10) = 1
240 HOME :NL = 6: GOSUB 1270: GET A$
250 FOR X = 24576 TO 24740: READ A: POKE X,A: NEXT X
260 POKE 232,0: POKE 233,96
270 HOME :NL = 5: GOSUB 1270
280 GET A$: IF A$ < "1" OR A$ > "3" OR LEN (A$) > 1 THEN GOTO 280
290 D = VAL (A$)
300 REM ******************
310 REM DISPLAY OPTION PAGE
320 REM ******************
330 HGR : HOME : HCOLOR= 6
340 VTAB 23: HTAB 1: PRINT "SCORE: ";
350 FOR X = 149 TO 159: HPLOT 0,X TO 279,X: NEXT X
360 FOR X = 125 TO 130: HPLOT 124,X TO 156,X: NEXT X
370 XDRAW 1 AT YP,147
380 REM ********************
390 REM INITIALIZE THE PLANE
400 REM ********************
410 ALT = INT ( RND (1) * 100 + 10)
420 DIR = INT ( RND (1) * 3 - 1): IF DIR = 0 THEN GOTO 420
430 IF DIR = - 1 THEN P = 3:PY = 16
440 IF DIR = 1 THEN P = 4:PY = 260
450 XDRAW P AT PY,ALT
460 REM ***************************
470 REM PLAYERS MOVE / SCAN KEYBOARD
480 REM ***************************
490 IF C = 1 THEN GOSUB 760
500 IF SD = 1 THEN SD = 0: GOTO 410
510 GOSUB 1310: VTAB 23: HTAB 8: PRINT PT;"": IF ST = 0 THEN GOTO 610
520 IF KEY = 83 THEN HCP = - 4: GOTO 560
530 IF KEY = 68 THEN HCP = 4: GOTO 560
540 IF KEY = 69 AND C = 0 THEN GOSUB 690
550 GOTO 610
560 IF YP < 5 AND HCP = - 4 OR YP > 274 AND HCP = 4 THEN GOTO 610
```

Continued on p. 69

NEED MORE MEMORY?

ORDER THE 128K MEMORY CARD BY FOUNDATION.



If you're writing programs and can use more memory. Or if you want to take advantage of large memory programs. The Foundation 128K Memory Card is just what you need.

Foundation has been shipping memory cards all over the country. And our customers have been calling us with nothing but praise. We're delighted, but not surprised.

Both our 32K and 128K Memory Cards begin and end with quality. From the design by our MIT-trained engineers. To using the finest components and prime integrated circuits from top manufacturers. Encased in a durable metal case that plugs into your Peripheral Expansion Box.

The Foundation 32K and 128K Memory Cards are designed from the beginning to be compatible with the TI99/4A Computer. The 128K Card comes with solid-state software that makes it easy to use the additional memory. Just write or call us and we'll send you a descriptive brochure.

The choice is yours. Order now and take advantage of Foundation's Special Introductory prices.*

The Foundation 32K Memory Card - $150.
The Foundation 128K Memory Card - $230.
Both with a full three month warranty.

FOUNDATION
74 Claire Way
Tiburon, CA 94920
415/388-3840

*Add 3%-shipping and handling charges. California residents add 6% sales tax. Allow 2-4 weeks for delivery.

ZEUS

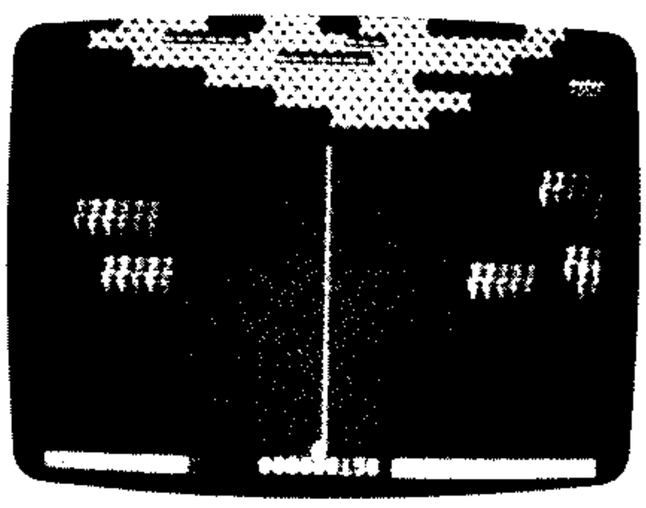
A Review by Greg Roberts

HCM Staff

Name:	**Zeus**
Program Type:	**Arcade game**
Machine:	**Commodore 64**
Distributor:	**Aardvark Systems, Ltd. 2352 S. Commerce Rd. Walled Lake, MI 48088**
Price:	**$29.95, diskette $24.95, cassette**

System Requirements:
Joysticks

	Poor	Fair	Good	Excellent
Performance				
Engrossment				
Documentation				

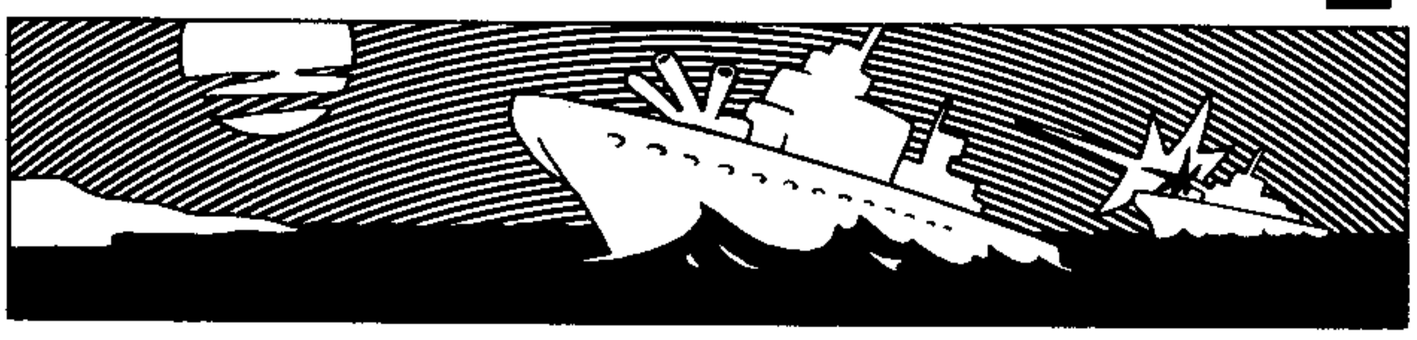


As advertised, this game is truly "fast and furious." The human eye and hand have a hard time following and blocking thunderbolts hurled down from the sky by an angry Zeus. You, the mighty wizard, can destroy them with cold rays, but if you hold down the fire button for more than a second or two, your power starts to fizzle. The strategy then, is to blast away when the bolts are dense and right overhead, and to hold fire when they are higher and more scattered. If you get in a real jam—with a thunderbolt at face level and a sick ray gun—you can pull back on the joystick and enclose yourself in a protective shield. To make this pay off, your timing must be extraordinary. We're talking about tenths of a second.

The first few levels of play are quite easy to achieve—a show of programming psychology I always welcome. There is no difference, other than speed, between one level and the next. The scene stays the same: a small Zeus figure in the clouds and a wizard on the ground. The clouds frequently flash different colors, but that's about the extent of the fireworks.

Zeus is a very simple action game that is best suited for the beginner. It does not require much strategy; rather it is a good diversion when you just want to blast at an enemy and to see how long you can survive.

HCM



BEACH-HEAD

A Review by Greg Roberts

HCM Staff

If Dad or Grandpa was part of the action in the Pacific in WWII, don't put on *Beach-Head* when these old boys are trying to sleep. It's bound to give them nightmares. For a computer game, it's as realistic as you can get.

Your first challenge in this program is to maneuver as many ships as you can (up to 10) through a hidden passage to the battle site. The channel here is full of mines and torpedoes, and the number of ships you bring through the passage helps determine your military strength

Continued on p. 66

Name:	**Beach-Head**
Program Type:	**Conventional Warfare**
Author:	**Bruce Carver**
Machine:	**Commodore 64**
Distributor:	**Access Software 925 East 900 South Salt Lake City, UT 84105**
Price:	**$34.95, diskette**

System Requirements:
Joystick

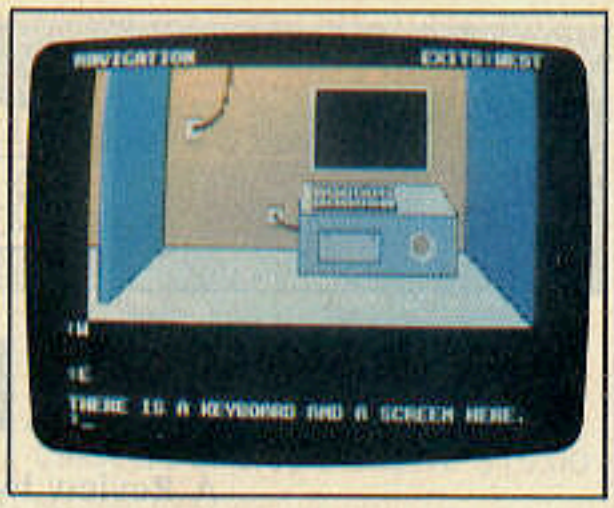
	Poor	Fair	Good	Excellent
Performance				
Engrossment				
Documentation				

cartoon characters are designed with a humorous touch.

The game certainly has an expansive feel: Not only does it take you through buildings with many rooms, passageways and stairwells, but it forces you to teleport yourself to other planets. You are called upon to deliver fuel to a stranded spy pilot on Pluto, and you end up planet-hopping via a teleport device that asks for a complex series of coordinates for each planet. Your reward for a successful mission? One million dollars. Such a grand interplanetary scale of play is more appealing, at least to this reviewer, than scenarios confined to a few claustrophobic crypts.

Like all good adventure games, *Gruds* gives up its goods grudgingly. One of the first rooms you walk into is fitted out with a keyboard and screen—but learning how to use that keyboard is not at all obvious. Note: You will need a color monitor for this game, not only because the graphics show good use of the spectrum, but because you have to choose between different colored buttons on control panels featured in the play.

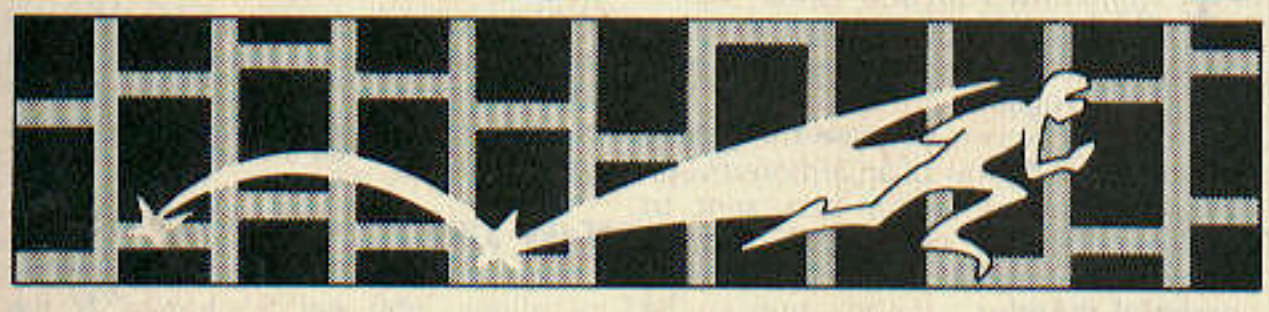
This game has one merciful feature designed to cut down on frustration: It shows the available directional exits in



the upper right-hand corner of the screen. It should keep you from having to put up with a barrage of CAN'T GO THAT WAY messages.

As in any good adventure game, you should be ready to commit considerable time to playing it. Be sure to keep a detailed map of your progress, and save your game on a separate diskette. If you are ready to immerse yourself in such play, you will find *Gruds In Space* one of the top-shelf programs of its kind. Even hardened adventure fans will find this game sufficiently entangling to make it playable for weeks. . .or years?

HCM



JUMPMAN JUNIOR

A Review by Judy Sanoian
HCM Staff

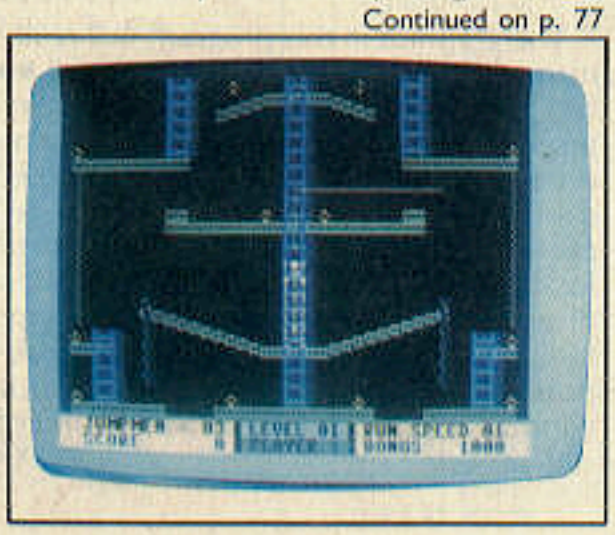
Name:	**Jumpman Jr.**
Program Type:	**Ladder Climbing**
Author:	**Randy Glover**
Machine:	**Commodore 64**
Distributor:	**Epyx Computer Software 1043 Kiel Court Sunnyvale, CA 94086**
Price:	**$39.95, cartridge**
System Requirements:	**Joysticks**

	Poor	Fair	Good	Excellent
Performance				
Engrossment				
Documentation				

If you're an old-timer in the computer game game, the first screen of *Jumpman Jr.* will immediately ring a bell. The blinking character scurrying up the rungs of the ladder, the dramatic leaps from one tier to the next. . .all that's missing are the hairy ape and the fair maiden. But *Jumpman Jr.* can't be shrugged off as just another *Donkey Kong* rip-off. For one thing, it offers a completely different scenario; we're defusing bombs instead of rescuing bombshells. For another, its sound is superior: harmonious tunes and crackling fires instead of quick blips and bleeps. *Jumpman Jr.* also excels when it comes to graphics. Compare the action of the falling jumpman—who tumbles head over heels, bouncing and ricocheting off tiers and ladders on his way down—with the quick splat in *Donkey Kong*.

For the game player, however, the important question is not whether *Jumpman Jr.* resembles *Donkey Kong* or the legions of *Kong* look-alikes. What really matters is whether it is an exciting game in its own right. I found *Jumpman Jr.* challenging, fun to play, and visually and aurally attractive to boot. Its animation and response are quite good, and

Continued on p. 77



VIC-20

```
370 DATA 7,31,127,255,255,127,63,15,255,255,255,255,255,255,255,255
380 DATA 224,248,254,255,255,254,252,224
390 FORN=8142 TO 8185:POKEN,13:POKE N+30720,5:NEXTN
400 FOR N=38400TO38861:POKEN,0:NEXTN
410 FORN=8062TO8066:POKEN,6:NEXTN
420 SC=0:TN=0
430 TK=8108:TC=10:RK=TK:POKETK,9:FORN=12TO14:POKE8117+N,N:NEXTN
440 PH=INT(RND(0)*12)+2:DR=INT(RND(0)*2)
450 IF DR=1 THEN GOTO470
460 I=0:J=21:C=1:GOTO480
470 I=21:J=0:C=-1:GOTO480
480 FOR CM=ITOJSTEPC
490 PP=(7680+CM+(22*PH))
500 IF DR=1 THEN POKEPP,11:POKEPP+1,32:GOTO520
510 POKEPP,10:POKEPP-1,32
520 IFF=0THENFORD=1TO35:NEXTD:GOTO560
530 POKE RK,32:RK=RK-22:POKE RK,9
540 IF(PH-1)>(RK-TT-7680)/22THENPOKERK,32:RK=TK:F=0
550 IFRK=PPORRK=PP+22ORPP=RK-CTHEN2000
560 K=PEEK(197):IF K=64 THEN 630
570 IFK=41THENGOSUB3000
580 IFK=18THENGOSUB4000
590 IFK<>49THEN630
600 IFF=1THEN630
610 IFPEEK(TK-44)<>32THEN630
620 F=1:RK=TK:RK=RK-22:POKE RK,9:TT=TC
630 Q=INT(RND(0)*500)+1
640 IFCM=TTTHEN660
650 IFQ>DFTHEN 770
660 IFQ>DF*20THEN 770
670 IFPEEK(8054+CM)<>32THENTR=(8054+CM):GOTO 710
680 IFCM<8ORCM>12THEN 690
690 IFCM>(TC-2)ANDCM<(TC+2)THENHT=1:TR=TK+1:GOTO 710
700 TR=8141
710 POKE36877,230:POKE36878,15:FORN=PP+22TOTRSTEP22:POKEN,8:NEXTN
720 FORN=PP+22TOTRSTEP22
730 IFN=RKANDF=1THENF=0:RK=TK
740 POKEN,32:NEXTN
750 IFHT=1THEN 800
760 POKE36877,0:POKE36878,0
770 NEXT CM
780 POKE PP,32
790 GOTO 440
800 POKE36877,200:TN=TN+1:HT=0
810 POKETK+30720,2:POKETK,7:FORN=TK+21TOTK+23:POKEN+30720,2:POKEN,7:NEXTN
820 FORD=1TO200:NEXTD:POKE36877,220:FORD=1TO500:NEXTD:POKE36877,0:POKE36878,0
830 POKE TK,32:POKETK+30720,0:FORN=TK+21TOTK+23:POKEN,32:POKEN+30720,0:NEXTN
840 POKEPP,32:POKERK,32:F=0:IFTN<>3THEN430
850 PRINT"{SHIFT CLR}":RESTORE:POKE36869,240:PRINT"{5CRSRDOWN}YOUR FINAL SCORE:";SC
860 PRINT"{3CRSRDOWN}PRESS RETURN TO CONTINUE"
870 GETA$:IFA$=""THEN 870
```

VIC-20

```
880 IFA$<>CHR$(13)THEN 870
890 GOTO220
2000 IFRK=PPTHENPOKERK,7:POKERK+30720,2:GOTO2020
2010 POKERK,32:POKEPP,7:POKEPP+30720,2
2020 POKE36878,15
2030 POKE36877,200:FORD=1TO750:NEXTD
2040 POKE36878,0
2050 POKEPP,32:POKEPP+30720,0:POKERK+30720,0
2060 IFPH>10THENSC=SC+5:GOTO2100
2070 IFPH>7THENSC=SC+10:GOTO2100
2080 IFPH>4THENSC=SC+15:GOTO2100
2090 SC=SC+20
2100 PRINT"{HOME}          {7SHIFT CRSRLEFT}";SC
2110 F=0:RK=TK:GOTO440
3000 IFTC=1THENRETURN
3010 POKETK-1,9:POKE TK,32
3020 FORN=12 TO 14:POKE (TK+N+8),N:NEXTN
3030 POKE TK+23,32:TK=TK-1:TC=TC-1:RETURN
4000 IF TC=20 THEN RETURN
4010 POKE TK+1,9:POKETK,32
4020 FORN=12TO14:POKE(TK+N+10),N:NEXTN
4030 POKE TK+21,32:TK=TK+1:TC=TC+1:RETURN
```

HCM

Flak Attack . . . from p. 61

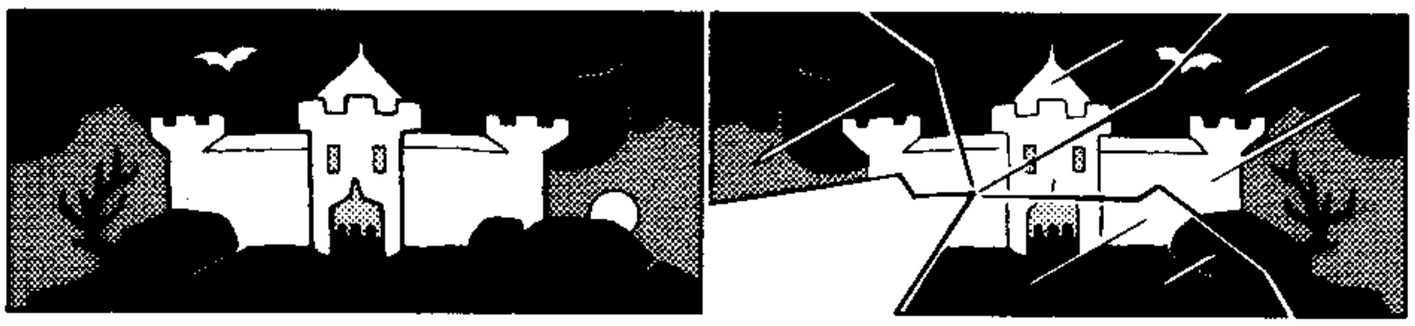
Flak Attack (IBM PCjr) Explanation of the Program

Line nos.	
100-180	Program header.
190-290	Display menu screen and input player's choice.
300-530	Set up string variables for shapes to be drawn.
550-650	Set up screen, and initialize flight of plane.
650-770	Input player's moves via keyboard.
780-830	Move plane.
840-880	Initialize cannonball.
890-940	Move cannonball.
950-1040	Hit plane with cannonball, destroy plane.
1050-1130	Plane fires laser at tank.
1140-1180	Tank blows up from direct hit.
1190-1250	Option to play again.
1260-1290	End of game.

IBM PCjr

```
100 REM ***************
110 REM *  FLAK ATTACK  *
120 REM ***************
130 REM BY MARK MOSELEY
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.1.1
170 REM PC BASICA
180 REM PCjr CARTRIDGE BASIC
190 REM ***********
```

Continued on p. 72



THE COVETED MIRROR

A Review by Greg Roberts
HCM Staff

Name:	**The Coveted Mirror**
Program Type:	**Adventure game**
Authors:	**Eagle Berns and Holly Thomason**
Machine:	**Apple II, II+, IIe**
Distributor:	**Penguin Software Post Office Box 311 Geneva, IL 60134**
Price:	**$19.95, diskette**

System Requirements:
48k memory, 1 disk drive

	Poor	Fair	Good	Excellent
Performance				
Engrossment				
Documentation				

Here is another illustrated adventure game that exemplifies the trend in this genre towards color, and lots of it. The makers of *The Coveted Mirror* claim that this one features "animation in virtually every frame." In fact, the animation consists mostly of eyes rolling or fingers wagging in an otherwise paralyzed frame. In addition, nearly half of every screen is needlessly taken up by the game title and an hourglass. But that isn't so important when you consider how bright and well-drawn the illustrations are. What's more, the story is well-written and intriguing.

The village of Starbury was once protected by a kind wizard who drew his power from a dazzling mirror. Now the mirror is broken into five pieces, four of which are in the hands of the evil King Voar.The fifth piece is lost. If you, the champion of this game, can find the fifth piece, you have a chance at defeating Voar. If the evil one should get the remaining mirror piece, his power will be absolute.

You always start your quest before the throne of Voar. No matter what you say, his dark sensibilities will be angered, and you will end up in the dungeon. Throughout the game, you risk being sent to the dungeon, and the program keeps track of how many times you go there. Twenty-five is your limit; if you reach it, the game ends.

Once you find your way out of the dungeon and into the street, the scenery goes by quickly. The buildings and the people are wonderful to behold, but your path gets unimaginably convoluted. How are you to know when to pick up such objects as horseshoes or crystal balls? You may find out too late that you need them. If you slip up and find yourself again in front of King Voar, you will lose the treasures you've picked up.

The many people you meet on the street can be most helpful. The simple commands TALK and HELP can be useful here. The game shows its flexibility in bringing on different characters each time you return to the street. *The Coveted Mirror* may be confusing, but you could never accuse it of being static. And even when the program refuses to give you what you want, it may at least give you a smile with with one of its smart aleck replies. For example, in trying to get out of the dungeon you may try to grab the ring that hangs on the wall. The computer will respond, THIS IS NOT A MERRY-GO-ROUND.

As you travel deeper into the medieval countryside, you find games within the game. A good one is the Sir Local Yokel jousting event, an arcade-type game within your adventure quest.

At this stage in gaming history, it is hard to think of an adventure that offers much more than *The Coveted Mirror* does. It should become a classic. **HCM**

Send in Your Photos and Anecdotes

Do you have a favorite photograph (color or black and white) featuring an unusual application of your Home Computer? Would you like to share your unusual or amusing anecdotes relevant to Home Computing? *Home Computer Magazine* will pay $25 for items it publishes. Material chosen will be subject to the same copyright treatment as "Letters to the Editor" as set forth on the Masthead page. No submissions can be returned. Send anecdotes and copies of photos to: Potpourri Editor, *Home Computer Magazine*, 1500 Valley River Drive, Suite 250, Eugene, Oregon 97401.

BASIC

Slots

by Bob Stoffers

It's your first day in Las Vegas, and you're a bit intimidated by the hubbub of an honest-to-goodness, big-time gambling casino. Deciding that the slot machines look tame enough for a grandmother like yourself, you timidly put a nickel into one of them. The woman playing next to you suddenly gets very angry and turns toward you. In almost one motion she pushes you aside, pulls down your machine's arm herself, reaches into the bag on her arm, and hands you a nickel.

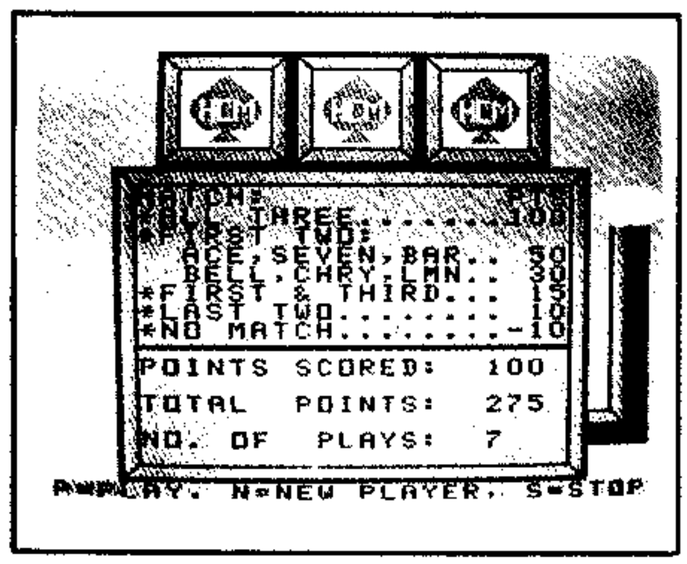
Shaking your head and clucking a little over this real-life exhibition of the gambling frenzy you've only read about, you move on—to another row entirely—just to be on the safe side.

The machines in this row are different. They look so modern, like TVs with typewriter keyboards. A nice young man tells you to keep your nickels. He types the word RUN and presses [ENTER] for you. Some writing on the screen explains that you get points if the symbols in the machine's three windows match: 100 if all three windows match; 50 if the machine's first two windows show aces, 7's or bars; and 30 if they show cherries, lemons, or bells. You get 15 points if the first and the third windows match, and 10 points if the last two match. If each window shows a different symbol, you lose 10 points.

GAMEWARE

BUFFET

You worry that you won't remember everything you've read, but the next display is reassuring. At the front of the slot machine is a summary of it all—both the scoring system and the instructions for play, new player, and stop. You press P—it's so easy, even a grandmother can do it. The slot machine's display shows how many points you've won with each play,

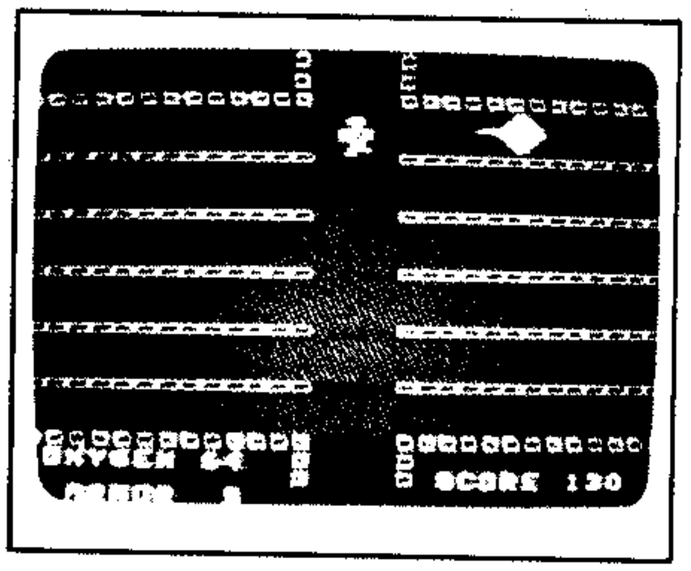


the total points you've accumulated, and how many plays it's taken you to get them.

There doesn't seem to be any system or temperament to it. You can't feel when the machine is "ready" to deliver a jackpot as the lady in the next row believes she can with her machines. It's always different and unpredictable: You might get 100 points with one press of P twice in a row, or you might end up with a mere 10 points for 10 plays. When you do hit the jackpot, you feel amply rewarded—not with cascading nickels, it's true, but with dazzling lights and an appropriate fanfare.

You know the grandchildren would love it too, but you're not about to bring them to a gambling hall. On the way back to the motel you resolve to talk your daughter into buying one of these home computers so you can key in this game for the children.

Continued on p. 190



EXTENDED BASIC

Meltdown

by Steve Langguth

To be awakened at three o'clock in the morning by a loudly ringing telephone is never a pleasant experience. And when you are the chief trouble-shooter for the world's first breeder reactor, you can be fairly sure that the news is going to be bad. This time, the voice on the other end of the line sounds even more rattled than usual. All you can make out is "bugs in the reactor," which is "approaching meltdown," and that you are the only person who can save the world from "total destruction."

Never one to pass up an interesting challenge, you race to the reactor site. Once there, you find that things are indeed as bad as you were told. Somehow bugs (cockroaches to be precise) have gotten into the reactor core and have fouled things up royally, so that at this very moment the reactor is running out of control. You must don your radiation-resistant armor, take your trusty "radiation neutralizer" in hand, and descend into the reactor core to clean up the subatomic particles which are floating about. Along the way you might even encounter some of those radioactive roaches which you must, of course, exterminate.

Unfortunately, a number of factors work against you:

1) The reactor core has five levels, with level one closest to the surface and level five deepest underground. The deeper you go, the more particles you will find which must be neutralized, and the more "energetic" the particles will be.
2) At all levels you will encounter gamma particles. These are a problem only when two of them come into your general vicinity at the same time. When that happens, they instantly become visible, fire off deadly gamma rays, and then disintegrate into nothingness. Luckily, they make distinctive sounds before they fire their rays, so you have a few seconds to get out of the way.
3) On levels two through five, particles of anti-matter will occasionally block your way. These are a problem only if they come into contact with normal matter (like you!). If that happens, the matter and the anti-matter tend to cancel each other out in a very large and violent explosion.
4) On levels three through five you will encounter the "nasty" roaches themselves. These little creatures seem to thrive on all that deadly radiation. In fact they have become so "energized" that it takes two hits from your neutralizer to exterminate them. (The first hit only makes them mad—and faster!)
5) Your supply of oxygen and radiation-resistant armor is limited. Spare oxygen tanks can be found along the way on all levels, but once the armor is gone, there is no more to be found.
6) And if all that were not enough, remember that the particles that you will meet in the tunnels of the reactor core are subatomic and therefore unpredictable. Our normal physical laws don't apply here, so be careful.

If you can successfully work your way through all five levels, you will be able to get back to bed. If you are not successful, it doesn't really matter because you will not have a bed to go home to anyway!

The Game

Meltdown is an arcade-style game for one player. After you load and run the program, you are given the choice of either joystick or keyboard to control the hero and the neutralizer. After you make that choice, the playing area will be drawn and the game will begin. Using the arrow keys or the joystick, move the hero character up or down through the central tunnel connecting the top and bottom of the screen. Fire your neutralizer into the corridors by pushing the joystick left or right or by using the left or right arrow key.

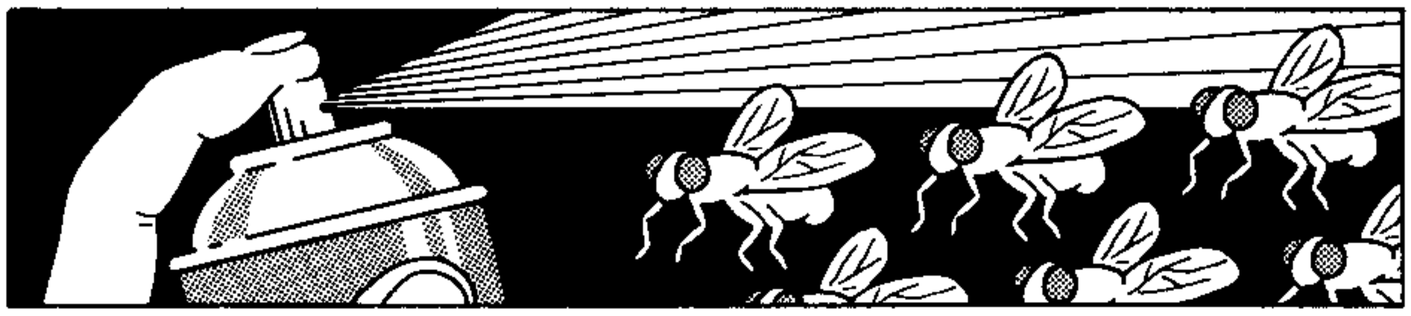
Neutralizing a particle is worth ten points, and allowing one to escape deducts five points from your score. Hitting a roach once gives you five points, changes the roach's color to white, and increases its speed. Hitting the same roach a second time counts for twenty points and neutralizes that roach.

Allowing a particle, roach, or gamma ray to touch you will cost you one of your five suits of armor. If you lose all five suits, the game is over. You begin with one hundred units of oxygen that get used up gradually as the game progresses. When a spare tank appears, touching it will transfer more oxygen to your supply. (Of course you have to stay in contact long enough for the transfer to be made!) If you should be unlucky enough to run out of oxygen, the game ends.

Anti-matter particles will occasionally block your path. Avoid them at all costs because touching one destroys everything—including you and the reactor.

There are five levels of play. As you descend deeper into the reactor, the number of particles you must face increases, and each particle moves faster than those on the level above. Only if you have the skill to survive all five levels will the reactor and the world be saved. Best of luck!

Continued on p. 110



FLY SNUFFER

A Review by Judy Sanoian
HCM Staff

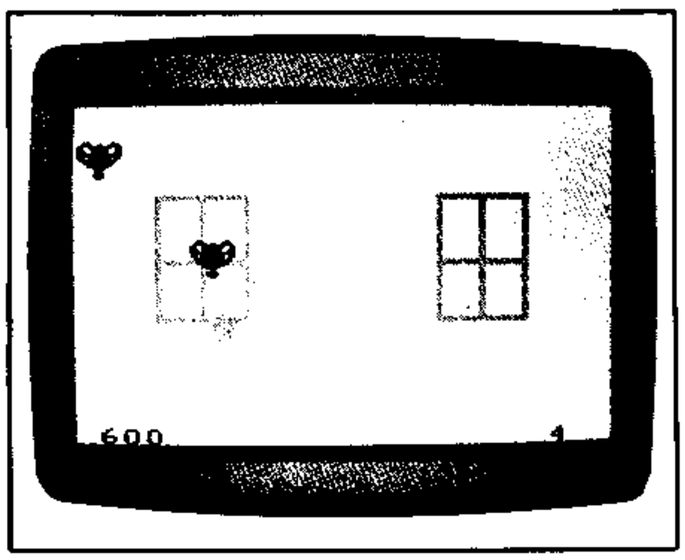
Name:	**Fly Snuffer**
Program Type:	**Pest Control**
Author:	**Larry Lewis**
Machine:	**TI-99/4A**
Distributor:	**Futura Software** **P.O. Box 5581** **Fort Worth, TX 76108**
Price:	**$19.95, cassette**

System Requirements:
Cassette Recorder
Extended BASIC Cartridge
Joysticks

	Poor	Fair	Good	Excellent
Performance				
Engrossment				
Documentation				

Anyone old enough to remember the game *Pong* knows that fancy graphics and complex scenarios are not necessary for an entertaining video game. *Pong*, a precursor of our modern cursor entertainment, featured a moving dot which players batted back and forth between simple vertical lines. *Fly Snuffer*, an offering from Futura Software, is a game formed in the *Pong* mold, although its graphics are not nearly so primitive. Your main goal in *Fly Snuffer* is to exterminate a horde of flies with a deadly aerosol spray. It is really just another variation on the hit-dot-with-stick theme, and yet, like *Pong*, it is surprisingly engrossing.

The opening screen of *Fly Snuffer* displays a room with two windows in the center. One by one, the flies enter from all sides of the screen. Using your joystick, you stalk the metamorphosed maggots, pressing your fire button to release the toxic gas. Kill ten flies and you win a round. But flies are not the only insects to invade your abode. There are also large orange cockroaches that scurry across the floor, and menacing bumble bees. To kill the roach, you must smash him with your can. The bee succumbs to your poisonous spray, but don't be slow in killing him. He is one of the long-awaited killer bees who have finally arrived from South America. If you don't vaporize him, you will fall prey to his deadly stinger.



The flies behave much like their real-life counterparts. Some are fast, some are slow, some buzz around the window trying to get out. Even though the little pests are quite harmless, you cannot take a live-and-let-live attitude. If you sit around marveling at this tiny member of God's creation, you will lose the game. You must gas ten of them before the time runs out. And don't go around spraying indiscriminately. Even if you don't care about our deteriorating ozone layer, keep in mind that you lose the game if you run out of insect spray.

Supersonic Cyber-flies

Fly Snuffer's graphics and animation are quite nicely done. The flies are fat, black and glossy with flapping wings. The cockroaches and bees also sport appropriate features—moving legs and menacing stingers—and the spray can emits a realistic puff of DDT. *Fly Snuffer's* sound effects are also well done: The flit of the flies' wings, the *pfft* of the aerosol can, and the crunch of the squashed roach are all quite realistic. At first I was disappointed that there was no buzzing fly noise, but on second thought realized that a constant cyber-fly buzz would be as annoying as the real thing.

The end of my first game of *Fly Snuffer* left me with a cramped right hand—always the sign of an absorbing joystick game. But even though it was exciting, it was also frustrating. Maybe I am exceptionally slow on the spray button, but I could not—try as I might—get past level three (of six levels). The flies on levels four and beyond are simply too fast to catch. Also the roach is too quick to smash unless you happen to be right in its vicinity when it appears. Perhaps staying low on the screen (where the roach crawls) should be part of your game strategy. All I know is that once I got to level three, I found myself darting in all directions, futilely spraying at the speeding flies while the roaches came and went as they pleased. And after 30 seconds in a room full of dive-bombing flies, I felt more inclined to wave the white flag than make use of the Black one.

The bottom line in any game review (or at least the bottom paragraph) is whether or not the game is any fun. In the case of *Fly Snuffer*, I must say it is. Playing to win is challenging and yet possible, up to a

point. . . and perhaps with practice you can even reach the highest level. The game loads easily and performs well if you follow its excellent documentation. (The instructions are among the most clear and complete I have ever seen.) *Fly Snuffer* is neither a visual masterpiece nor an exciting mental challenge. Like *Pong*, it is a good game to play when you want to give your cerebral side a rest. So if simple joystick action is your cup of tea, I'd recommend you give *Fly Snuffer* a try.

HCM

Jumpman Jr. . . . from p. 67

the scenario is exciting and varied enough to keep you at it until your joystick hand cramps in protest.

Space Chase Scenario

Although there are twelve levels of play and eight degrees of difficulty, your goal in *Jumpman Jr.* remains a simple one. As apprentice to Jupiter's secret agent, you must drive out the invading Alienators who have planted bombs throughout your substation. Using your joystick, you move up and down a network of ladders, ropes and elevators, working to defuse the bombs and at the same time, dodge the enemy bullets, fires and other dangers that threaten your existence. Your most effective means of movement (and your most thrilling) is the death-defying leap. By pressing your fire button and moving your joystick in the desired direction, you can execute a long, graceful jump that would put a gazelle to shame.

As soon as you complete your mission, the game moves on to the next level, where fresh obstacles await. It may take several tries just to figure out *how* to avoid or outsmart the flames, moving walls and electrocution traps that threaten you. Unfortunately, you are sent back to level 1 each time you lose. I would have appreciated a level selection option so I could skip right to the screen I wanted without having to work my way back through the preceding levels every time.

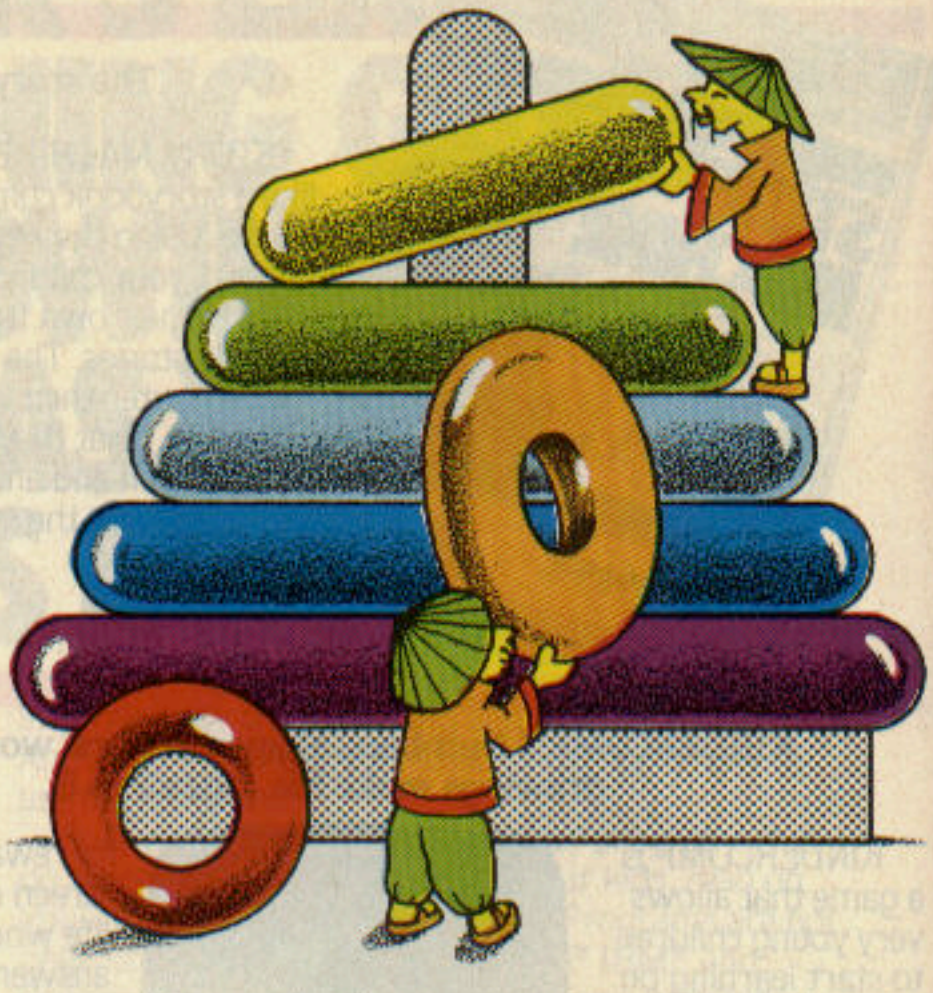
I may be a lightweight, but I also would have liked to see explanations of the various foes and obstacles (and the means of beating them) in the game's documentation. The instructions are otherwise complete, with ample descriptions of loading, movement and scoring. They describe *Jumpman Jr.* as a game of variety and suspense. I would add quick response, fast action, and superior graphics and sound to that description. I would also add a word of warning to the competition: *Donkey Kong*, watch out.

HCM

PLEASE DON'T FORGET TO RETURN THE QUESTIONNAIRE ON THE FRONT BIND-IN CARD.

Challenging the Tower of Hanoi

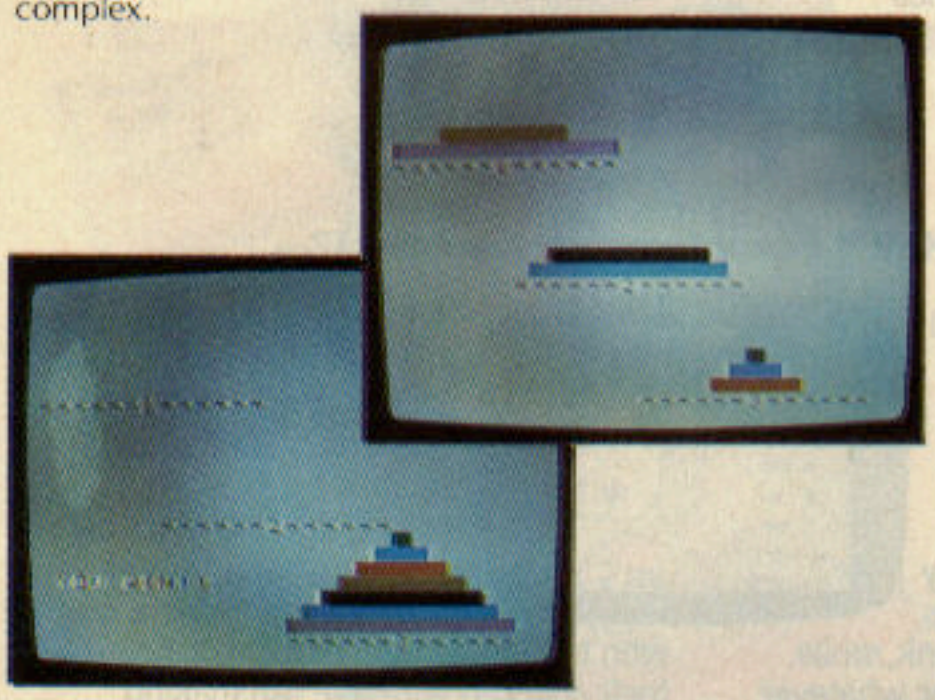
We at HCM have slightly modified Professor Holl's* Tower of Hanoi *(in the February, 1983, issue of* 99'er Home Computer Magazine*) so as to add more rings—and thus challenge—to the puzzle. We invite our readers to further modify this program by adding a routine that will automatically solve the puzzle and display the moves as they occur. If you develop a good routine, send it to us. We'll screen your results and publish a version for each machine.



You are in an ancient temple at the center of the earth, where three diamond needles bear eighty golden rings of graduated sizes. At the beginning of time the rings were all on one needle; but now the temple monks are transferring the rings, one at a time from needle to needle, never setting a ring on a smaller ring. When they have moved all eighty rings to one of the other two needles, the world will end . . .

Professor Holl solved the mystery of adapting this ancient Buddhist puzzle for the TI-99/4A using four rings, and now the Zen masters at *HCM* have discovered algorithms to expand the game to seven rings and convert it for the IBM, Apple IIe, Commodore 64, and VIC-20 disciples. We've also included a new expanded version for TI enthusiasts.

You may have seen something like the *Tower of Hanoi* in a baby's playpen—a set of multi-colored rings stacked on a wooden post. A child is supposed to advance from teething on the rings to trying to pile them up in size order, but your challenge in this deceptively simple brain-teaser is a bit more complex.



On the initial screen of this program, the seven colored "rings" (actually, they look more like short bars) appear on the left side of the screen, arranged in descending size order. The object of the game is to transfer the whole stack, one ring at a time, to another pile (the pegs are left out for simplicity's sake). You have three piles to work with, and one basic rule—you may never lay a larger ring on top of a smaller one.

Stacks

Now that you know how to play the game, let's think about the logic the computer uses to keep track of the stacks of rings. No matter what method is used, two conditions must be met. First, the rings must be kept in order within each stack. And rings may only be added to or removed from the top of the stack.

These conditions are easily met by representing a stack as a string of numeric characters. Each character represents one ring, and its numeric value corresponds to its size. When the game begins, the first peg (containing all seven rings) is "1234567", and the other two pegs are represented by the empty string (""). Keep in mind that the leftmost character is the top ring of a stack, and the numeric value of each character corresponds to the size of the ring.

Whenever we want to add or remove a ring, we must add or remove the leftmost character of the string. If, at the beginning of the game, you move the smallest ring from peg 1 to 2, peg 1 will become "234567" and peg 2 will be "1". Peg 3 remains empty (""). (The only exception to this is the TI version, which uses a floating point number instead of a string of numbers. The first stack is represented as 1.234567. The integer of the value represents the size of the top ring.)

The three stacks are stored as elements in an array called PEG$( ) with subscripts (1), (2), and (3). (This is PEG( ) on the TI version.) When you play the game you enter the numbers (1 to 3) of the peg each ring is to be taken from, and the peg that it is to be transferred to. These numbers are then used as the subscripts of the PEG$ array, to select the proper peg.

So far we have been talking about the part of the program you never see as you play the game. The computer's internal manipulation of the rings is actually quite simple and applies to all versions of the program. But to play the game, you need to see something on the screen. Because each computer has different methods and capabilities for displaying graphics, we'll give you a brief overview for each machine.

jr

In IBM's medium resolution graphics mode you have control of 64,000 points of light arranged in a grid 320 wide by 200 high. Keeping track of all those pixels is easier than you might expect. The LINE statement makes it easy to draw the different colored bars in the screen locations needed for the *Tower of Hanoi*.

In its simplest form, the LINE statement draws a line between any two points on the screen. You specify the points by naming the X and Y coordinates corresponding to the rows and columns. For example, LINE (0,0)-(319,199) would draw a diagonal line across the screen from the upper left to the lower right corner.

The LINE statement can also be used to draw a box and color it in. When we use LINE in this way, we use the same sets of coordinates, but this time they determine the opposite corners of the box instead of the endpoints of a line segment.

The first statement in line 210 draws the top ring of the tower in its initial position on the first peg. The upper left corner of this bar is 36,79, and the lower right is 48,85. The number 1 is the code for the color blue. BF indicates two things: B tells the computer that a box (rather than a line) should be drawn, and F says the box should be filled in with color.

While you're playing the game, the LINE statement in program line 410 draws the rings in any of the legal positions on the three pegs. Line 390 erases the old ring. You must indicate which peg the ring is being moved FROM as well as the SIZE and the vertical position of the ring, specified by TP(FROM). Line 410 uses TOO to draw the bar at the destination peg.

TOWER OF HANOI (IBM PCjr) Explanation of the Program

Line nos.	
100-160	Program header.
170-190	Turn off the KEY line, and set up the graphics screen mode as medium resolution with sixteen available colors.
200	Dimension arrays.
210-260	Display playing screen.
270-360	Input player's move.
370	Check for legal moves.
380-470	Move ring.
480-520	End of the game option to play again.

Continued on p. 86

The TI version of *Tower of Hanoi* uses the HCHAR command to display the rings. On the 99/4A, colors are assigned to character sets in groups of eight characters. Two characters are chosen from each set to make two rings. One of the characters is defined as a solid block of the foreground color. The other ring character is a blank and so displays the background color. The PC( ) array keeps track of which character is used for each of the seven rings. The PEG( ) array keeps track of which rings are on the three pegs. It does this with a number which starts out as 1.234567. When a ring is taken off of a peg, the integer that represents that ring is subtracted from PEG(FROM), the floating point number that represents that peg. The ring integer is then stored in the SIZE variable. PEG(FROM) is multiplied by 10, moving the decimal point one place to the right. It then has the value 2.34567, and SIZE becomes equal to 1. The peg that the ring is moving to, PEG(TOO), is multiplied by .1, and SIZE is added to that value. This way you can keep track of which rings are on which pegs, and the order of the stacks. The TOP( ) array keeps track of the top of each peg. This is necessary so that the computer will know where to place the peg.

TOWER OF HANOI (TI-99/4A) Explanation of the Program

Line Nos.	
100-170	Program header.
180-320	Initialize arrays, characters, and color.
330-440	Display playing screen.
450-560	Input the peg to move from and the peg to move to.
570-670	Move ring.
680-740	End of game message.
750	Game data for arrays and color.
760	End.

Continued on p. 87

Because the Commodore computers have no built-in "line-drawing" function, we wrote our own subroutine, which begins at line 1000. To draw the bar for a particular ring, the subroutine needs three pieces of information: the size of the ring being moved, the peg number it will be moved to, and the ROW of the stack where it will be displayed.

First the size (SZ) of the ring to be moved is determined in line 490 by getting the VALue of the leftmost member of the from string. Then, in line 570, this character is added to the left of the string selected by the T variable. Line 510 also uses SZ to determine the values needed to erase and draw the rings. Notice that the color code variable (CC) is used to choose the color and the size of the ring in line 250.

There is a slight idiosyncracy in Commodore BASIC which creates the need for line 580. When you use the STR$ function on a number to get its equivalent string value, a blank is added to the left side. If you then take the leftmost character (using the LEFT$ function) you'll get a blank instead of the 1 through 7 you expect. Line 580 strips these added spaces one at a time until the LEFT character is not a space.

Tower of Hanoi (C-64 and VIC-20) Explanation of the Program

Line Nos.	
100-160	Program header.
170-210	Dimension arrays, initialize variables.
220-310	Display initial game position.
320-470	Get input and check to see if it is legal.
480-580	Move rings and change arrays.
590-600	Check to see if game is over.
610-660	Play again routine.
1000-1040	Subroutine to draw rings.

VIC-20 continued on p. 88, C-64 continued on p. 87

For the Apple IIe version of the *Tower of Hanoi,* we selected the low resolution graphics mode because it is well-suited for putting bars of color on the screen. We used the low resolution command HLIN, which displays a row of block characters between specified columns at a specified row.

In line 410 we derive the size of the ring to be erased and moved. We use HLIN to erase the old bar (in line 430) by setting the color to black, the same as the background color. Then the F variable tells the computer which peg will have its top ring erased. The proper row is stored in the TP array. SZ specifies the width of the bar.

We use HLIN again in line 450 to display the bar in its new position. This time we use T to specify the peg. Because each ring is a different size, we use the SZ variable to derive different colors for each of the rings. Otherwise, line 450 uses HLIN in the same way as line 430 did.

TOWER OF HANOI (APPLE) Explanation of the Program

Line nos.	
100-170	Program header.
180-290	Display playing screen.
300-370	Input player's move.
380-400	Check for legal moves.
410-500	Move the ring from one peg to another.
510	Error routine.
520-550	End of the game option to play again.
560	End of the program.

Continued on p. 86

plenty of errors. But you can correct them quickly using Commodore's editing features.

Suppose you mistyped line 20 so that it read 20 FER K = 1 TO 440. In this case you could simply move the cursor over the E, replace it with an O, and then hit the [RETURN] key to re-enter the line.

But what if you typed 20 FOUR K = 1 TO 440? Now you'll need to use the [INST/DEL] key in the upper right corner of your keyboard. Move the cursor to the R, the position after the U, which is the character we want to delete, and hit the [INST/DEL] key.

There is yet another type of common error. You may have typed 20 FR K = 1 TO 440 and need to insert another character. This time, position the cursor over the R, hold down [SHIFT], hit [INST/DEL] to make a space, and then type in the missing O. Don't forget to re-enter the corrected line by pressing [RETURN].

Editing line 30 presents a small problem. Because the Commodore computers allow you to put cursor commands into PRINT instructions, they will not allow you to move the cursor to the left or right while you are in quote mode (following an open quote). If you try to edit between quotation marks, the computer thinks you are trying to put your cursor commands into the quotes. To get around this problem, press [SHIFT] and [RETURN] at the same time and then move the cursor back up to the line. Now you can insert any characters *except* the cursor commands. There is no way around this Catch-22, so you may as well hit [RETURN] and type line 30 again.

When you get these first four lines right, type RUN and enter it. If you did everything correctly, a colorful geometric pattern will be displayed on the screen, row by row. If you have any problems, type LIST, then press [RETURN] to see your program again. Check each line carefully, and keep editing until your program RUNs properly.

Now let's see how we can use the cursor keys inside quotes to make this display more intricate. First, type the word LIST and enter it to have a look at the program we just ran. The next lines we type will be part of the same program and will be performed after the geometric pattern is displayed. To maintain this sequence our next instructions should begin at line 50:

```
50 FOR L=1 TO 11:REM VIC 20
50 FOR L=1 TO 19:REM COMMODORE 64
60 PRINT "(SHIFT CRSRUP)2(SHIFT CRSRLE
   FT)(CTRL BLK)";
70 FOR M=1 TO 100:NEXT M
80 NEXT L
```

The 2 before SHIFT CRSRLEFT in line 60 is a subscript telling you to press [SHIFT] and [CRSRLEFT] together twice.

Enter these lines and RUN this latest version. We put line 70 in only to slow the computer down so that we could see how the cursor is moved up and to the left just before a black block is put on the screen. There is one more important key you should know about. As the black blocks "step" up the screen, hit the [RUN/STOP] key on the far left side of the keyboard. This stops the program dead in its tracks. As you learn to program, you will need this key from time to time to stop a program that has gone astray. To start again, simply type RUN.

We now have a stairway going up; we can just as easily put in a stairway going down:

```
90 FOR N=1 TO 10:REM VIC 20
90 FOR N=1 TO 18:REM COMMODORE 64
100 PRINT "(2 SHIFT CRSRLEFT)(CRSRDOWN)
   ";:
110 FOR M=1 TO 100:NEXT M
120 NEXT N
```

Study these lines carefully to make sure you understand how they work. But don't stop there. Experiment with new colors, shapes, patterns, and always try to have the computer do as much of the work for you as it can.

HCM

WHAT NOW?

QUALITY TI-COMPATIBLE PERIPHERALS WILL CONTINUE

32K Memory Expansion Card
RS232 Interface Card

SPECIAL INTRODUCTORY OFFER:
ORDER BOTH CARDS TOGETHER AND DEDUCT $10.00.....

CORCOMP is a former TI-99/4A supplier and its Model Masters Division is the developer of the TI Mini-Writer software and soon to be released action game "Intercept" on module.

Send Check or Money Order to: TEX-COMP • P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •

TEX-COMP™

TI USERS SUPPLY

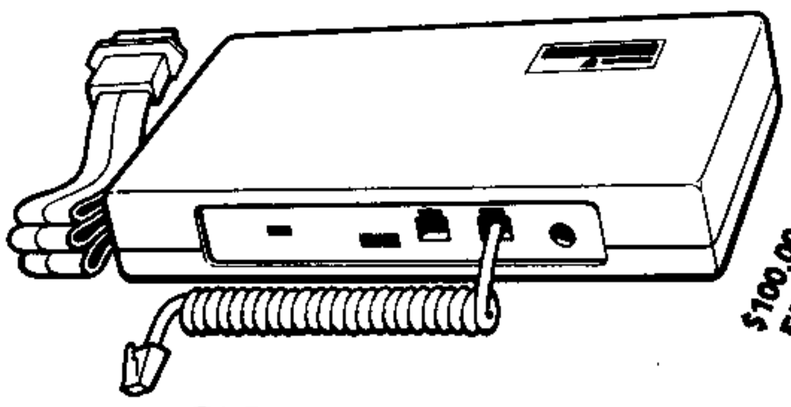
TEX-COMP is pround to announce that it has been appointed an authorized dealer by CORCOMP the leading manufacturer of TI compatible peripherals and accessories. As reported in InfoWorld and the Los Angeles Times, CORCOMP is now marketing a fully compatible 32K memory expansion card and RS232 card with a disk controller card to follow soon.

TEX-COMP is now taking orders on the 32K card and RS232 card for immediate delivery from stock. Both cards meet or exceed TI performance and quality specifications and carry a 12 month limited factory warranty.

32K Memory Expansion Card.....$119.95

RS232 Card.....$79.95

Visa and MasterCard
Order Direct
[818] 366-6631



$100.00 SPECIAL $100.00
FREE Source/TEXNET
sign on & first hour
with purchase
of Signalman MK III

Comes complete with cable and cord. This is a unit specially designed for the TI-99/4A and should not be confused with other Anchor/Signalman models that sell for less and/or are not compatible with the TI-99/4A.

SPECIAL: WE HAVE A LIMITED SUPPLY OF TI TERMINAL EMULATOR II WHICH CAN BE ORDERED WITH THE MODEM FOR $24.95. (one per modem please)

ACCESSORY: 9VOLT BATTERY ELIMINATOR FOR MARK III.....$10.95

SIGNALMAN™ MARK III
TI 99/4A COMPATABLE
MODEM

Finally, a low cost, direct connect, high quality and super reliable TI-99/4 and 99/4A compatable modem that comes complete and ready to use — just plug it into a TI RS/232 expansion card.

Mfg. Suggested List Price $139.00	**TEX-COMP PRICE** Limited-Time Introductory Offer **$94.95** Including shipping & insurance

Credit Cards add 3%

(818) 366-6631

MasterCard VISA

SUPER JOYSTICK II

The Only JOYSTICK that can control THE MUNCHMAN!

NEW Available exclusively from Tex-Comp

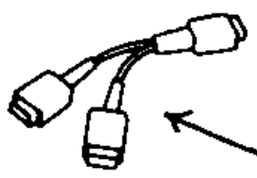
Rated #1 August 1982 Creative Computing

Improved version

The case has been redesigned for easy hand-held operation. The solid steel, short-throw control shaft with an arcade-style ball on the top offers precise, 8-position action. The "soft-touch" fire button is located at the top end of the molded base allowing either right or left-hand play. This improved fire button location is designed to take advantage of the greater dexterity of the index finger resulting in quicker response, time and less fatigue during extended play.

SAME ACTION AS THE REAL ARCADE GAMES

"SWITCHABLE GATE-PLATE™: Setting the switchable gate-plate indicator to the 4-way position allows only horizontal or vertical motion dramatically improving response for maze-type games.



Helps you make higher levels of Munchman
. . . if you're fast enough

NEW PRICE . 29.95

Free Shipping*
Second Unit Only $24.95
*Continental U.S.

With FREE Port Expander

TEX-COMP™

TI USERS SUPPLY COMPANY
"YOUR LUBBOCK CONNECTION™"

• P.O. BOX 33084, GRANADA HILLS, CA 91334 •

Authorized Dealer

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). East of Mississippi 4½%. (Free shipping on all software orders over $100 and on postpaid specials). Add 6½% S.T. for delivery in Calif. Prices and availability subject to change without notice. We reserve the right to limit quantities. *this ad supercedes all previous ads*

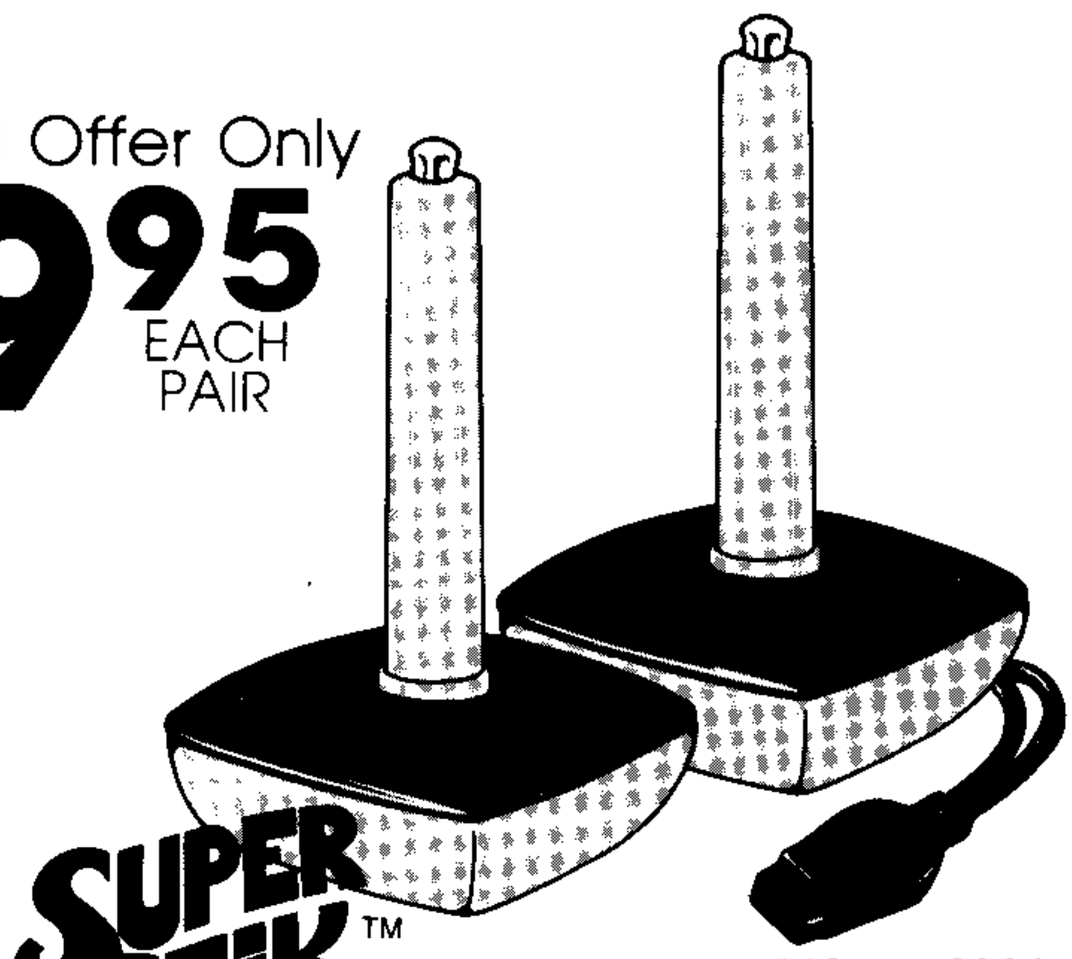
JOYSTICKS BUILT TO LAST!

MADE SPECIFICALLY FOR TI 99/4A
NO ADAPTORS NECESSARY

SPECIAL MAIL ORDER OFFERING

Limited Offer Only

$19.95 EACH PAIR



MODEL J1020

WORKS BETTER & LASTS LONGER

If you've tried to use other joysticks on your 99/4A, odds are they have not worked properly and/or broken down in a short period of time. Super Stik™ is the only joystick on the market (other than the TI joystick) made specifically for the TI 99/4A Home Computer with no adaptors necessary. What's more, Super Stik is made to last longer and work better than any other . . . backed by a one year limited warranty. No other joystick can match this claim.

WHY SUPER STIK IS BETTER

As mentioned, Super Stik is made specifically for the TI 99/4A home computer. This means that there is no need for special adaptors like other joysticks. A common 5' cord is included with each order.

The Super Stik Model J1020 has "built-in" isolation electronics that prevents "Crosstalk" between the two units. Without this isolation, joysticks will not function properly on the TI 99/4A.

J1020 units feature a simple, yet durable design with fewer moving parts. The housing is designed to completely stop handle movement before damage can occur to the contacts, providing a unique "positive stop" action. The assembly withstands a breaking strength of more than 50 lbs. and has been tested to more than 500 continuous hours without failure.

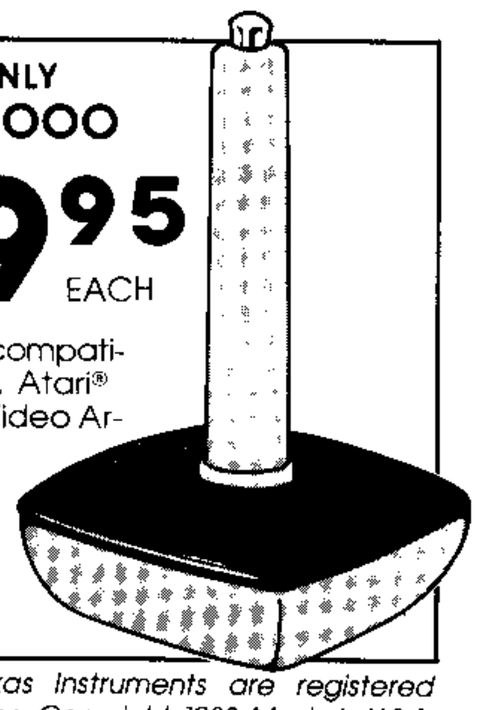
ARCADE FEEL & PERFORMANCE

Users of the Super Stik love its feel and performance. It features a long, comfort contoured handle, 8-way 45° angle control, and handle mounted firing button.

WITH THIS COUPON OFFER ONLY
SUPER STIK MODEL J1000

Also Available For Only **$9.95** EACH

The Super Stik model J1000 is compatible with Atari® Video Game, Atari® 400/800 Computers, Sears Video Arcade and Commodore VIC-20/C-64 Computers. The same rugged construction and super performance is offered. Don't miss the Super Stik at this price.



Atari, Commodore, Sears and Texas Instruments are registered trademarks. Super Stik is patent pending. Copyright, 1983. Made in U.S.A.

DON'T MISS THIS SPECIAL MAIL ORDER OPPORTUNITY

This may be your only opportunity to purchase the right joystick for your TI 99/4A at this low price. All orders will be allowed a 30-day money back guarantee if not **completely** delighted by its performance.

SEND YOUR PAYMENT WITH THE FOLLOWING COUPON TO:
RC DISTRIBUTING COMPANY
4016 Sanguinet
Fort Worth, Texas 76107

To Figure Your Payment For Each Joystick:

Super Stik J1020 (pair)	**$19.95** each
Super Stik J1000	**$9.95** each
Postage & Handling for Each Item Ordered	**$2.00**
For Texas Deliveries Add 5% Sales Tax,	**$1.00** for each J1020
	$0.50 for each J1000

Please rush my joystick(s) to the address below. My payment of $ _____ is enclosed for the appropriate Super Stik Model(s). If not completely satisfied with your purchase, return your joystick(s) within 30 days of receipt (in the box it came in) for a full and prompt refund. Offer good only while supply lasts.

☐ J1020 Super Stik Quantity _____

☐ J1000 Super Stik Quantity _____

NAME _____

STREET ADDRESS _____

CITY _____

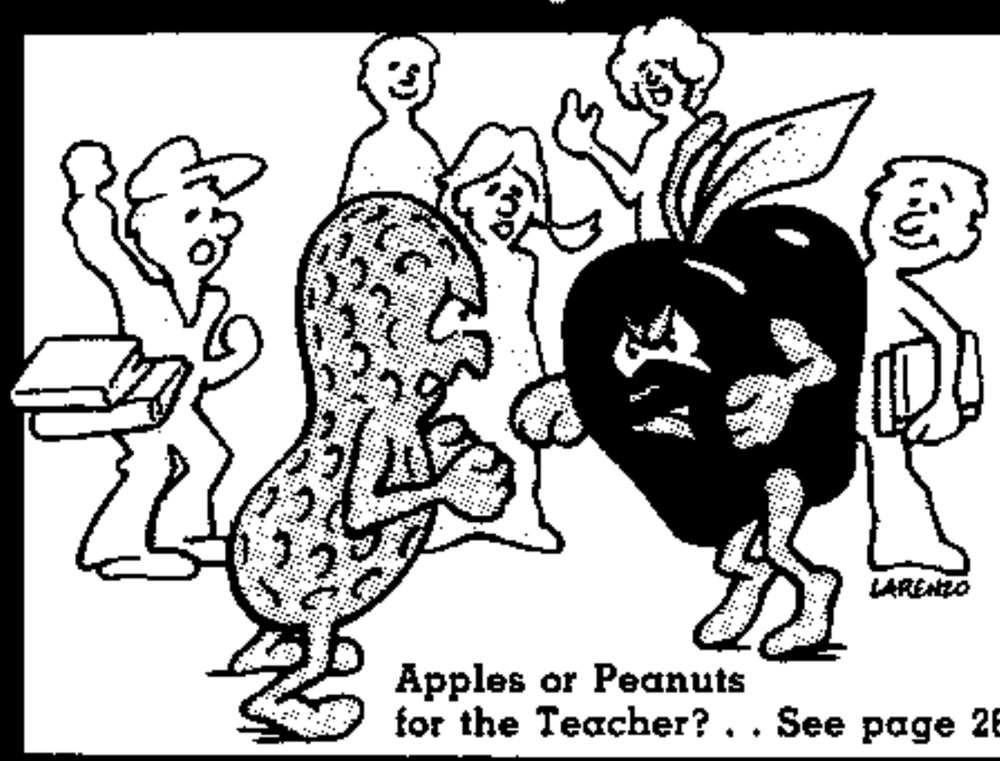
STATE _____ ZIP _____

Volume 1, Number 1 $1.50

HOME COMPUTER DIGEST

News and Happenings in the Home Computer World

- **Window Wars Rage On**
- **Home Computer Market Shake Out**
- **Las Vegas Show Stars**
- **TI PLATO: To Be or Not To Be**
- **Industry Standards Made in Japan?**



Apples or Peanuts for the Teacher? . . See page 26

CONTENTS

DEPARTMENTS

Publisher/Editor-In-Chief
Gary M. Kaplan

Executive Editor
David G. Brader

Acting Managing Editor
Sharyn Lyon

Features Editor
Greg Roberts

Technical Editors
Roger Wood
Peter Bloch

Assistant Editors
Erin O'Connor
Joan Killough-Miller

Research Editors
Deborah Amity
Dana R. Otis

Production Manager
Norman Winney, Jr.

Acting Design Coordinator
Larenzo

Production and Design
Larry Fisk
Eric Martin
Barbara Mickelson
Mohsen Zarrabi

Typesetting
Kathy Garcia
Julienne Laabs
Louisa Ada Hooven

Customer Relations
Denise McGuire
Tel. 503-341-1029

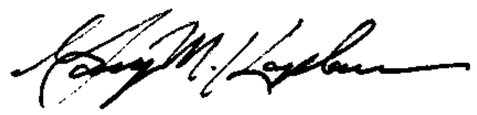
Bulk Distribution
Paula Hoiland
Tel. 503-341-1053

On Line

We are pleased to introduce the first issue of *Home Computer Digest*. This journal from Emerald Valley Publishing Co. is a companion to *Home Computer Magazine*, our long-standing resource for home users of the leading microcomputers. *Home Computer Digest* is designed to complement the practical, user-oriented nature of our larger magazine by offering readers a broader view of the ever-changing home computer industry.

Within these compact pages, we will cover all phases of the computer world. The *Digest's* regular columns—*New Tech News, Industry Watch, Software Trends, Novel Applications, Gameware Updates,* and *Computers in Education*—will compile essential industry developments in a concise, easy-to-read format. Each month, comprehensive feature articles will focus on important dimensions of the new technology to help you develop an in-depth understanding of current directions in the computer field.

Like *Home Computer Magazine*, this publication is written especially for you, the *home* computer user, and we welcome your input. We will consider carefully any questions or suggestions we receive from our readers, and we encourage those of you with inside sources or keen ears to send in relevant news for publication. Let us know how *Home Computer Digest* can help keep you well-informed in this dynamic, exciting Information Age.



HOME COMPUTER DIGEST is published by Emerald Valley Publishing Co., P.O. Box 5537, Eugene, OR 97405.

Any Questions?

The present global ignorance of computing may come, in part, from our natural aversion to asking simple questions—for fear of revealing only a shallow knowledge of vital topics.

Why not let someone else ask the questions while we sit back and benefit from the reply? That's the purpose of this column.

Q. Why do I need a data storage device? What kinds are there?

A. Unless you use your computer strictly as an entertainment machine, you will need to store your programs or data either on tape or on a disk system. The cheapest option is an ordinary cassette tape player. It is easily plugged into your computer and is fairly convenient for storing data; but sometimes it won't pick up the data unless the volume and tone are adjusted to a precise setting. In addition, taped information transfers slowly compared to other media. At this point, the most popular storage method is a disk system using floppy diskettes. These thin little wheels of mylar encased in plasticized envelopes were once a luxury for the few. Disk drives are now available for about $250, and $150 drives are in the works.

Q. I bought a "64K" computer, thinking I had a machine powerful enough for word processing and for running a small business out of my home. When I plugged it in, it turned out I had only 38K of memory available. What happened? Is this false advertising?

A. The claims of some computer manufacturers may be misleading, but they are not false. Your machine is endowed with 64K, but nearly half of that capacity is being used to carry out non-storage functions such as interpreting BASIC statements and running the hardware. Fortunately, some micros let you add on the memory you need; for $100 or so you can get extra memory in the form of a "card" that plugs into your computer console. Some users may be annoyed at having to make this extra purchase, but it may still be a cheaper option than the models that boast a large usable memory—and a big price to go with it.

Q. Sometimes I leave my computer monitor on for hours at a time. Can this practice harm the machine?

A. The back of your television screen or monitor is coated with phosphors which glow when bombarded with a stream of electrons. This electronic bombardment can be controlled to create precise images on your screen, but if an image is kept on for a long period, the phosphors of that pattern will burn out, leaving a permanent "ghost" on your screen. Some computers have an automatic switch-off device to prevent this problem—the machine goes blank if a key is not pressed within a certain period.

Q. What exactly happens when I transfer data from the diskette or tape to the computer? Can data be lost during transfer?

A. One of the sublime beauties of computing is that your disk drive or cassette player will keep your data safe, no matter how much you mess up the program loaded into the computer. Many people are unaware that loading a program from tape or disk into the computer merely copies the material on the disk into the computer's memory. This is analogous to playing a record on your stereo which

doesn't affect music stored on the record. Programs are destroyed only if you forget to record them, or if you take deliberate steps to erase them from the storage medium.

Q. We are concerned about getting a computer for our child as soon as possible. We know that he will need one in order to keep up with other children his age. The problem is money. How much will it cost to buy a computer for a five-year-old?

A. For many months now, massive ad campaigns have been urging parents to buy computers for their children so that they can "compete with the others." But it isn't clear what all these millions of youngsters are expected to *do* with their machines. It seems unlikely that many will become professional programmers. As for using a computer at work, it is not likely that great skill or long training will be necessary for most jobs—especially in years to come when computers will be extremely accessible. Computers can, however, be powerful educational tools. The LOGO language has proven very beneficial in helping kids learn math and logic. And a computer is great for helping children learn to read and spell. Fortunately, many such activities can be carried out on a simple computer system. An under-$100 machine attached to the family TV set and used with a cassette player is a perfectly good means of introducing your child to computing.

Q. I need a printer. What kind should I get?

A. It all depends on your needs. Basically, there are two kinds of printers: fast ones and slow ones. People who choose fast ones get a print-out with a "computerized" look. The letters are made up of dots printed out by a matrix of tiny pins striking the printer's ribbon. These dot-matrix printers for home systems can crank out copy at 50 to 200 characters per second. If you want typewriter-quality characters, you will have to sacrifice speed. A page printed on a daisy-wheel printer is indistinguishable from hand-typed copy, but such machines write at 12 to 55 characters per second. The dot-matrix printer is normally used for utilitarian data display such as program listings or department store sales slips. Letter-quality printers are generally used for business letters, press releases, and manuscripts.

Q. I ordered a printer through a mail-order house, and now that it's here I can't get it to work because my computer doesn't have what they call an RS232. What gives?

A. One of the curiosities and curses of buying computer products is that many items come strictly a la carte. The RS232 "card" is simply another collection of integrated circuits that conveys data from a computer's memory to a printer. This accessory can cost anywhere from $50-$200. In addition, you will have to spend $20-$50 for a cable to connect your printer to the RS232 card.

Window Wars

When Apple introduced the Lisa a year ago, it was met with less than the rousing reception Apple had hoped for, primarily because of its sky-high price tag. But Lisa's use of the screen as an "electronic desktop" caught the imagination of software developers and computer users alike. Extending the desktop idea to other computers is now becoming a major focus of the big software houses.

VisiCorp has been the pioneer in developing what is called an *integrated operating environment*. Designed to free the user from shuffling back and forth among programs and files for spreadsheet, graphics and word processor, *Visi On* opens up *windows* to several programs simultaneously. The user gains access to the system by moving a "mouse" (a sort of desktop joystick) across the desk. Using *Visi On*, the user can select a spreadsheet program and enter data, then move the mouse to select a graphics program to draw a graph based on the data just entered on the spreadsheet. Then, via the mouse again, the user can select the word processor and transfer the results to a report. This program interaction is all accomplished from the same screen, with no long delays between programs! The package was developed with the first-time user in mind, so there are no special languages or codes to learn.

What Price Windows?

One major stumbling block in this scheme of easy program and file interaction is the amount of memory required. VisiCorp hoped its *Visi On* would provide a window environment requiring only 128K of RAM and two floppy-disk drives. But the product as it is now being shipped requires a hard disk, a floppy-disk drive, a minimum of 384K of RAM (with 512K recommended), and the mouse device. The package lists for $495 without the mouse; the mouse adds $250 to that price. *Visi On* is available for several IBM PC-compatible machines and the DEC Rainbow, among others (as long as they have plenty of memory).

Microsoft's answer to *Visi On* — *MS-WINDOWS* (slated for release

in the second quarter of 1984) — will require only 192K of RAM and two floppy-disk drives. Because it is an extension of MS-DOS (so Microsoft claims), it will be compatible with nearly all existing application software that runs from MS-DOS. This is one-up on *Visi On* which supports only applications that are designed exclusively for use with the *Visi On* system. Microsoft expects the retail price on their package to range from $50 to $150, depending on the hardware supported, and maintains that *MS-WINDOWS* should work on any computer running MS-DOS. However, *Visi On*'s late appearance on the market and larger-than-anticipated memory and data storage requirements might cause an impartial observer to take a somewhat skeptical view of Microsoft's ambitious promises. We'll have to wait and see if *MS-WINDOWS* is the look of the future or just a promising silhouette on a shade that hasn't yet gone up.

The rest of the software world is not just standing by to see which of these two giants will emerge from all this window innuendo. The DesQ system from Quarterdeck (planned release was in December) already integrates existing programs in a window-like environment, but Microsoft's software may prove to be more flexible. Meanwhile, Digital Research (CP/M's developer) has developed an approach quite similar to the window concept with its Concurrent CP/M operating system.

Windows for the Home

So when can the home computer user expect to be looking at his programs through these marvelous windows? The Apple Macintosh, which was introduced January 24, has a mouse and windowing built in. At $2,495, it will compete directly with the IBM PC. Apple also recently announced an *AppleMouse II* package ($150 retail) to bring mouse software to the Apple IIe in March. In addition, a program called *Appleworks,* which integrates word processing, database management, and financial modeling, will be available in March ($250 retail). While this package does not incorporate windows as such, it does boast that "users can easily move information... from any file to a word processing file, using the program's 'cut and paste' functions." With Apple marketing these programs, can windows be far off?

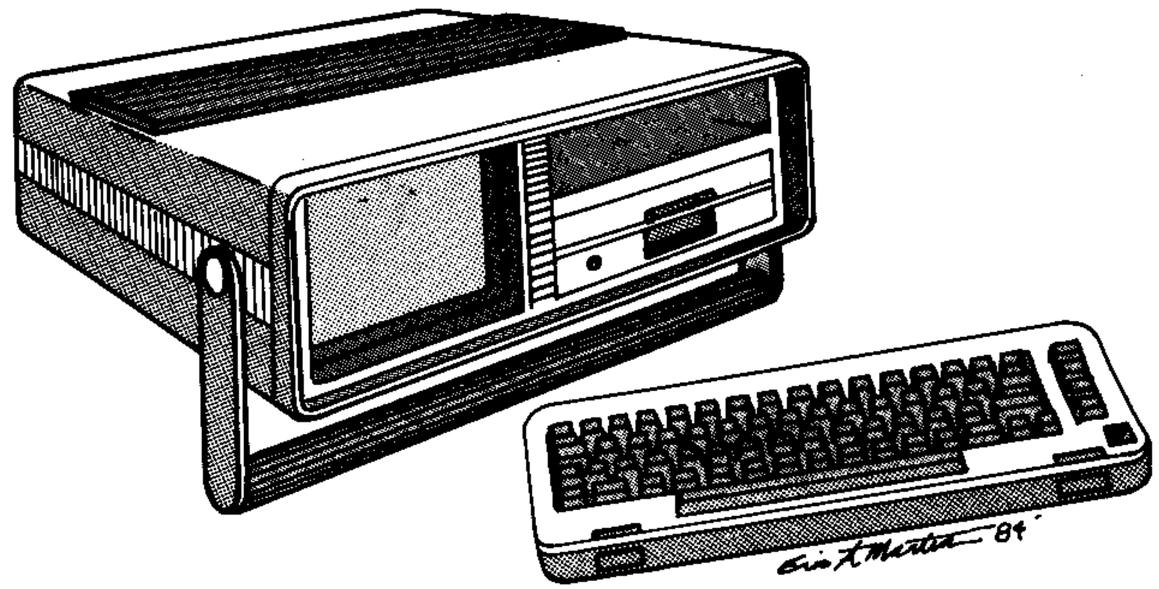
Commodore is also getting into the integrated software market. Their *3-PLUS-1* software package (a built-in option on their new home computer, the Commodore 264) boasts a windowing capability that allows simultaneous viewing of word processor and spreadsheet screens. Available as a ROM cartridge for the C-64 (and as an expansion for the 264), it will not require the large RAM of *Visi On* or *MS-WINDOWS*, but it will probably not be as flexible as these other software packages.

Software development is not restricted to large, established corporations. For example, a new company, Arktronics (established August, 1983, by two Michigan college students and the owner of the Golden Nugget casinos in Las Vegas and Atlantic City), is releasing an integrated package (February 1984) called *Jane* that runs on Apple II series and Commodore 64 computers. *Jane*'s developers claim that it could be modified to run on any computer with 64K of memory, and they hope to market versions for the IBM PC and PCjr in February 1984. *Jane* comes complete with mouse and runs with just one disk drive (Apple version) or cassette storage (C-64). This package includes word processing, spreadsheet, and file/list management, with more to come. *Jane*'s $295 price tag gives one pause, and we can hardly wait to see *Jane* RUN.

—Roger Wood

Sleeper of the Year:

The Commodore Executive



While everyone was buzzing with speculation about the Peanut, the Adam, and other "startling" announcements from the home computer front, Commodore came out with a breakthrough that has been kept surprisingly quiet, considering its importance. The Commodore 64 system has been bundled into a 5" × 14½" × 14½" brown box that includes every essential but a printer. The Executive features a detachable keyboard, 5-inch color monitor, disk drive, and 64K RAM. Now the big surprise: a price tag of $995. That's little more than half the cost of the typical portable system.

When the first Executive 64 press releases came out last fall, some of the machine's specifications put me on my guard. Would a 5-inch screen even be readable? And how substantial was that detachable keyboard? Wasn't 27.6 pounds a bit heavy for a new-model portable?

When I finally got close to the machine, I was suprised by the clarity of the monitor display. It was small, yes, but the resolution was better than I had expected. The screen's readability, however, depends on your eyesight. The tiny screen is simply the portable option—this machine can easily be plugged into a standard television or monitor.

The keyboard is a joy to type on. It seems even smoother and more responsive than the standard Commodore model. The Executive is the first low-end system that is both powerful and portable. It can be used as a writing machine or business tool at the office, then brought home at night and plugged into the family entertainment center.

This model takes advantage of the huge library of software written for the original Commodore 64, and even has a slot for cartridges. It is compatible with all Commodore and VIC-20 peripherals, including modems.

Despite the extremely sparse press coverage given to this new computer, the demand for it has been overwhelming. No one should be surprised. The public has long been waiting for a reasonably-priced, powerful, portable computer.

—Greg Roberts

CES SHOW STOPPERS

Short on Hardware—Long on Software

Even though many of last year's exhibitors didn't appear at this winter's Consumer Electronics Show—or showed up as registrants but didn't exhibit—there were some who came to Las Vegas with promising new products to unveil. The big story was Commodore's introduction of its new "dedicated" computers.

According to sources inside Commodore, enthusiasm for the long-awaited 128K, 16-bit Zilog 8000 has waned. Their excitement is now centered around the new series of 64K "Specialty Computers." Commodore Market Development reps say these computers—the 264 series—will carry a price tag in the neighborhood of $500, well below prices for the Adam and PCjr home computers.

These application-oriented "cousins" of the Commodore 64 with their built-in software were the talk of CES. Reportedly, users need only turn on a particular model and its specialty program will RUN automatically. Word processing and an integrated, consumer-level LOTUS-style package called 3-PLUS-1 were on view at the show. Commodore's 3-PLUS-1 will be a cartridge option for the C-64 and will be offered as a built-in or cartridge option for the 264. The functions of the package are simplified word processing, electronic spreadsheet, and data-base management; the "PLUS 1" is a business graphics program.

There was a lot of pre-CES talk about a 364 model with built-in speech synthesis and a numeric key-pad, but the machine was a no-show at the booth.

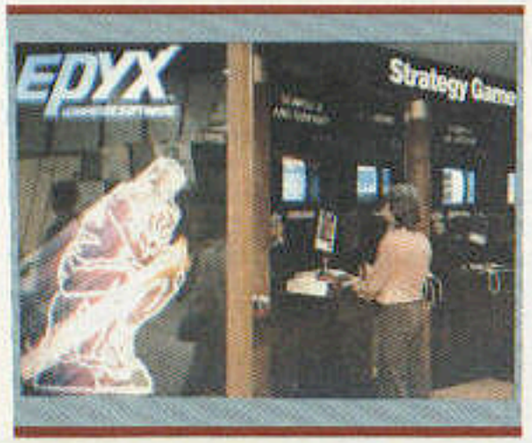
Even though these new-breed 64's look different from the original Commodore 64—sporting a stylish low profile with a streamlined tilted keyboard—they are compatible with most of the Commodore peripherals. Their software, however, will not be C-64 compatible because the new series uses a custom chip—the 7501 microprocessor—that is a combination of the 6502 and the 6510.

Commodore fans can also look forward to a built-in machine language monitor with 12 commands as well as more available memory. The new machines have a total of 64K of RAM with 60K RAM accessible through BASIC. This arrangement has more RAM available than the C-64 for users to PEEK and POKE about in to their heart's content! Other outstanding features include four separate function keys, an ESCape key, over 75 built-in commands, including RENumber and DELete, and graphics plotting using a special graphics mode. This mode uses all Super Expander Commands to give users a full 16 colors plus 8 luminance levels for a total of 128 working colors (just like Atari). The pixel resolution is the same as on the 64, but sprites are not available.

Apart from Commodore's new releases, this was a small producers' show. Timex introduced a 300-baud modem that may make a lasting industry impression with its automatic dial and answer features. A major attraction with this modem is free membership in the Source Telecomputing Network (a hundred-dollar value)! Another scene-stealer was the home version of Video Technology's Laser 3000. Yet another low-priced ($700) Apple-compatible computer, this one boasts four-channel sound, eight function keys, and built-in Microsoft BASIC. Although representatives claim that the Laser 3000 will run most existing Apple software, the user-jury is still out on whether this machine will take any sizeable bites out of big Apple's profits.

Software trendsetters on the scene at CES were showing titles that reflected a new focus. As industry analysts had predicted, arcade classic titles were outnumbered by games that required some strategy and even considerable thought.

Leading the way in this transition is Epyx, of *Jumpman* fame. Following a re-organization of their popular software line into four different categories—"action," "action-strategy," "pure strategy" and "learning fun"—Epyx representatives report that most of the company's new titles will be action-strategy games like *Pit-stop*. Not just another fastest-take-all race around a speedway, this game requires players to decide when to take a pitstop, what to do when they get there, and how much time to take. Each of these variables actually does affect the outcome of the game so that play will be different every time. This kind of software could provide users with a longer lasting challenge for their money.



While arcade-oriented publishers are putting more cognition than ammunition into their software, educational software producers are adding more action to their learning activities. CBS Software's "edutainment" line of learning games stresses "multiple-player, multiple cooperation" for users from preschoolers to adults, according to Ed Auer, CBS Software President. Look for *Strategy World*, *Design World*, and *Knowledge World* to debut soon. DesignWare is looking at the number two spot on the list of top ten producers of educational software with 24 action-learning titles.

Spinnaker, long-time leader in the educational marketplace, introduced *Trains*, an economics simulation package available on Apple, Atari,

cont. on pg. 15

Software Trends

CROSS FERTILIZATION

Simon & Schuster has joined the pell-mell rush of book publishers to jump into the computer age with the announcement of their Electronic Publishing Division to produce home computer disks. As could be expected, Simon & Schuster will concentrate on programming for education. The S&S book publishing operation has been up-dated with a new Computer Book Division. Barnes & Noble, whose main New York bookstore is the world's largest, will bow to the publishing trend with a separate store next door for computer books and software. B&N's Computer Book Shop and Software Center is a physical acknowledgement of the fast-increasing dominance of computer books in the publishing industry. Outstripping even the fiction market, computer book sales are taking over. *Publishers Weekly* has run features on computer book and software retailing weekly, and recently devoted a special issue to this hottest of all publishing topics. Software companies themselves are getting into the act. Microsoft Corporation, the big-time hardware and software company in Bellevue, Washington, will begin to publish books, many of which will be guides to Microsoft hardware and software products. It looks like it will be some time before computers kill off the age of Gutenberg, so widely predicted a few years ago. They may even have resuscitated it.

ALL THE TIME AND EVERYWHERE

ROMOX thinks the new reprogrammable cartridges they've just begun to market in convenience stores in California will become the most popular kind of software purchase. Retailers will lease ROMOX's reprogramming terminals for $160 per month, and customers will be able to choose and produce new software via electronic downloading on site at stores 'round the clock and—if ROMOX's ambitions pan out—'round the world. ROMOX plans to contract with one software distributor per country: with Japan's Aster Co. first, to be followed by distributors in Canada, England, France, Denmark, Sweden, Norway, and Finland. Domestically, ROMOX would like to persuade major companies such as Microsoft to distribute their software electronically through ROMOX's machines. Right now, ROMOX handles programs for the TI, Commodore, and Atari machines and has licensing agreements to electronically distribute programs from Creative Software, Epyx, Fox Video, Funware, HES, Mattel, Navarone, Sierra On-Line, Telsys, UMI, and Vidtec. Microsoft will be producing software for the IBM PCjr, and ROMOX would like to distribute those programs.

VENTURE-CAPITAL MUSCLE TO SQUEEZE THE LITTLE GUY?

If we're to believe Steve Axelrod, a "software agent" interviewed by Mary Alice Kellog for *ADWEEK*, the days of the kitchen table software outfit are coming to a close. According to Axelrod, marketing know-how accompanies the big-time venture capital that's now going into software companies. Axelrod foresees bigger and fewer software companies who produce sophisticated, highly competitive programs for several machines at once and who market their products with a skill and aggressiveness formerly reserved for hardware selling. As if to substantiate Axelrod's identification of a trend, TA Associates of Boston has just put $2 million venture capital into Alpha Software of Burlington, MA. This is the first time outside capital has gone into Alpha, and they've since come out with full-page glossy magazine ads.

STAR-STUDDED AFFILIATIONS

Rapidly expanding First Star Software has sold a non-controlling interest to Warner Software (a division of Warner Publishing, itself a division of beleaguered Warner Communications). First Star, whose best-selling *Astro Chase* for Atari game systems was authored by award-winner Fernando Herrera, has announced *Bristles*(tm), a new game by Herrera for the Atari and the C-64. Herrera, now Head of Design and Engineering for First Star, is notable as the first software author to conduct an autograph signing. The Toronto Film Festival sponsored an evening commemorating Herrera's work, and UCLA's Video Game Conference will honor Herrera this spring. First Star will pursue its interest in tie-ins with other media via independent film producers Richard Spitalny and Bill Blake, and in a joint venture with MARVEL COMICS Group to design and market software featuring Marvel Comics characters. Warner Software/Warner Books in the meantime will affiliate with LIST Magazine to produce a line of integrated computer book and software packages, its first for IBM PC business applications. Meanwhile, the parent company, Warner Communications, will try to fend off a takeover attempt by Rupert Murdoch, the controversial Australian newspaper baron who now controls major newspapers in England and the U.S.

MILTON BRADLEY-TI AGREEMENT

Milton Bradley and Texas Instruments agreed last year that MB would market their MBX Expansion System with three of the ten games coproduced by the two companies, and that TI would sell the other 7 games. This agreement was to have gone into effect during the first quarter of 1984. When TI left the home computer market, Milton Bradley and TI came to an amicable agreement rather easily: Manufacture of the MBX Expansion System has been terminated to the mutual satisfaction of both parties, and neither company has further obligations to that project. Milton Bradley, who had produced the MBX for only two weeks prior to the TI pull-out announcement, will not market the device, but has distributed the surplus through internal employee sales at both companies. Production of the software for the MBX had begun and will be completed. The three games that require use of the MBX unit (*Terry Turtle's Adventure*, *I'm Hiding*, and *Championship Baseball*) will be sold to company employees with the MBX units. The seven titles TI was to have marketed (*Honey Hunt*, *Sound Track Trolley*, *Space Bandit*, *Sewermania*, *Big Foot*, *Meteor Belt*, and *Super Fly*) have been shipped and will be sold internally as well as through TI's regular retail channels. The suggested retail price of the software has been cut from $49.95 to $29.95. When the current software inventory is gone, MB and TI will discuss future licensing arrangements.

TI Pullout

The Aftermath

A lead player walked off the set, a tardy prima donna pirouetted to center stage, and the rest of the troupe waited in the wings for their roles to be re-cast. Before the curtain had fallen on this end-of-the-year drama in the home computer world, industry analysts had filed their reviews and were braced for the big showdown. But the last act is still being written by consumers, who are clearly ad-libbing in this "who'll-buy-it" mystery.

Instability is nothing new in the high-tech market. But the concurrent demise of Texas Instruments' 99/4A and the unveiling of IBM's PCjr gave rise to predictions of an inversion of established buying patterns. Sales of low-end computers were expected to suffer from consumers' fears of being "orphaned" by the price wars that knocked out TI, whereas manufacturers of higher-priced models anticipated a boost from the reflected glory of IBM's impeccable reputation. Market analysts predicted an overall "wait and see" attitude, with consumers holding out for stabilization in the low end and postponing expensive purchases until the long-delayed arrival of the Peanut.

TI Loyalists and Converts

Instead, a wave of buying mania swept the country and left dealers struggling to keep up with the demand for lower-priced models. Heavy holiday sales depleted retailers' stocks as drastic price cuts stimulated new consumer interest in joining the ranks of home computer owners. "Fire sale" prices made TI wares irresistible to Christmas shoppers, who queued up at discount chains to purchase the remaining $149 consoles for a mere $49.

The estimated 2 million + TI user base is both a temptation and a frustration to third-party hardware and software suppliers. A number of manufacturers have expressed interest in supporting the 99/4A group, but these suppliers must seek alternate methods of distribution, as most retailers will not continue to carry TI compatibles. Only those companies that can convert to mail-order and access communication channels with TI users (e.g., mailing lists, users groups, and targeted publications) are expected to succeed.

Rumors of 99/4A look-alikes have surfaced, and many concerns are expected to cash in on the machine's popularity. But Texas Instruments may not be through with the home computer world for good. Some analysts believe that the corporation's cautious exit from the scene indicates a desire to slip back into the running at a future time. TI has been very protective of its distribution network, and even continued to support sales of the discontinued 99/4A through the Christmas season with at least half of the company's prepurchased TV ad time. Sales of other Texas Instruments products (hand calculators, educational aids, and the CC-40 portable computer) have been strong.

The Upshot Still Not Clear

IBM may have kept the public waiting too long. When the release of the PCjr was delayed beyond December's peak market, frustrated consumers who felt they had to have a micro under the tree turned elsewhere. Advance publicity for the Peanut may have over-inflated consumers' expectations; then a wave of cynical reviews hit, suggesting that the Peanut's capabilities might not justify its cost. Many sources began stressing the lack of compatible software for home applications. Executives with an IBM PC at the office might work at home in the evening on a junior, but garden variety home computer users will have to wait for IBM to woo them with software, or stick with micros that put more emphasis on the "home" in home computers.

Although IBM can be accused of neglecting the home market, its appeal to business and educational institutions has been strong. Virginia Polytechnic Institution and State University will require all entering engineering students to purchase a PCjr, PC, or XT. The College of Engineering at VPI contracted with IBM for 1,600 PCjrs sight unseen, and is willing to increase the order to 4,000 if the actual goods prove satisfactory.

Other high-end producers may profit from these year-end dynamics. Apple executives have stood firm against the tide of price slashing, discount distribution channels, and home entertainment marketing. Apple's sales approach still emphasizes user guidance and support, with sales through authorized dealers only. The delayed release of the Peanut may have created a windfall for Apple.

Flash-in-the-pan profits generated by the TI pull-out were apparent in the low-end market. Discount chains across the country reported record holiday sales of lower-priced models. Christmas shoppers stood a better chance of finding a Commodore 64 or VIC-20 on the depleted shelves, and both models moved quickly. Commodore's new line of 64K "Specialty Computers" and the Commodore Executive may be the reason Commodore did not raise prices, as did Coleco and Atari, who moved into the gap left by the TI's departure.

The sudden pull-out may have been too much for manufacturers at the extreme low end of the price scale, however. Doubts have surfaced about the longevity of manufacturers like Timex/Sinclair who fought vigorously against TI in the price slashing competition, but may not have enough of a computer to withstand market pressures. The clearance prices on the TI-99/4A have also infringed on sales of Radio Shack's TRS-80 color computers.

Advances in technology and high-powered advertising strategies will continue to sway a fickle public. But independent-minded consumers may still have a few surprises in store for industry experts who attempt to make predictions in this quickly changing market.

—Joan Killough-Miller

Novel Applications

NO SOONER SAID THAN DONE

New hardware developments will mean greater independence for handicapped users who may have difficulty with the standard computer keyboard. Voice recognition modules such as Voice Driver, designed by Voice Recognition Systems of San Francisco for use with the Apple II and IIe, allow system operation via speech commands without any special programming. The user's voice is imprinted by voice utility software, which can input a vocabulary of up to 80 words. The module can screen out ambient noise and performs with a recognition accuracy of 98%.

THE NEXT BEST THING TO BEING THERE

Rand McNally & Co. has gone beyond the boundaries of the conventional atlas with a line of educational games designed to teach geography, U.S. history, meteorology, and the time and seasons. The new skill-building software for the Apple II and Atari 800 computers is aimed at fourth to sixth graders. *Unlocking the Map Code* includes a "simulated flight plan" in which students pilot their way between world capitals managing risky terrain, limited fuel, and other flight obstacles.

WHAT NEXT?

Just in time for 1984 comes the Expando-Vision interfacing box by Stimutech, Inc. Commodore and Atari users can hook it up to their television sets and undergo self-improvement programs with subliminal messages flashed on the screen every 2½ minutes. The messages, which appear too quickly to be consciously noticed, are said to be "imprinted" on the subconscious to help the viewer reduce stress, lose weight, or stop smoking, for example. So now you can watch *Love Boat* and *The Dukes of Hazzard* and know that you will be a better human being for the experience.

BETTER HOMES AND CONSOLES

Look for new hardware and software systems that can manage home operations and act as central timers. Products are being developed to control house lights, give you a wake-up call, detect smoke, activate sprinklers, and even heat up your hot tub. Our editors have even seen a TI console turned into a burgler alarm. Now if they could only come up with something to get the kids off to school. . .

THE FAMILY THAT COMPUTES TOGETHER. . .

The Enchanted Village(tm) is a new concept in retailing that offers computer-related merchandise, in-store seminars, a library, and live performances—all dedicated to the concept of "Edutainment"(tm), a marriage of education and entertainment. Here customers can shop and learn in a family environment that even includes a supervised playroom. The first two outlets opened this fall in Pittsburgh, PA, and Fairfax County, VA, with a national chain slated for the future.

NO FLAB FLOPPIES

Good news, micro buffs. You don't have to leave your console to keep the old body in shape. Spinnaker has introduced *Aerobics*, the company's first venture into the adult software market. Led by a computer-generated figure called "Jane" (who else), this at-home fitness program offers a customized work out with challenging exercises and coordinated music. The program is available on diskette for the Apple, Atari and Commodore 64 computers.

SEND IN YOUR ANECDOTES AND JOKES

Do you have a good anecdote or joke to share? Don't limit it to the breakfast table or the office—Home Computer Digest will not only listen, but will send a $25 check to the authors of the ones we print. We are always looking for cartoons, bizarre news items, jokes and short anecdotes having to do with home computers, robotics, or any other aspect of the computer industry. Short items need not be typewritten. Be sure to put your name, age, address, and telephone number on each item you submit. Due to the large number of contributions received, none can be returned. Materials chosen will be subject to conditions set forth on the Masthead page of *Home Computer Digest*. Mail all submissions to:

Features Editor
Home Computer Digest
44 Club Road,Suite 210
Eugene, Oregon 97401

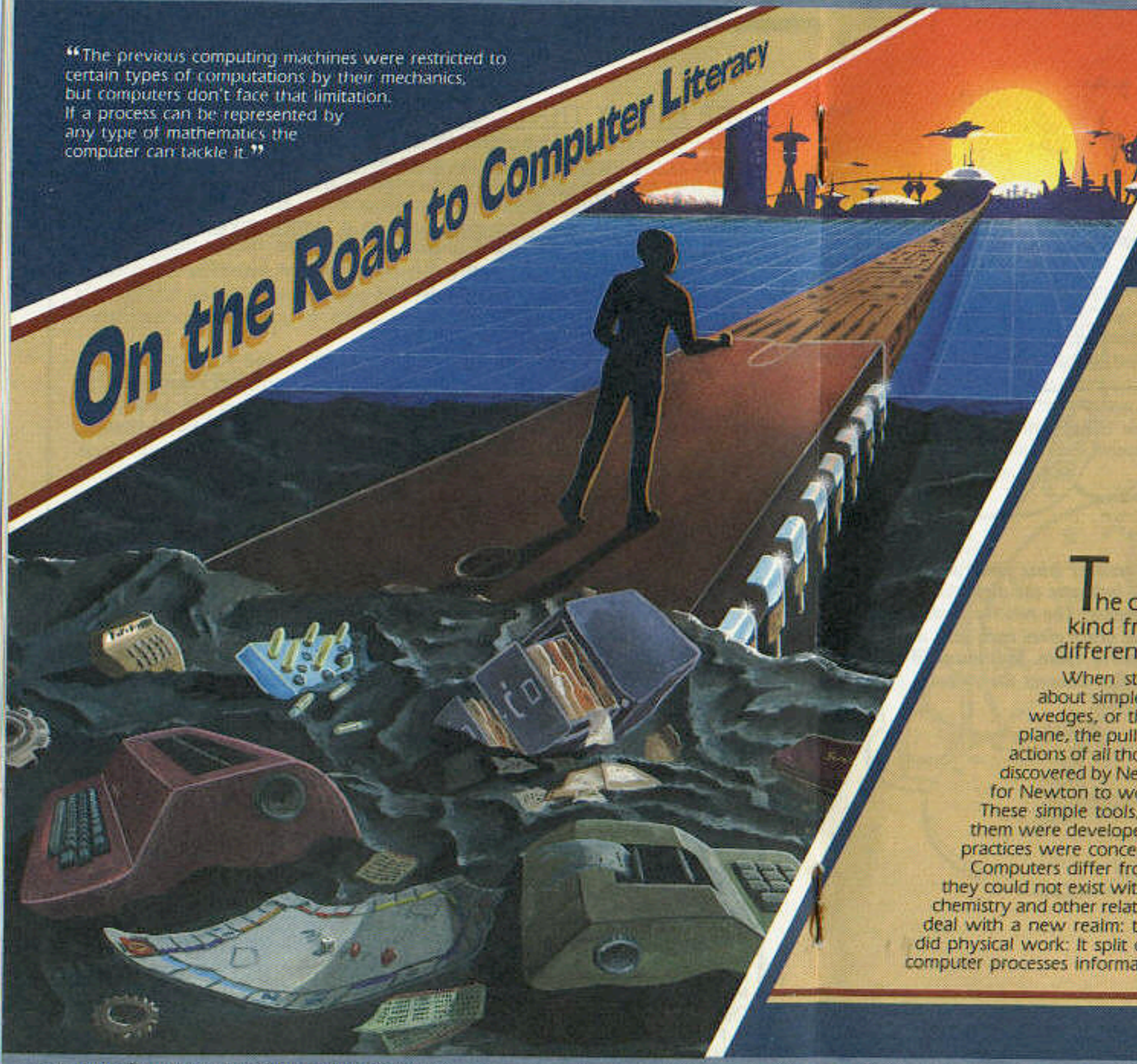
CES Show... cont. from pg. 9

and Commodore at a modest $39.95. With each player taking on the role of railroad supervisor, mental and physical action go hand-in-hand. Spinnaker leads the pack when it comes to putting *real* physical action into home computer software. Showgoers gave *Aerobics* a real work-out (or was it the other way around!). In fact, health and self-improvement packages abounded at WCES. With products like this one and Synapse's *Relax* (for the IBM PC, soon to be available on the Commodore and Apple computers), the phrase "Let's get physical" may take on new meaning for software developers in 1984.

—Sharyn Lyon

"The previous computing machines were restricted to certain types of computations by their mechanics, but computers don't face that limitation. If a process can be represented by any type of mathematics the computer can tackle it."

On the Road to Computer Literacy



COMPUTERS: *THE NEW TOOLS*

ANALOG VS. DIGITAL

Part 1

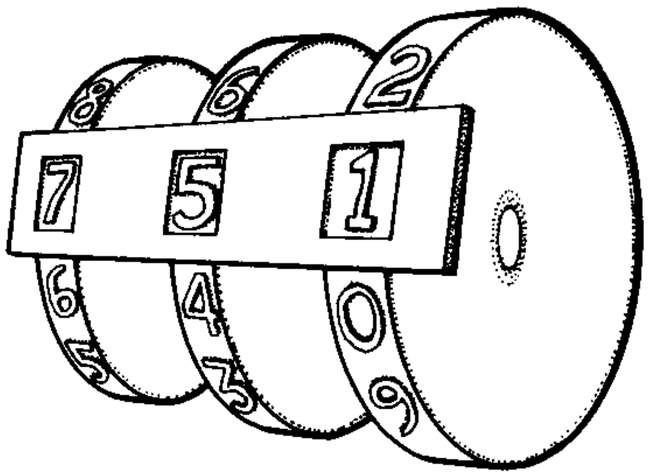
The computer is a machine. But it's a different kind from any other we have known so far—different in several important ways.

When stone age hunters made axes, they knew nothing about simple machines. They didn't realize that their axes were wedges, or that the wedge is a simple machine like the inclined plane, the pulley, the screw, the lever, or the wheel and axle. The actions of all those simple machines fall under the laws of mechanics discovered by Newton. Their inventors, however, didn't have to wait for Newton to work out his theories before they devised their tools. These simple tools, and other machines made from combinations of them were developed in practice long before the theories behind those practices were conceived.

Computers differ from those older tools in two important ways: First, they could not exist without numerous innovations in the fields of physics, chemistry and other related branches of science. But beyond that, computers deal with a new realm: the abstract rather than the concrete. A stone axe did physical work: It split open a billet of wood or an opponent's skull. The computer processes information: It splits numbers.

The manipulation of numbers is a relatively recent development in human history. We have tools from every period of history as far back as we have human remains. But the manipulation of numbers—mathematics—probably began only about the time recorded history began. And we can, in fact, date precisely a number of discoveries which had to precede the appearance of the computer.

Counting the Hours



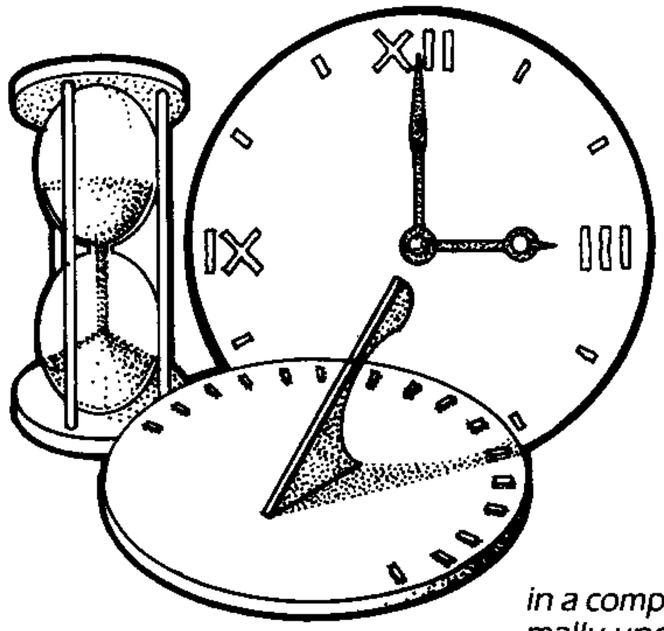
The number lying under a given mark indicates the digit that gear represents. The relative position of the gears gives each gear its respective value. Movement of the gears changes the values displayed.

There are older computing devices which also dealt in abstractions and also depended on a variety of these discoveries. Sundials, for instance, measured the progress of the day, and divided it up into hours. Hours, of course, are abstractions, arbitrary divisions of the day devised by humans. Once the division of the day into the periods we call hours had occurred, other methods for keeping track of these periods, methods entirely independent of the sun's movement, came into being: water clocks, hourglasses, mechanical clocks. . .

These mechanisms are all ***analog*** *devices. Analog devices use the position or movement of some physical object to produce their information. Sundials depended on the movement of the sun; water clocks and hourglasses on the movement of water and sand, respectively; mechanical clocks on the movement of springs, gears and pointers. The motion of the substance was, in effect, the computation; the physical location of the substance was the result—that is, the information.*

In the case of an hourglass, for instance, the movement of sand corresponds to the passage of hours in a day. And we read the results from analog devices by measuring a quantity of the substance—to read an hourglass, we have to be able to figure out what portion of the total sand in the glass has moved from one compartment of the glass to another. On some glasses, this is easier than on others because a scale is etched on the glass. Nonetheless, we have to look at the ***quantities*** *of sand.*

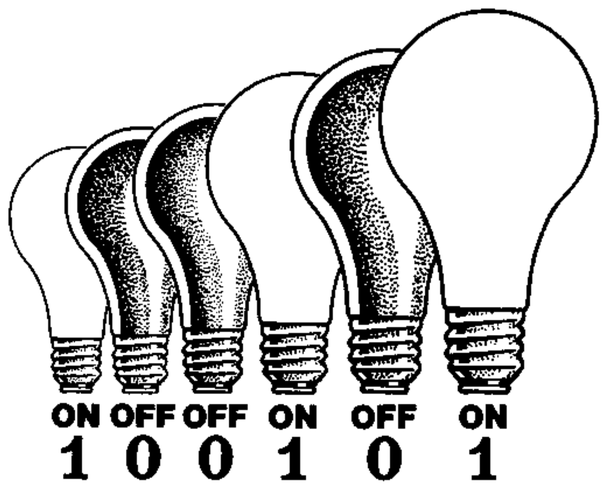
The shadow of the sundial's pointer marks the hour of the day. Hourglasses and mechanical clocks use the movement of sand and gears to compute hours. They measure only the passage of hours; when synchronized with the sun's movement, they denote a particular hour.



The States of Computing

Later computing machines, like Babbage's computing engines from the mid-1800's or the more recent mechanical adding machines, were also analog devices. In addition to the movements of physical objects, however, they relied on ***positional notation,*** *which is the basis for our everyday numbering system. Every number we see has a value which depends on the position and value of the individual digits in the number. When we see 751, we know that it represents seven hundreds, five tens and one one.*

Most computers, by contrast, are not analog devices, but ***digital*** *devices. Rather than having moving parts, they use electronic components and changes in these components to represent their calculations and results. (There is, of course, movement in a computer, but not any kind we normally understand as movement. It takes place in the realm of subatomic particles.) These components have a special characteristic: They take on* ***states*** *—i.e., they can take on only one of two states. In essence, either they are charged or they are not. Computers use this system because there is a form of mathematics that exactly corresponds to this dual-state concept.*



This mathematics, although developed over the years by a number of people, is most often associated with the work done by an English mathematician, George Boole. His system, called Boolean algebra, reduces mathematics to its very basics. Boole said that there are two basic elements and three basic operations that can define all mathematics. Usually, the basic elements are represented as 0 and 1. We can think of the operations as addition, multiplication and negation. With combinations of these elements and operations, we can represent all of mathematics.

Many electronic components are like light bulbs: They can only be on or off. They also can be interpreted as the digits of a number in their respective positions, but they can represent only two digits: 1 or 0, on or off. This numbering system is known as* binary *(from the Latin for two).

The same operations used in Boolean algebra to combine elements have exact counterparts in the computer world. These counterparts manipulate the states of a computer's electronic components. With this binary-state manipulation at the component level, we can also represent all of mathematics—electronically.

This opens a vast range of potential for the computer. The previous computing machines were restricted to certain types of computations by their mechanics; computers don't face that limitation. If a process can be represented by any type of mathematics, the computer can tackle it.

That immediately presents us with two difficulties central to our use of the computer: (1) finding an appropriate mathematical representation for the problem we're working on, and (2) finding a method for translating that representation into terms compatible with our particular computer.

DEVELOPING ALGORITHMS

The task will be to write a program instructing an all-purpose robot to change a lightbulb. First, spell out a step-by-step procedure for replacing a burnt-out lightbulb in order to see how to develop an algorithm, in English, for this seemingly simple task. The robot must have a complete and unambiguous set of instructions to follow.

At the very top level, the name of the program might be "Change the lightbulb in the kitchen ceiling." Rewrite this top level description of the algorithm to read: CHANGE LIGHTBULB(KITCHEN)

Then divide the problem, or task, into its main subtasks. There is no single correct way to divide up a task, although some ways will be better than others. Try:

1. GO TO THE (KITCHEN)
2. REMOVE OLD LIGHTBULB
3. NOTE WATTAGE OF OLD BULB AND DISCARD
4. GET NEW BULB WITH SAME WATTAGE
5. INSTALL NEW BULB

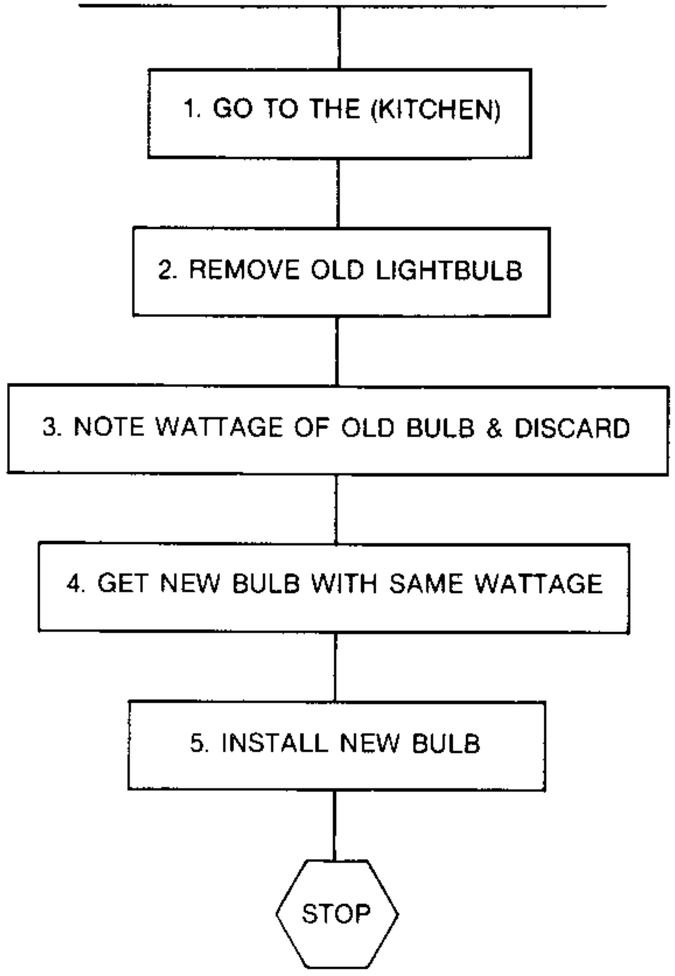
Notice that these subtasks are arranged in a natural *sequence* that cannot be changed. For example, it is simply impossible to remove the old bulb before going to the room where it is to be found. Also, there are not yet sub-tasks for handling problems—e.g., if at Subtask 4 there are no fresh bulbs with the correct wattage. Though at this level they can be ignored, such details must eventually be taken into account.

The procedure is similar to the grammar school exercise of outlining a report. Each step is broken down into sub-steps until they can be translated into instructions the robot can understand. Development of all algorithms follows this procedure of refining the subtasks into progressively smaller steps until each step is simple and unambiguous enough for the robot to execute. The flow chart on the next page is one way to represent an algorithm.

DEVELOPING AN ALGORITHM

(A Flow Chart)

CHANGE LIGHTBULB (KITCHEN)

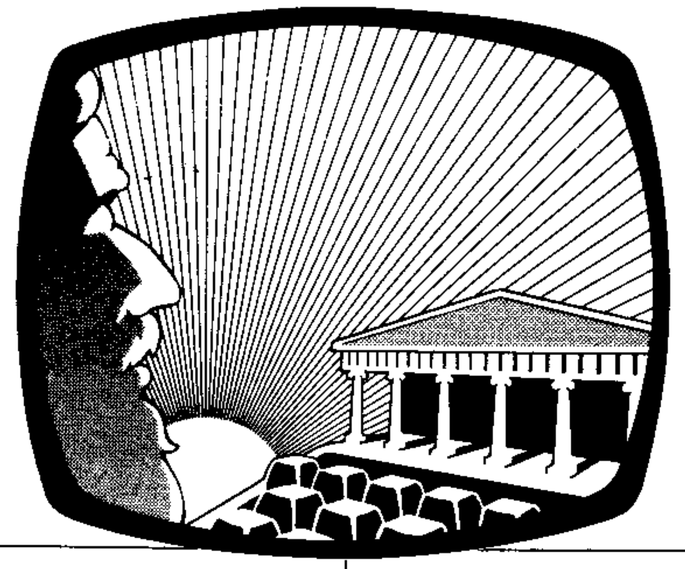


Clear Steps

Once we have defined the problem in mathematical terms, we next have to present it to the computer in clear steps. We have to devise a procedure with a definite beginning, definite steps, and a definite end. Such a procedure is called an ***algorithm****. We use algorithms in various guises in everyday life. Recipes are the most common examples: They tell how much of each ingredient to use and how to prepare them for a given dish. We don't have to use a computer language to define steps clearly, though. Clear plain English sometimes is sufficient; other cases call for mathematical notation. In most cases, however, a combination of both is required.*

Once we have an algorithm which defines all the steps necessary to solve a problem, and indicates all the data (elements and transformations) necessary for that solution, we can translate that algorithm into forms computers can use: Boolean elements and Boolean operators. HCD

PLATO Lives



Even though they've moved out of the home computer market, Texas Instruments has no plans to abandon their PLATO project. In response to the anxious inquiries coming into our offices, we sought out both Dale Osborne, head of TI's Educational Products Division, and Ken Modesitt, Manager of TI's Computer Based Learning Consumer Group. Both gentlemen assured us that TI remains committed to and will bring to market all 108 PLATO program packages as originally scheduled.

The ambitious educational software library for the TI computers grew out of a long collaboration between the University of Illinois and Control Data Corporation. TI bought the field-tested but text-heavy and expensive-to-run Illinois-Control Data educational packages (known as CENTRAL PLATO) in order to format them to run on an inexpensive computer. Although early TI PLATO releases relied heavily on text, their computer format opened up the PLATO library to a host of users who could now bypass the high telephone communication costs that had confined CENTRAL PLATO sales to a well-endowed institutional market. More recent TI PLATO releases redress the text imbalance of the earlier educational packages and capitalize on the full capabilities (including graphics and sound) of the TI-99/4A computer.

School districts and individuals can rest assured that in addition to the *Basic Skills* and *High School Skills* packages already available for the 99/4A, there will soon be a full 108 TI PLATO packages reaching from the third grade through the university level. Of special note in the upcoming TI PLATO packages is an in-depth science series (the Chemistry package alone consists of 7 double-sided floppy disks). At the high school level, packages in social studies are promised. *Basic Reading, Basic Math, Basic Skills Grammar, High School Writing*, poetry, and drama packages are among those scheduled for an early first-quarter release.

TI plans to prepare a not-too-modest 500 sets of each of the first 80 PLATO packages and will make more available when those are sold.

—Erin O'Connor

Les Izmore and DeBug

BY LAREDO & ROBERTS

LES AND DeBUG STILL HAVEN'T FOUND THE MARKETING SCAM THAT WILL MAKE CHAMELEON COMPUTER & PETFOOD Co. #1.

WHAT'S THAT IN MY BREAKFAST?

SOME COMPANIES WILL IMPLEMENT ANYTHING!

BUT EVERY MARKETING PRO WE'VE HIRED HAS BEEN A DUD!

SOME CONSULTANTS WERE STILL PUSHING A GAMES EMPHASIS WHEN THAT MARKET HAD ALREADY LEVELED OFF!

AWESOME!

WE SPENT A FORTUNE GETTING IMPORTANT PEOPLE TO ENDORSE OUR PRODUCTS, BUT FEW CONSUMERS WERE IMPRESSED

WHOOZAT?

WE TRIED TAKING OUR PRODUCT TO FAIRS, BUT NOBODY WANTED TO TALK DATA. THEY ALL JUST WANTED TO RIDE THE OCTOPUS.

PROBLEM IS, LES, NOBODY KNOWS WHO THE AVERAGE CONSUMER IS. LOOK AT ALL THOSE FOLKS OUT THERE. COULD YOU SAY ANY OF THEM ARE AVERAGE?

I HAVE IT! DELETE ALL OLD DATA! FORGET CONSULTANTS AND MARKETEERS. WE'RE GOING TO TRY SOMETHING RADICAL — CUSTOM BUILT MACHINES — ORIGINALS TAILORED TO THE INDIVIDUAL!

WILL THEY SELL?

THEY'VE GOT TO! IT'S OUR LAST HOPE!

LES AND DeBUG SCRAPE TOGETHER EVERY PENNY THEY CAN FOR A MASSIVE ADVERTISING CAMPAIGN!

... AND THEN WAIT...

THEN COMES THAT FATEFUL KNOCK AT THE DOOR!

MY NAME IS HOAGY QUIHOADY. I NEED A COMPUTER!

WHAT'D YOU HAVE IN MIND?

I'M SICK OF SMALL! I'M SICK OF PLASTIC! I'M SICK OF HIGH-TECH. I'M SICK OF INTEGRATED CIRCUITS YOU CAN'T EVEN SEE. I WANT SOMETHING BIG, LUXURIOUS, FLAMBOUYANT WITH CLASS AND EVEN TRADITION! WAIT! WHAT'S THAT MODEL ON THE WALL?

BUT THAT'S ENIAC!!

1949

BUILD IT!

BUT THAT WOULD COST MILLIONS!

I DON'T CARE SCALEFACE! I'M AN ECCENTRIC BILLIONAIRE!

LATER, THINGS ARE REALLY HUMMING!

THE GLASSBLOWERS ARE HERE WITH THE VACUUM TUBES!

A WILD AND COURAGEOUS MARKETING PLOY HAS PAID OFF!

Computers in Education

SCHOOL MARKET WIDE OPEN (BRIEFLY)

Schools used to conservatively buy Apples or frugally go with the TRS-80. Nowadays, when it comes time to buy micros for computer literacy instruction, popular choices include Atari, Commodore, and Franklin home computers. And the entry of the IBM PCjr may further change the face of the market. Despite the drawbacks of the PCjr's infrared keyboard (inappropriate for school settings where the presence of other computers will interfere with its operation), it seems likely that schools will pay the extra $20 for a keyboard cable in a tradeoff for the "serious computing" and "reliable" reputation of IBM. Companies such as The Learning Company (TLC) are going all out to alleviate the lack of PCjr-compatible educational software by September. The cost of the PCjr may be an inhibiting factor, though. School budgets have shrunk in recent years, and schools are under even more pressure than ever to seek out heavy discounting for their large volume purchases.

A COMPUTER IN EVERY CLASSROOM

In a recent message to educators, President Reagan gave hardware manufacturers more incentive to go after the education market. The proverbial promise of a chicken in every pot will be replaced with the equivalent of a computer in every classroom, in keeping with the Chief Executive's recommendation that every student take a half year of computer science.

BIG CATCH IN EDUCATION MARKET

Apple and IBM may be butting heads in a competition to get the most hardware into the schools. While Apple has stopped "giving away" its machines, we expect to see this company cast out some new bait to lure this profitable fish away from Big Blue. Meanwhile, IBM sits smugly on piles of orders for their new PCjr (sight unseen!) from educational institutions. Neither company is talking strategy yet, but buzzings have been heard that one or both may offer sizeable scholarships and grants to schools stocked with their machines. We look forward to counting the orders again in September when the real angler comes forward to claim the prize.

SOUNDS OF MUSIC AT CES

The show was alive with the sounds of music coming from Syntauri Corporation's portion of the Apple Computer Exhibit. From the joystick-controlled *Musicland*, to *Simply Music* (a total keyboard learning and performance system) crowds at the Winter Consumer Electronics Show gathered to watch as Apples were transformed into interactive instruments. Professional-sounding results came from the tuned-in and the tone-deaf alike, and everyone seemed to learn something from these newly-announced packages.

New Tech News

RELIABLE STORAGE FOR THE ADAM

Although Coleco claims that "Adam includes all the hardware and software necessary for immediate use in the home," reports of data losses from their digital tape drive cartridges indicate that long-term reliable data storage remains something of a problem. It seems that inadequate shielding around the tape cartridge and a "residual electromagnetic charge" that occurs when the tape drive is turned on can make the tapes temporarily or permanently unusable. This, coupled with the difficulty of producing prerecorded tapes in volume, spells trouble. Coleco's introduction of an optional 5¼" floppy disk drive at January's CES could signal that the end is near for the digital tape drive. The new disk drive—scheduled for release in second quarter, 1984—will be available for "under $400." Let's hope that this is all the hardware Adam users will need.

AN A.T.&T. PC?

Though it's keeping developments under wraps, A.T.&T. is rumored to be close to releasing their own home computer. Would you believe a 64K unit complete with 1 disk drive and monitor for less than $400? Ma Bell isn't making any public statements, but news of the product is leaking out slowly but surely.

KOALA WIDENS FOCUS

Koala technologies, fresh from their great Koala Pad success, will soon market a device known as the Gibson Light Pen. Designed for use with the Apple II series, IBM PC, IBM PCjr, and Commodore computers, it is expected to retail for under $300. The pen, the interface card, and software allowing the user to create and store high-resolution shapes and designs are included in the package. Properly interfaced, the pen can also be used as a pointing device in lieu of a desk-top mouse. The pen should allow the user to choose and manipulate programs as well as data displayed on the screen in a window environment.

THE KODAK DISK—BUT NOT A CAMERA

Yes, Kodak has gone into the disk drive business, and oh, what a disk drive. The 3.3 Flexible 5¼" Disk Drive—named for its 3.3 MByte capacity (2.62 MByte formated)—boasts a transfer rate of 500 Kbits/sec. (twice the IBM PC's rate) and a track-to-track access time of 3 msec. (half the IBM PC's time).

Kodak also claims the drive is downward-compatible to read conventional 5¼" diskettes because its microprocessor logic detects a conventional disk inserted in the drive and automatically adjusts motor speed to the conventional rate. At CES they had the drive interfaced to a PC via a modified 8" controller using a third-party operating system called DOS-2.08. The drive, produced by Kodak, will be marketed by Data Technology Corporation at a retail price of $495. Release is planned for third quarter, 1984.

Remembering Kodak's marketing ploy—selling cameras to sell their film—it should come as no surprise that Spin Physics of San Diego, is making special pre-formated diskettes for Kodak to market at an anticipated price of $12-$15 each. Also not surprising is that Kodak is the parent company of Spin Physics.

Gameware Updates

WELCOME ABOARD, PAC-MAN?

The reason your flight is off-course may not be hijackers. Recently, intermittent radio interference from a passenger's Pac-Man game shut down an engine on an Eastern Airlines plane. The Radio Technical Commission for Aeronautics (RTCA) will investigate the problem of portable computers interfering with aircraft navigational systems.

EN GARDE!

Atari and Coleco are battling it out with Wico, Discwasher, and Zircon International for the best joystick at the lowest price. Although the Atari 2600 joystick is a traditional best-seller at $9-$11, short supplies have led many gamesters to make other selections from the myriad available in the $10-$50 range. Beefed-up ad campaigns, like Wico's pre-Christmas debut on national television, are expected as the contenders take jabs at Atari's 4-1 sales lead.

TRIAL OFFER

A new marketing approach by Epyx offers a "Preview Disk" that allows consumers to sample excerpts from five games (*Gateway to Apshai, Jumpman Junior, Pitstop, Seawolf,* and *Gunfight*). Epyx has mailed out a quarter-million flyers advertising the sample disks to users who can order one for $2.50, which is refundable with proof of purchase from any of the games.

SEND IN THE CLOWNS

Although games might amount to only half of a software publisher's sales, many feel that popular game titles are important in establishing brand recognition. Advertising for entertainment-oriented software can help lure consumers towards more serious applications packages such as education and home management. The games themselves can be constructed as tutorials that introduce gamesters to other areas of computer use. Watch for a glut of new game titles during the first half of 1984, and new lines of "hybrid" software that mix business with pleasure.

FROM SILVER SCREEN TO MONITOR

The trend in gameware for '84 is movie titles. Box office hits *Krull* and *Star Trek* are slated for release in adventure game versions, and a new game features Kung Fu fighter Bruce Lee. Even Buck Rogers will enter the 21st century immortalized in a video game.

Industry Watch

COMPAQ TRIO LICENSED—OTHERS IN TRIPLE TROUBLE?

Now that they've gone out of the home computer business, Texas Instruments is catching up with encroachments on its always closely-held technology. The three former Texas Instruments employees who founded Compaq some two years ago recently agreed to purchase licenses from TI for personal computer technology used in Compaq's portable microcomputer. Encouraged by this settlement, TI is pushing company patent attorneys to get other personal computer manufacturers to honor three of TI's patents for widely used computer technology—2 bit-pusher patents and a patent to protect TI's combining of manual input device, single-chip personal computer, and screen display. Analysts inside TI insist that many computer manufacturers violate these patents and will be affected by TI's decision to "collect" on their earlier contributions to microprocessor circuitry.

COMPANIES STAY AWAY, OR GO BUT DON'T SHOW

It used to be that a company's appearance at COMDEX or CES signalled the industry that the company was doing all right. Failure to appear was sure to provoke rumors that the company was in trouble. But lately, with the proliferation of computer expositions threatening a schedule of one every six weeks, some companies, (confident that their absence won't be misinterpreted) are being selective about the shows their money and energy go into. Sirius, Sierra On-Line, and Broderbund, for instance, decided this year to relinquish the pleasures of their regular January CES floor exhibit space in Las Vegas in anticipation of a more productive sales effort at CES in June. Others came only to see and be seen.

TI USERS, TAKE HEART

TI will continue to support its huge Home Computer user base. March Direct Marketing (MDM) has been contracted by TI to keep users apprised of third-party hardware and software, via a quarterly catalog. TI will fully honor its warranties and maintain out-of-warranty service in addition to its helpline, 800-TI-CARES. GROM licenses for TI's auto-incrementing memory feature are in the works, as are agreements for selling GROMs to interested third-party software manufacturers. TI is also negotiating with third parties interested in manufacturing 99/4A software formerly produced by TI. So far, both Sierra On-Line and Imagic have agreed to "take back" and distribute the TI software. Sierra On-Line will also manufacture and market 5 educational programs developed by Walt Disney Studios for TI, and Imagic will manufacture and market 5 of its programs for which TI had previously purchased rights.

Toward an Industry Standard

While American computer devotees curse software incompatibilities, the Japanese are doing something about the situation. No fewer than a dozen Japanese firms (including Matsushita, Mitsubishi, Toshiba, Sanyo, and Yamaha) are releasing new MSX-standard personal computers. These machines are hardware compatible, using a Z-80A microprocessor, a TI video chip (the same one as in the 99/4A), and identical sound generators. They are software compatible too, and the MSX standard designation comes from their uniform ability to run Microsoft Super Extended BASIC.

The only American companies to jump on the MSX bandwagon thus far are Spectravideo and ROMOX. Spectravideo will release an MSX-compatible model, the SVI-728, sometime in the first quarter of 1984. Although they have not settled on a price yet, reports place it within the under $500 range. The computer will feature a keyboard with 87 keys (including a numeric key-pad and 10 user-programmable function keys), 32K of ROM (which contains MSX-BASIC), and 80K of RAM. It will have CP/M capability and be expandable to 96K of ROM and 144K of RAM. This should be strong competition in the low-end home computer market.

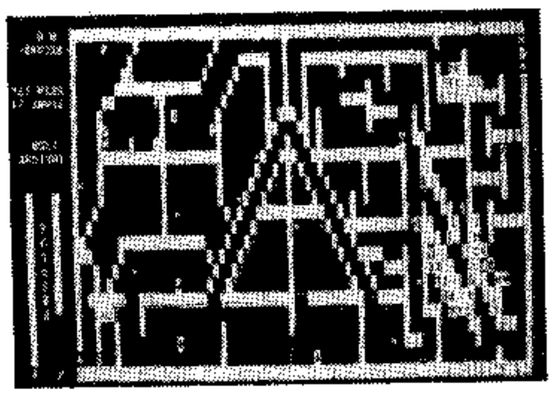
ROMOX feels so far that the MSX is most appropriate for the Japanese market, but hopes to produce software for the MSX system. ROMOX already has arrangements with Aster Co. in Japan for its electronic software distribution system, so this would seem like a natural.

Why such a lack of American interest in the MSX? It's too late: The U.S. home computer market is already bursting with competition. The 8088 microprocessor is used in so many machines now that a Z-80A-based machine might be an out-of-date "standard" in the American market. Right now the PC-compatible computer is the de facto American "standard," although discussions of a U.S. standard these days must include the possibility of an A.T.&T. Unix-based home computer. Mysterious TV commercials featuring the undisclosed contents of a home-computer-sized box identified as "the future" add fuel to expectations that A.T.&T. will exercise its considerable clout to establish a Unix-compatible standard in our American marketplace.

But we Americans don't have the added push for standardization that accompanies language considerations in Japan. High level computer languages (like BASIC) are written in English, so for the Japanese to work in their own language, an extra translation step is necessary. Dealing with a single BASIC (like MSX) instead of several incompatible BASICs greatly reduces these translation difficulties.

— Roger Wood

HOME COMPUTER product news



EVEN THE MOST RELUCTANT CAN BE PERSUADED TO PLAY

Olorin Games has announced the release of *Persuasion*, a fast-action strategy game that will run on all IBM PC computers (64K memory and disk drive required). The game offers an alternative to the themes of killing, destroying, eating dots or chasing aardvarks. Fast-action graphics, multiple difficulty levels and variable speeds make it suitable for players of all skill levels. It can be played with joysticks or keyboard, with monochrome or color display.

Olorin Games
P.O. Box 719
Amherst, MA 01004
(413) 549-4786

INTEGRATED MONEY MANAGEMENT PROGRAMS

Sundex Software Corporation has announced three integrated money management programs for the TI PC, IBM PC, and Apple II and IIe computers, with plans to offer the package for the IBM PCjr and other home computers in early 1984. *Personal Payables*(tm), the bill-paying program retailing at $49.95, handles up to 10 separate checking and saving accounts and can store over 1,000 transactions. Accompanied by Sundex's check-handling holster and a printer, the program can print on the user's own personalized checks or any type of continuous form checks. The program produces financial reports in any combination of check, date, payee, or tax implication. The *Certified Personal Accountant*(tm) program at $99.95 will pay bills and prepare budgets and reports on net worth and cash flow. It will produce income and expense statements and tax information. The *Certified Personal Investor*(tm) at $149.95 is designed for personal portfolio management and analysis and tax form preparation. The program will track up to 150 investments and produce information for the 1040B interest reporting and 1040D capital gains forms.

Sundex Software Corporation
Boulder, CO
(303) 440-3600

VDT EXPANSION BOARD & PSIO CARD THAT REMEMBERS

Videx, Inc.(tm), has released the UltraTerm(tm), an expansion board for video display to augment the power of *VisiCalc*(tm) and word processing for the Apple IIe, Apple II+, Apple III, and Franklin computers. The board features eight software-selectable modes and makes possible screen displays of as many as 160 columns by 24 lines. It is compatible with BASIC, Pascal, and CP/M. The Apple III monitor or the Amdek Video 310A amber monitor, or any CRT with a high- or medium-persistence phosphor is recommended. The UltraTerm package retails for $379.00 and includes board, utilities disk, firmware, and manual. *VisiCalc* applications of UltraTerm require a pre-boot package at $69.00, and *Applewriter II* requires a pre-boot package at $29.00.

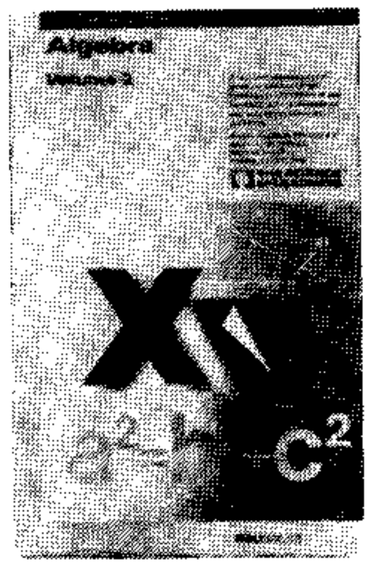
Videx has also announced a PSIO Dual Function Interface Card for the Apple II, Apple IIe, Apple III, and Franklin computers. The PSIO card allows simultaneous use of printer (parallel output) and modem (serial I/O port) with one card. The PSIO card will work with any printer/modem and is compatible with BASIC, Pascal, and CP/M systems. It uses a Non-Volatile Random Access Memory (NOVRAM) to maintain configuration options. The PSIO card retails for $229.00.

Videx, Inc.
897 NW Grant Ave.
Corvallis, OR 97330
(503) 758-0521.

PRINTING SERVICE FOR TI USERS

The Micros' Ink is announcing a printing service for 99/4 and 99/4A users. The service will list programs, print word processing, and plot black and white graphics from cassette or disk. For users without a printer, the service provides hard copies of programs. For users with thermal or other 40-column printers, the service provides 80-column dot matrix copy. Users may specify data processing, emphasized, enhanced, or letter-quality printing. The Micros' Ink has also developed a word processor for TI BASIC users. The user can type text on the TV or monitor screen and correct it. Cassette or disk can then be sent to Micros' Ink for printing. The introductory price for listing programs of less than 16K, printing word processing of less than 10 pages, or plotting one black and white graph is $5.00. Reduced rates are offered for multiple copies, additional pages, or graphics.

The Micros' Ink
P.O. Box 3725
Alliance, OH 44601.



ALGEBRA SELF-TAUGHT WITH NEW SOFTWARE

Eduware has announced *Algebra Volume 2*, the second part of their six-volume series for the IBM PC. The series comprises a first-year course taught in a flexible style which utilizes high-resolution color-coded maps to chart progress. Volumes 3-6 of the IBM version are scheduled for later release, and the entire series is currently available for the Apple II. The 5 1/4" floppy disks require 64K of user memory with IBM DOS 1.1 and 128K with IBM DOS 2.0. The price is $39.95.

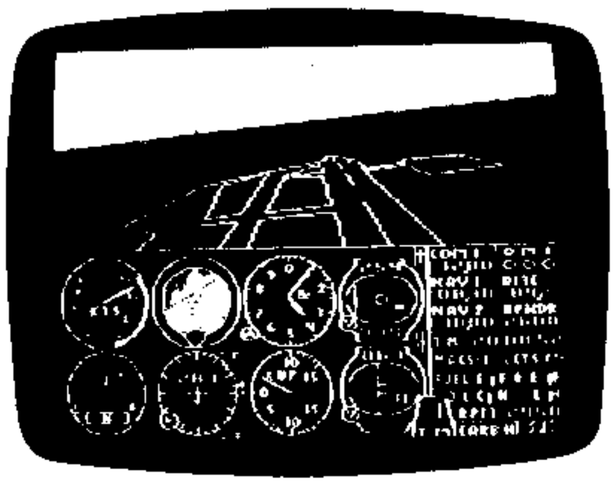
Eduware Services, Inc.
28035 Dorothy Dr.
Agoura Hills, CA 91301

"HOW TO" BOOK FOR HOME COMPUTERS

Texas Instruments, Inc., has recently announced the publication of *How to Feel at Home with a Home Computer* by Gary G. Bitter and Roger S. Walker. The illustrated 264-page book contains step-by-step instructions, and guide maps begin and end each of the nine chapters. In addition to "Getting Acquainted" information, the book contains instruction in creating programs for education, information management, entertainment, speech, music, and graphics. The book retails for $12.95.

Texas Instruments, Inc.
P.O. Box 3640 M/S 54
Dallas, TX 75285

HOME COMPUTER product news



NEW FLIGHT SIMULATOR LANDS ON HOME COMPUTER MARKET

SubLOGIC Corporation has announced *Flight Simulator II* on diskette for the Apple II, Commodore 64, and Atari (48K minimum memory required). The user can practice take-offs and landings at over 80 airports with full flight instrumentation (avionics included) in a Piper 181 Cherokee Archer. The program comes with four scenery areas—New York, Chicago, Seattle, and Los Angeles—and additional scenery areas are available. Weather conditons are user-adjustable, and the program provides for aerobatics and includes a WWI aerial battle game. The price is $49.95.

SubLOGIC Corporation
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482
Telex 206995

SURVEY DESIGN AND ANALYSIS

Telofacts 1 and *2* for the IBM PC has been added to the dilithium Press catalog of computer books and software. The package provides the capability to design and automate questionnaires, tests, polls and other forms. It can also be used to gather opinions, resume information, marketing data, and survey data. The software also provides analysis capability. *Telofacts* runs on the IBM PC or IBM PC XT, with a UCSD p-system, with 128K, two disk drives, and monochrome adapter or color graphics adapter with parallel printer. *Telofacts 1* retails for $49.95. *Telofacts 2*, the enhanced version that can be used with a card reader to rank, list, and score respondents, retails for $199.95. The *Telofacts* packages are also available at the same prices for the Apple II or IIe with 64K and one disk drive. 80-column screen card, second disk drive, and printer are recommended.

How to Use the Peanut for IBM PCjr users has been scheduled for January, 1984, publication.

dilithium Press
8285 SW Nimbus, Suite 151
Beaverton, OR 97005
(800) 547-1842

COMPILER FOR TI-BASIC

SST Software, Inc., has announced the release of its *BASIC Compiler Package* for the TI-99/4A. The compiler translates BASIC programs into TMS9900 machine language for speedier execution. TI console, cassette recorder and cable, and Mini Memory are required. The compiler can write and compile 150 lines of BASIC using the minimal configuration, and 500 lines of BASIC with Memory Expansion with or without disk drive. It can dimension up to 1800-element floating-point arrays with minimal configuration and up to 12,000-element integer arrays using Memory Expansion. It provides for the writing and debugging of BASIC programs using the TI console and interpreter, and then compiles without the need for retyping. *The BASIC Compiler Package* retails for $50.00.

SST Software, Inc.
Box 26
Cedarburg, WI 53012.

PRODUCTIVITY & GAME SOFTWARE FOR TI-99/4A

Konrad Komputerware is announcing a new line of cassette software for the TI-99/4A. Initial offerings include low-cost packages containing games and home finance aids.

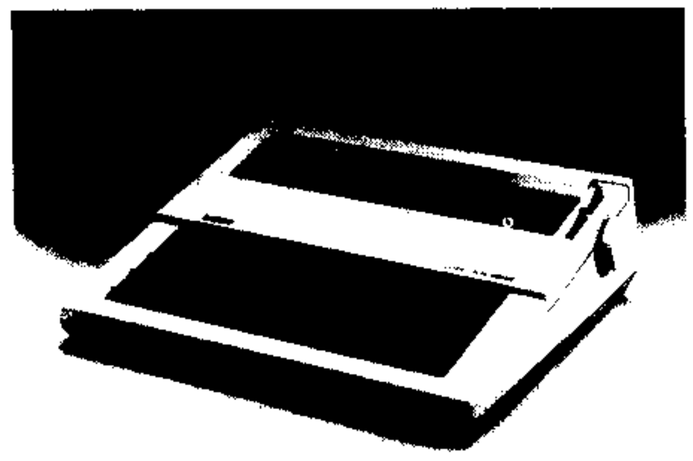
Konrad's *Kasino* is an assortment of gambling games, including Blackjack, Video Poker, Craps, and HiLo, in both BASIC and Extended BASIC. *El Hango*, a Spanish version of Hangman, is available in Extended BASIC for vocabulary practice. Each package retails at $9.95. *IQ Test* for anyone over 12 is available at $7.95. Konrad also offers *Fun with Money* in BASIC and Extended BASIC—a home financial decision-making aid for checkbook, loans, mortgages, investments, periodic savings, and future values. It retails for $9.95.

Konrad Komputerware
P.O.Box 26741
Fort Worth, TX 76126.

GAMES FOR THE ARMCHAIR TRAVELER & HOME SLEUTH

Available for the TI-99/4A and Commodore 64 from Briley Software are five new programs in BASIC on cassette. The Explorer Series includes two new variable text adventures, *High Seas* and *Fur Trapper*. The Detective Series offers three games of deduction and logic with maps and changing solutions: *Mansion*, *Pentagon*, and *Museum*. Each game is priced at $14.95

Briley Software
Box 2913
Livermore, CA 94550.



ELECTRONIC TYPEWRITERS DOUBLE AS PRINTING TERMINALS

Brother International Corporation has introduced the first of their series of electronic compact typewriters that can double as printer terminals by interfacing with a computer using the built-in interface port. The Correctronic 50 offers triple pitch selection, a full one-line correction memory, interchangeable cassette daisy wheel, and cassette ribbon system. The typewriter comes with a built-in carrying case. The Correctronic 50 is priced at $499.95.

Brother International Corporation
8 Corporate Place
Piscataway, NJ 08854.

ELECTRONIC ROAD ATLAS

Columbia Software has introduced *Roadsearch-Plus*, a computerized road atlas available on disk for the Apple II and IIe computers with DOS 3.3. *Roadsearch-Plus* can determine and print the shortest practical route between cities, helps you avoid toll or other specified kinds of roads, and find feature-specific routes. Its database of 406 cities/roads intersections and 70,000 miles of interstate and major through highways can be updated and revised with the user's shortcuts, local roads, and new destinations. Printouts will provide route, distances, travel times, and fuel usage. *Roadsearch-Plus* retails for $74.95.

Columbia Software
5461 Marsh Hawk
Columbia, MD 21045
(301) 997-3100

HOME COMPUTER product news

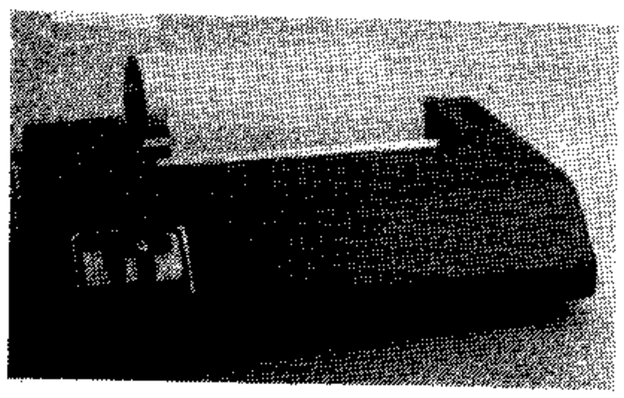
SAKATA MONITORS & STAND

SAKATA U.S.A. Corporation has introduced its SC-100 CRT Composite Color Display Monitor with controls including phone pin jack, sound, contrast, power switch, vertical hold, color, tint, and brightness. The 13" monitor, which weighs 30.8 lbs., can display up to 1,000 characters and retails for $329.00. It is compatible with the Apple II, Atari 800, Commodore 64 and VIC-20, IBM PC, TI-99/4A, and Osborne computers.

For those computers and the Apple III and NEC PC as well, SAKATA has introduced its SG-1000 Monochrome Monitor. The high-resolution CRT is phosphor green with a non-glare high contrast dark face plate. With a 12" screen, the unit weighs 16.5 lbs. and retails for $129.00.

SAKATA has also announced a CRT stand for its color display monitors. It tilts up and down and swivels left and right and to 90°. Made of polystyrene in a neutral color, the monitor stand retails for $49.00.

SAKATA U.S.A. Corporation
651 Bonnie Lane
Elk Grove Village, IL 60007
(800) 323-6647



PRINTER AND INTERFACE BRING DOWN THE COST OF EXPANSION

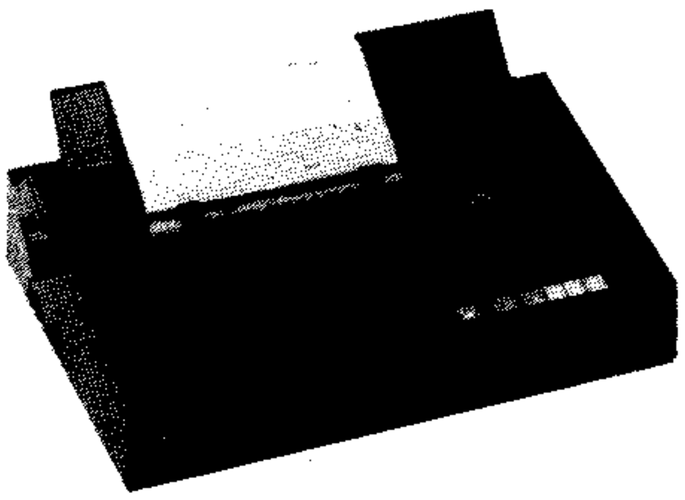
Pueblo Instruments has introduced the PICO-1(tm), a low-cost dot matrix printer compatible with the Apple, Atari, TRS-80, and TI-99/4A. The draft-quality, 7-pin, 80 cps printer is available in beige or grey, and can be ordered with an adapter and cable that allow direct hook-up. A unique interface board, the TIPI(tm) (Texas Instruments Printer Interface), gives TI users printing capabilities without the expense of the TI peripheral expansion system and RS232 card. The PICO-1 printer retails for $330, and the TIPI adapter costs $65.

Pueblo Instruments, Inc.
P.O. Box 3367
Pueblo, CO 81005
(303) 544-7700

"MISSING LINK" INTERFACE

Midwest Engineering Consultants has announced a 200 cps printer adapter called "The Missing Link" that uses the TI joystick port and requires only cassette tape and TI Mini Memory or Extended BASIC cartridge. The adapter package is priced at $36.95. The package includes 35 pages of documentation, fully assembled interface, and nine Mini Memory and Extended BASIC programs, including Mini Memory utilities and Extended BASIC engineering programs: *Ladder Network Analysis*, *Home Security Program*, and *Sprite Program*. Also included are four electrical engineering programs.

Midwest Engineering Consultants
P.O.Box 159
Hawthorn Center
Vernon Hills, IL 60061.



SELF-DIAGNOSTIC PRINTER

Data Terminals and Communications has just announced the DTC Style Writer, a daisy wheel printer with 35K buffer memory that permits simultaneous use of printer and other computer applications. The Style Writer, designed for use with most major personal computers, including IBM, Apple, and TRS-80, features multicopy capability, full bi-directional printing, automatic proportional spacing, standard Centronics parallel interface, graphics plotting, two-color printing, and a momentary pause for paper, print wheel, and ribbon changing. The print wheel is available in 17 different type fonts. A self-test diagnostic routine that operates independent of the computer will evaluate the printer's internal electronic circuits and print mechanism. Error conditions are indicated by LED lights. The DTC Style Writer retails for $899.00.

Data Terminals and Communications
590 Division Street
Campbell CA 95008
(800) 962-8185

NEW SOFTWARE FOR BUDDING ARTISTS

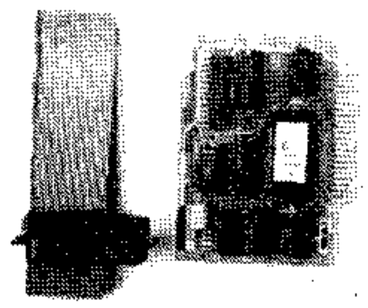
Scarborough Systems has announced a new line of software to encourage young children to experiment with art and music. *PICTUREWRITER* makes it easy to draw and "paint" on the screen using keyboard and joysticks. *SONGWRITER* brings composition skills to the novice and is versatile enough to accommodate the experienced musician. Both of these Art Series programs are available for Apple users for $39.95 each, with versions for the IBM PC, Commodore 64, and Atari in the making.

Scarborough Systems, Inc.
25 North Broadway
Tarrytown, NY 10591
(800) 882-8222

QUICK DISK COPIER

Quality Software has announced the *QUICK-COPYer*(tm), a program for backing up disks on the TI-99/4A. Designed to be faster than the Disk Manager cartridge, the *QUICK-COPYer* will back up a disk in three passes or less and back up a double-sided disk in six passes or less. The *QUICK COPYer* requires the TI-99/4A, disk drive, 32K Memory Expansion, and either the Extended BASIC, Mini Memory, or Editor-Assembler cartridge. The price is $39.95.

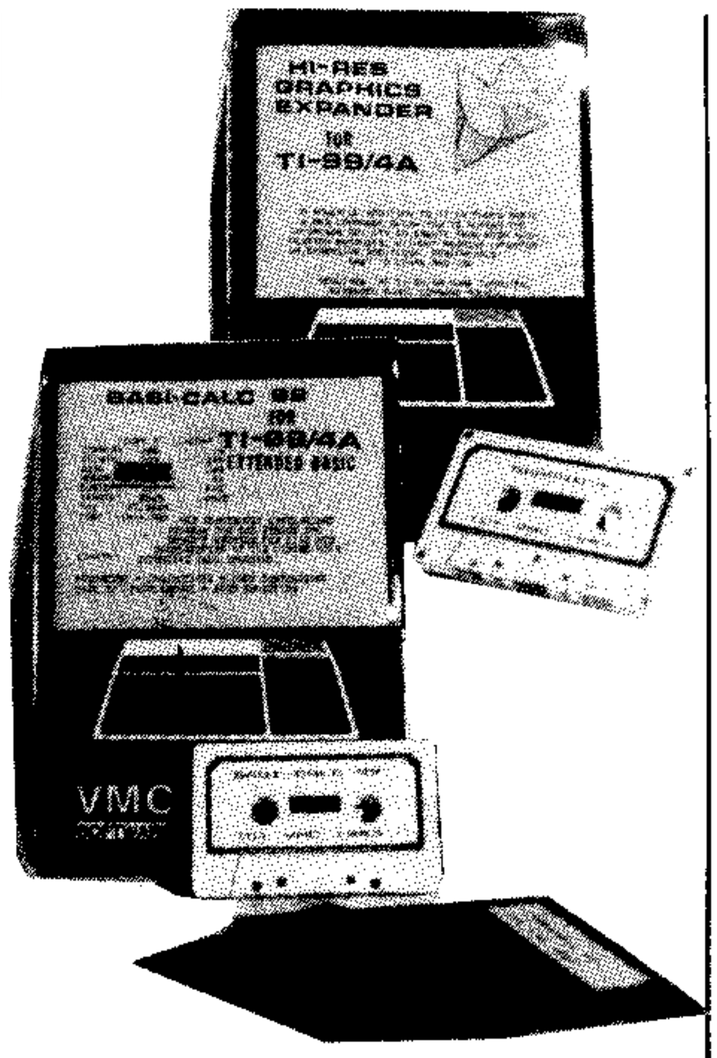
Quality Software
1884 Columbia Rd., #500
Washington, D.C. 20009
(202) 667-3574



WHAT YOU SEE IS WHAT YOU GET WITH PRINT-IT!

Texprint has announced a new printer interface card with 64 kilobytes of internal program ROM, 202 graphics modes, 25 text modes and 7 format modes. PRINT-IT! Model 2 is compatible with all Apple II computers and over 15 models of printers. The self-contained card can handle 40- or 80-column text and high and low resolution graphics in color and black & white, and automatically selects the print format to match the current screen display in the Apple IIe. PRINT-IT! Model 2 retails for $174.00.

Texprint, Inc.
8 Blanchard Rd.
Burlington, MA 01803
(800) 255-1510



HI-RES GRAPHICS & SPREADSHEET FOR EXTENDED BASIC

VMC SOFTWARE has announced a series of new programs for the TI-99/4A. The *Hi Res Graphics Expander* in Extended BASIC includes six new commands and is available on cassette for $15.95. *Basi-Calc 99*, a mini spreadsheet in Extended BASIC requiring no additional memory or peripherals, retails for $16.95 on cassette and $18.95 on disk. Also available are *Mini-Mail 2* and *Checkbook Plus*, each in BASIC on cassette at $13.95, and *Disk Mail 99* in BASIC on disk at $17.95.

VMC SOFTWARE
P.O. Box 326
Cambria Heights, NY 11411.

NEW FANTASY WITH INTERLOGIC

Infocom, Inc., has just announced *ENCHANTER*(tm), a fantasy game co-authored by Mark Blank and Dave Lebling, writers of the *ZORK*(tm) trilogy. The prose adventure game is written for the IBM PC, TI PC, Apple II and Commodore 64 computers. Each package includes diskette, standard documentation, and an eight-page user's manual. Priced at $49.95, the game uses INTERLOGIC(tm), a programming system with a vocabulary of 600 words that enables players to communicate with the program in ordinary English.

Infocom, Inc.
55 Wheeler Street
Cambridge, MA 02138
(617) 492-1031.

AD PILOT PROGRAM & RESEARCH/ ANALYSIS SYSTEM GOES ON-LINE

CompuServe, the commercial consumer videotex/database service, and L.M. Berry and Company have announced a program to offer advertising and direct marketing via the CompuServe Information Service. The four-month pilot program is scheduled to start in January, 1984. A variety of products will be test-advertised for consumer responses, i.e., merchandise orders or requests for product information. Spot advertisements, databases of in-depth product descriptions, and electronic catalogs, as well as order placement and delivery and payment information will be offered.

CompuServe has also announced the availability of MarketScope, a computerized database aid to marketing media research and analysis. MarketScope's information sources include databases from Mediamark Research, Inc., Simmons Market Research Bureau, Broadcast Advertiser Reports, and Arbitron. Demographic data, sales potential data, and regional newspaper studies are also included. Capabilities offered by MarketScope include screening and reporting, consumer profiles, cross tabulation, competitive product analyses, media analyses, reach and frequency studies, sales analyses, statistical modeling, graphics, and mapping.

CompuServe, Inc.
5000 Arlington Centre Blvd.
Columbus, OH 43220
A.R. Char, (614) 457-8600.

DISK/CASSETTE ACCESSORIES

Discwasher has announced the DiscKeeper(tm), a magnetically shielded storage system for floppy disks designed to protect disks against stray magnetic fields from fans, printers, peripheral motors and CRTs, as well as from airport security systems, mail handling equipment, and telephone ringers. The DiscKeeper is available in three sizes: two 5 1/4" floppy disks at $12.95, four 5 1/4" floppy disks at $16.95, and two 8" floppy disks at $14.95.

Discwasher also offers a Disk Drive Cleaner with cleaning programs for both single- and double-sided disk drive heads. The cleaning programs are for the Apple II, IBM, CP/M and VIC-20 computers. Retail prices are $24.95 for 5 1/4" drives and $29.95 for 8" drives.

Discwasher's Computer Cassette Drive CareSet(tm) includes a computer cassette drive head cleaner and a computer cassette drive mechanism cleaner for $14.95.

Discwasher
1407 N. Providence Road
Columbia, MO 65205.

DIGITAL RECORDER FOR COMMODORE 64

Covox Co. has announced the Voice Master(tm), a digital recording device for the Commodore 64. The Voice Master permits the user to enter up to 150 words and phrases and compute responses in BASIC. The Voice Master package includes cassette software—transferable to diskette—with demonstration programs for a talking clock, calculator, and Blackjack, as well as instructions for defining keys for spoken phrases, song notes, or other sounds. The Voice Master with microphone, cassette software, manual, and newsletter updates retails for $119.95, and can be demonstrated over the telephone.

Covox Co.
675-D Conger St.
Eugene, OR 97402
(503) 342-1271
Telex 706017.

Setting Up Cassette Data Files on the 99/4A

There is more to your TI cassette system than just SAVEing and OLDing programs. By learning to use four other TI BASIC commands (OPEN#, INPUT#, PRINT#, and CLOSE#), you will be able to save and recall data with the cassette.

Below is a short program to maintain a list of telephone numbers in an array (a contiguous block of the computer's memory for storage of similar items) whose data may be saved on tape.

There is a very important factor to remember when using a cassette tape for storage—always use a separate tape for your data, not the tape that contains your program!

OPEN # This statement prepares the computer to use a data file on an accessory device. With cassette tapes, you may "OPEN a file" in either input or output mode. See line 260 for an example using OPEN# with a cassette file in the input mode and see line 480 for an example in output mode.

Until you have studied the *TI User's Reference Guide* and mastered cassette data storage, always specify "CS1", INTERNAL, SEQUENTIAL, and FIXED as part of the OPEN# statement. You may specify any number from 1 to 255 for the device (after the # sign), but this number must be the same in all PRINT#, INPUT#, and CLOSE# statements that refer to the device. The number following the FIXED option should be 64, 128, or 192, specifying the number of bytes in each record. Select this number to be larger than the amount of data (characters) you are PRINTing to the cassette.

```
100 REM ********************
110 REM *       PHONE LIST       *
120 REM ********************
130 REM HOME COMPUTER MAGAZINE
140 REM VERSION 4.1.1
150 REM TI BASIC
160 REM CASSETTE DATA
170 REM STORAGE EXAMPLE
180 DIM NAME$(10),PHONE$(10)
190 CALL CLEAR
200 PRINT " P H O N E  L I S T"::::
210 PRINT "1-READ FILE FROM TAPE"::"2-REVIEW AND ENTRY OF DATA"::"3-SAVE FILE TO TAPE"::"4-QUIT":::::
220 INPUT CHOICE
230 IF (CHOICE>4)+(CHOICE<1)=-1 THEN 190
240 ON CHOICE GOTO 250,320,470,540
250 REM READ FILE FROM TAPE
260 OPEN #1:"CS1",INPUT ,INTERNAL,SEQUENTIAL,FIXED 64
270 FOR N=1 TO 10
280 INPUT #1:NAME$(N),PHONE$(N)
290 NEXT N
300 CLOSE #1
310 GOTO 190
320 REM ENTER DATA IN FILE
330 CALL CLEAR
340 PRINT "WHICH RECORD NUMBER"
350 INPUT ENTRY
360 IF (ENTRY>10)+(ENTRY<1)=-1 THEN 190
370 CALL CLEAR
380 PRINT "ENTRY NUMBER ",ENTRY," IS:":NAME$(ENTRY)::"WHOSE PHONE # IS:":PHONE$(ENTRY)::
390 PRINT "1-ENTER NEW NAME":"2-ENTER NEW PHONE NUMBER":"3-TRY ANOTHER ENTRY":"4-EXIT REVIEW AND ENTRY MODE"
400 INPUT CHOICE
410 IF (CHOICE>4)+(CHOICE<1)=-1 THEN 370
420 ON CHOICE GOTO 430,450,330,190
430 INPUT "NAME? ":NAME$(ENTRY)
440 GOTO 370
450 INPUT "PHONE #? ":PHONE$(ENTRY)
460 GOTO 370
470 REM SAVE FILE TO TAPE
480 OPEN #1:"CS1",OUTPUT,INTERNAL,SEQUENTIAL,FIXED 64
490 FOR N=1 TO 10
500 PRINT #1:NAME$(N),PHONE$(N)
510 NEXT N
520 CLOSE #1
530 GOTO 190
540 END
```

CLOSE # This statement causes the computer to finish any pending action with the OPENed device and then "forget" about it. See lines 300 and 520.

PRINT # To save data to the cassette tape, the PRINT# statement is used. The items to be saved are formated into a "variable-list." An example is in line 500.

INPUT # To load data from the cassette tape, the INPUT# statement is used. An example of this is shown in line 280 of the program. Note that the format of the "variable-list" must match that used in the PRINT# statement that recorded the data.

Just entering this short program and seeing how it works with the cassette will help you think of other ways to use this technique for saving other types of data.

—*David Brader*

Have No Fear:

Assembly Language Won't Byte!

Part III

by Peter Lottrup

In this segment of our tutorial on Assembly Language for the Mini Memory cartridge, we'll develop another short demonstration program and execute it. With this program, we'll learn how to enter a name in the Mini Memory's table of references and definitions (REF/DEF Table). An entry in this table allows us to run a program by *name*, rather than executing it using EASY BUG with a memory *address*.

In the following program, we will learn two new words of the Assembly Language vocabulary and how to use them. Later in this series, we'll provide a list of words with explanations and examples of the most important ones. To enter the program, load the *Line-by-Line Assembler*, using EASY BUG. Then select option 2 from the Mini Memory menu and type OLD in response to the prompt.

Shooting Asterisks

The object of the following program is to display an asterisk in the first screen position, erase it, print it again in the next screen positions, and so on. This will demonstrate the speed of Assembly Language. The following listing will display the asterisk:

```
7D00      AQ   LI R0,0
7D04      WD   LI R1,LB
7D08           LI R2,1
7D0C           BLWP @>6028
7D10           LI R1,LS
7D14           BLWP @>6028
7D18           INC R0
7D1A           CI R0,768
7D1E           JNE WD
7D20           JMP AQ
7D22      LB   TEXT '*'
7D24      LS   TEXT ' '
```

This program was written using the VMBW routine (VDP Multiple Byte Write) stored in memory location >6028. Note that it could be done just as well with the VSBW routine (VDP Single Byte Write) stored in Memory location >6024. Both of these routines are covered in the Mini Memory manual, starting on page 35. To give you the best understanding of how to use each routine, the first program listing uses the VMBW routine, and the second uses the VSBW routine.

Load, Skip and Jump

The first line loads the value of zero into register number zero, which is the position on the screen where we want the first asterisk printed. Next we load the label where our text is—in this case, LB—into register number one. In the third line, we load into register number two the length of our text (one byte long). In the fourth line we tell the computer to "GOSUB" (BLWP: Branch and Load Workspace Pointer) to the VMBW routine located in memory location >6028. That prints the first asterisk. In the fifth line, we load into register number one the label—or memory location—where the new text (a space) is and then repeat the printing procedure. In the seventh line, we INCrement the value in register number zero by one. INC always increases a value by one. Then, before returning to the printing routines, we must test to see if the value in register zero is equal to the last screen position (768). If it is, the value in the register must be reset to zero so that the printing will begin again in the first screen position. In the eighth line, we compare the value in register zero with 768, using the CI (Compare Immediate) instruction.

At this point we should clarify the use of the Compare instruction. When comparing a register to a number, we use CI (Compare Immediate), and when comparing two registers, we use C (Compare words). Both of the instructions alter bits in the CPU's Status Register. As your knowledge of Assembly Language increases, you will learn to differentiate among all these instructions and definitions with ease.

In the next line we write JNE (Jump if Not Equal) to the label WD. Otherwise, in the next line, we jump (JMP) to label AQ. The jump instructions test the bits in the Status Register and execute according to the conditions those bits indicate. Finally, we add the two labels with the text—one with the asterisk, the other with the blank.

Now END the program and execute it. (Remember: To execute, return to the master title screen, select EASY BUG, and use the Execute command with a starting address of 7D00.) You will see that the program works so quickly that you cannot see the full asterisk printed out before it is erased. To make the program execute better, we will have to slow it down a bit.

To do this, we will add two delay loops in the right places. Here is the new listing:

```
 1.  7D00      AQ   LI R0,0
 2.  7D04      WD   LI R1,>2A00
 3.  7D08           BLWP @>6024
 4.  7D0C           LI R2,300
 5.  7D10           LI R3,1
 6.  7D14      LO   INC R3
 7.  7D16           C R2,R3
 8.  7D18           JNE LO
 9.  7D1A           LI R1,>2000
10.  7D1E           BLWP @>6024
11.  7D22           LI R3,1
12.  7D26      KL   INC R3
13.  7D28           C R2,R3
14.  7D2A           JNE KL
15.  7D2C           INC R0
16.  7D2E           CI R0,768
17.  7D32           JNE WD
18.  7D34           JMP AQ
```

This listing is the same as the first one, except that it has been written using the VSBW (VDP Single Byte Write) routine and has a couple of delay loops added to it. Please note that the line numbers (left column) have no relation whatsoever to the program listing, but were included to make it easier to understand the explanation of the program. Note also that the hexadecimal translations of the contents of each memory location (second column above) have not been included because they are unnecessary at this point.

Explanation of the Program

Line 1, with the label AQ, loads into register number zero the value of the first screen position (0). Line 2, with the label WD, loads into register number one the hexadecimal ASCII code of the asterisk (>2A). This hexadecimal value is entered, followed by two trailing zeros (>2A00). Line 3 writes the asterisk into VDP RAM. (This time we've switched to the VSBW routine in memory location >6024.) Lines 4 through 8 are the delay loop. First we load the value 300 (decimal) into register two and the value 1 into register number three. Then all we do is INCrement register three (line 6) and Compare it to register number two (line 7). Line 8 means Jump if Not Equal to label LO. The loop continues until both registers are equal (300 times).

At this point then, it is clear that lines 4 through 8 are similar to the following Extended BASIC statement:

```
FOR G = 1 TO 300 :: NEXT G
```

Just remember, though, that 300 times in Assembly Language is much faster than 300 in BASIC. We will experiment with the speed again further on.

Line 9 changes the hexadecimal ASCII value in register one to the value corresponding to a space (decimal value 32, hexadecimal 20) instead of an asterisk. Line 10 writes it onto the screen.

Lines 11 through 14 are the new delay loop. Note that the value in register number two is not included because it has not been changed from the first delay loop. Finally, the last four lines are the same as in the first program. Now END the program and execute it. Does the asterisk move better now? If you want to make it go slower or faster, just invoke the *Line-by-Line Assembler* and type:

```
7D00    AORG >7D0C
7D0C    LI   R2, number
```

Then press [ENTER], type END, and execute the program again.

"You will see that the program works so quickly that you cannot see the full asterisk printed out before it is erased."

Checking for Space

If you want to save the program by name, you will first have to find out whether there is enough space to add the name and starting location of the program. The table where this information is stored is called the REF/DEF table. You need eight bytes in this table to store the name. Your next step—checking the remaining space—is really not necessary if your programs are short, but you should still know how to do it.

First, with the AORG Directive, you will check the value in address >701C (the location of the First Free Address of the Module) and the value in >701E (the Last Free Address of the Module). Then find the difference between the values to see if you have eight bytes or more. So right after you have copied the last line of the example program, type:

```
7D30 83A0 AORG >701C
701C 7FB2 AORG >701E
701E 7FE8 ■
```

(■ shows the position of the cursor.)

7FB2 is the value in 701C, and 7FE8 is the value in 701E. Subtracting the first from the second (7FB2 from 7FE8), we get hexadecimal 36 (or 54 decimal), enough to store the information about our program.

Now that we know there is enough space remaining, we will tell the computer which is the new First Free Address of the Module (the one right after our program finishes) by placing that value into >701C, thus updating the First Free Address. So type:

```
701E 7604 AORG >701C
701C 7D30 DATA >7D36
701E 7FE8 ■
```

Next, we must change the value in the Last Free Address of the Module (>701E) to a number eight bytes smaller so the name and starting point of our program can be added:

```
701E 7FE0 DATA >7FE0
7020 71A6 ■
```

The number we give the computer (7FE8 − 8 = 7FE0) is the place where the information about our program will be found.

Now we can give a name to the program. This name can be from one to six

Continued on p. 137

Meltdown . . . from p. 75

Meltdown
Explanation of the Program

Line Nos.	
100-170	Program Header.
180-210	Title screen.
220-330	Define characters.
340-380	Initialize variables and fill arrays.
390-440	Choose joystick or keyboard.
450-540	Draw playing area and hidden gamma particles and rays.
550-580	Change letters and numbers to write. Display scorekeeping.
590-630	Display level number.
640-660	Draw hero sprite. Check to see if enough particles have been created to move to higher level.
670-710	If no oxygen tank already present, randomly create and place one.
720-890	Randomly turn on hidden gamma rays and check to see if hero gets hit.
900-950	If level is greater than 1, then randomly place anti-matter particles (which appear more frequently at higher levels).
960-990	Choose particle or roach. (Roaches first appear at level 3 and appear more often at higher levels.)
1000-1050	Randomly choose position and color of particle or roach. Play appropriate sound. Move hero.
1060-1090	Shoot neutralizer. Check for various coincidences.
1100-1130	Move hero. Check for coincidences. Shoot.
1140-1180	Delete bullet or particle if off screen.
1190-1240	Update score.
1250-1270	Destroy particle if hit. Change roach to "super roach" if hit only once.
1280-1330	Destroy particle or roach. Update score.
1340-1390	Destroy hero. Update score.
1400-1420	Obtain oxygen. Update score.
1430-1750	Choose appropriate end of game.
1760-1830	Hero move subroutine.
1840-1940	Shoot subroutine.
1950-2010	Level color change.
2020-2110	Data.

TI-99/4A

```
100 REM ************
110 REM *  MELTDOWN  *
120 REM *          *
130 REM ************
140 REM BY STEVE LANGGUTH
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.1.1
170 REM TI EXTENDED BASIC
180 REM
190 REM**TITLE SCREEN**
200 REM
210 CALL CLEAR :: DISPLAY AT(11,11):"ME
LTDOWN"
220 REM *DEFINE CHARACTERS*
230 CALL CHAR(43,"0000FF8181FF0000",37,
"007E7E42427E7E00")
240 CALL CHAR(96,"03060603013F343534370
3030303030FC06060C080FC2CAC2CE06060
60780000")
250 CALL CHAR(100,"03060603013F34353407
06060 61E0000C06060C080FC2CAC2CECC0C
0C0C0C0F0")
260 CALL CHAR(104,"00000000000038EE0300
0000000000000000000000038EE830000
0000000000")
270 CALL CHAR(108,"0103060C193366CCCC66
33190C06030180C0603098CC66333366CC9
83060C080")
280 CALL CHAR(132,"66",116,"66",124,"66
")
290 CALL CHAR(112,"001F2040404E5B51515B
4E4040201F0000F80402020202324A12227
A0204F800")
300 CALL CHAR(120,"0000103B12021E10101E
02123B100000000008DC484078080878404
8DC080000")
310 CALL CHAR(140,"C1E171381C0E07E3E307
0E1C3871E1C183878E1C3870E0C7C7E0703
81C8E8783")
320 CALL CHAR(128,"012110080204011BC309
04112140000000000284882090C3D880204
010088480")
330 CALL CHAR(136,"00001212123F7FD5D57F
3F121212000000000484848FCFEABABFEFC4
848480000")
340 REM **INITIALIZATION**
```

.....Texas Instruments TI-99/4A — COMPUTERS, COMPONENTS AND SOFTWARE.....

TEX-COMP™

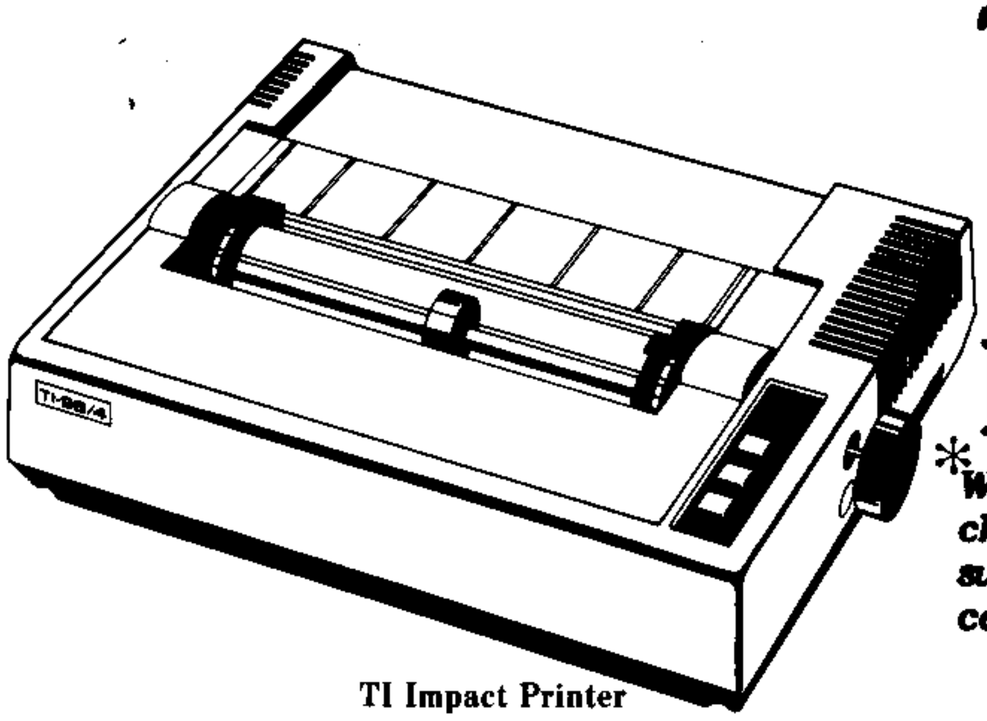
TI USERS SUPPLY COMPANY "YOUR LUBBOCK CONNECTION™"

America's Number One TI computer retailer

TI-99/4 PRODUCTS AT PROFESSIONAL PRICES

ATTENTION TI-99/4 USERS TEX COMP HAS BEEN SUPPORTING THE ENTIRE 99/4 PRODUCT LINE TO PROFESSIONAL AND INDUSTRIAL ACCOUNTS. SINCE THE INTRODUCTION OF THE TI-99/4 NOW YOU CAN PURCHASE ALL YOUR TI-99/4 REQUIREMENTS FROM ONE SOURCE AT LOW "PROFESSIONALLY DISCOUNTED" PRICES ON THE **ENTIRE** 99/4 PRODUCT LINE. WE ARE A DIRECT TI NATIONAL ACCOUNT AND DEAL DIRECTLY WITH TI ASSURING YOU THE BEST SERVICE, EARLIEST SHIPMENT AND LOWEST POSSIBLE PRICES

AUTHORIZED DEALER



TI Impact Printer

TI IMPACT $369.95 PRINTER

IN STOCK

IMMEDIATE SHIPMENT

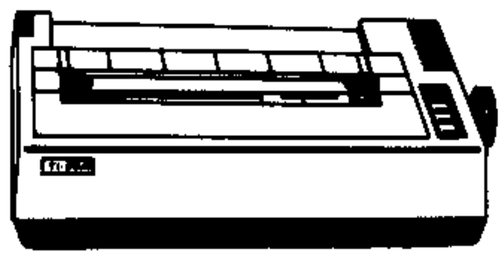
With each printer you may order one Fingerprint upgrade chip for the printer which lets you select different features such as type size and density without entering a software command . Just press the on button to enter each feature.

Reg Retail..$59.95....with printer....$$36.95

This quality printer carrying the Texas Instruments name and backed by the TI warranty and exchange centers, is made for TI by Epson, the number one name in home and personal computer printers. This model, the MX80 is also used by IBM and HP as their "official " printer. Unlike a standard Epson MX80, this TI model comes with many extra features including a cable ($25.00value), a serial and 2K buffer board($150.00value) and a users manual written just for the TI-99/4A.

It prints at a fast 80-character-per-second speed...bidirectionally and you can select from three type sizes and four printing densities. It features a long life 9 x 9 dot matrix user replaceable print head and can print both text and graphics.

There may never be a better time than now to buy!



Special: With each printer you may order extra ribbon cartridges at $9.95 (reg.$12.95)

This printer originally sold for $750.00 at retail.

TEX-COMP PRICE.....$369.95

With each printer you may order one RS232 Card (required for any printer at $69.95.

TEX-COMP USERS SUPPLY
P.O.BOX 33084, GRANADA HILLS, CA 91344

Send Check or Money Order to:

TEX-COMP™
TI USERS SUPPLY COMPANY
"YOUR LUBBOCK CONNECTION™"
TI-99/4 PRODUCTS AT PROFESSIONAL PRICES
WHEN IN SOUTHERN CALIFORNIA VISIT OUR MODERN WAREHOUSE OUTLET STORE WHERE YOU CAN PURCHASE THE ENTIRE TI LINE AT OUR REGULAR DISCOUNT PRICES.

TEX-COMP • P.O.BOX 33084 • GRANADA HILLS, CA. 91344 • (213) 366-6631

Add 3% for Visa or MC

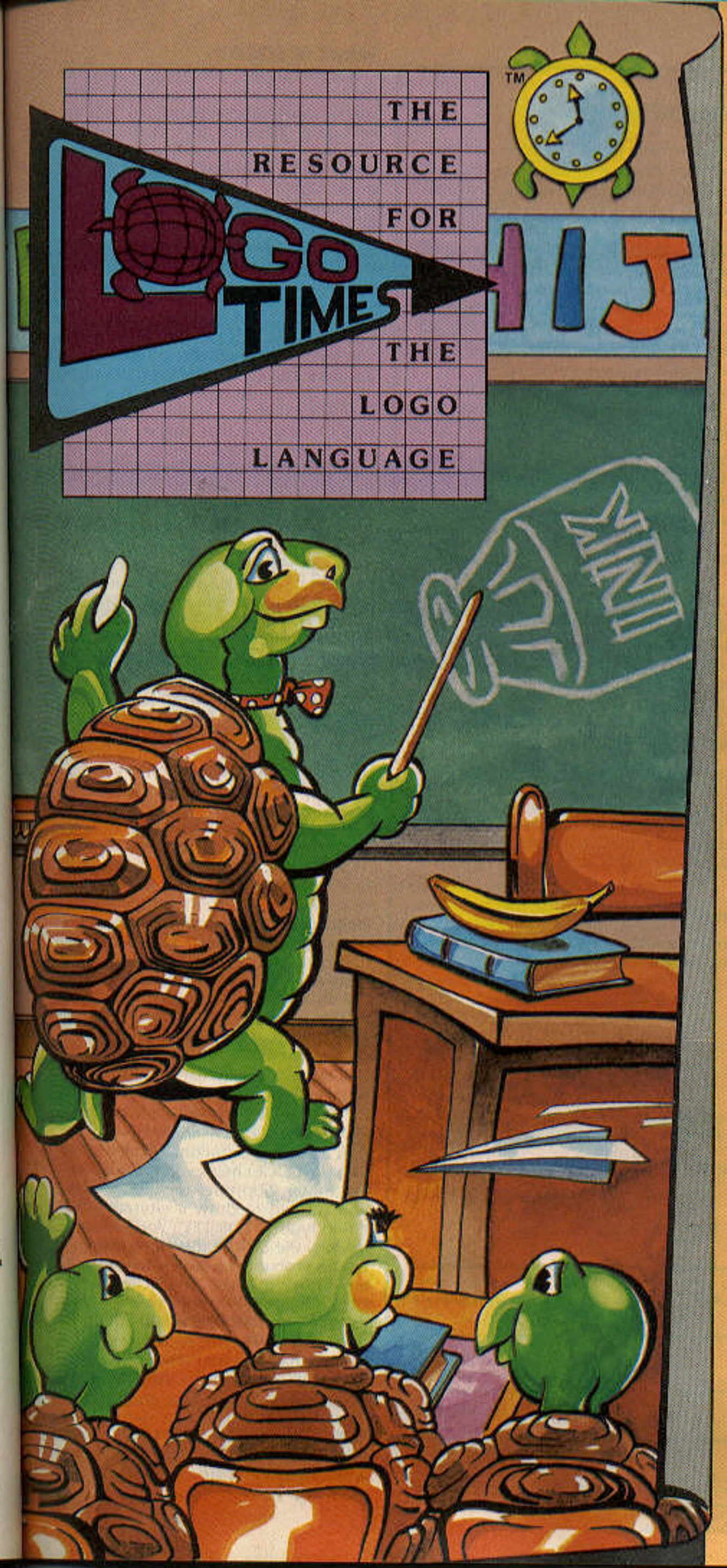
TEXAS INSTRUMENTS Authorized Dealer

Visa and MasterCard Holders Order Direct [818] 366-6631

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). East of Mississippi 4½%. (Free shipping on all software orders over $100.00) We will ship UPS C.O.D. with 25% deposit C.O.D. to be paid by cash or certified check. All products are sold with the original manufacturer's guarantee (sent on request). Prices and availability subject to change without notice. We reserve the right to limit quantities.

SEND FOR THE LATEST TEX-COMP CATALOG OF TI -99/4A PRODUCTS AND ACCESSORIES. ONLY $2.00 (cash or check only)

We include a $5.00 merchandise certificate



WHAT IS LOGO

by Robert Ackerman
Managing Editor

As much as anything else, LOGO is a way of thinking. It is, of course, primarily known as a programming language and a part of a computer system. But it came into being specifically from attempts to aid beginners in computing to think about problems. As a result, it developed into a language that helps you attack problems in a structured way.

In its beginnings, LOGO used a robot that rolled across the floor on command from a computer console. On command, it changed direction, backed up, turned. On command, it also lowered a pen to a sheet of paper on the floor, and the pen traced out a pattern as the robot moved. Because of its domed shape, this robot became known as the *turtle*. Even after its evolution from a robot on the floor to a graphics character on a video display screen, the name stuck.

The form stuck as well. *Turtle graphics*, as it's sometimes called, gives an immediate and visual response to commands typed at the keyboard. Beginners can see instantaneously the result of a command. A beginner can build on that command, follow it with others and create shapes on the screen. After some experimentation, you might find, for example, that the following sequence of commands builds a square:

```
TELL TURTLE
PENDOWN
RIGHT 180
FORWARD 20
RIGHT 90
FORWARD 20
RIGHT 90
FORWARD 20
RIGHT 90
FORWARD 20
```

In some versions of LOGO, you may have to type TELL 0 or SHOWTURTLE to get the turtle on the screen initially.

If you now use the following commands to make two legs of a triangle, you have a house—a bit simple, but a house:

```
PENDOWN
LEFT 120
FORWARD 20
LEFT 120
FORWARD 20
```

Introduction

LOGO Times is an information resource for users who want to create their own *personal* languages—languages that will easily allow them to communicate with the computer in a totally new audiovisual realm of applied imagination, exploration, and self-discovery. The articles on these pages concern the use of the LOGO language, but readers do *not* need any additional software or equipment (or even a computer) to understand and learn from the material presented here.

If readers want to actually *experience* a LOGO environment, they will need a computer, the requisite software and/or cartridges, and any additional hardware required for a particular implementation. A disk drive is required for some LOGO implementations, but in other cases, a user's work may be saved on cassette tape, or copied into a notebook (for later re-keyboarding).

The varieties of LOGO we'll consider include—but are not limited to—Terrapin LOGO for the Apple II, II+ or IIe and the Commodore 64, TI LOGO for the TI-99/4A, and LOGO Computer Systems LOGO for the IBM PC and PCjr.

- **Apple:** Terrapin LOGO requires an Apple II, II+ or IIe with 64K of RAM, one disk drive with controller, and a blank, initialized disk.
- **Commodore 64:** Terrapin LOGO requires a Commodore 64 with a VIC-1541 Disk Drive and a blank, initialized disk.
- **TI-99/4A:** TI LOGO requires the TI LOGO or TI LOGO II cartridge and a compatible 32K memory expansion unit. A cassette recorder may be used for storage, but a compatible disk system is recommended for convenience.
- **IBM PC or PCjr:** LOGO Computer Systems LOGO requires the PC or PCjr with 128 bytes of RAM, one disk drive, and a blank, initialized disk.

In each issue, one or more of the articles may refer to or build upon the topics discussed in a previous article. It is therefore recommended that for maximum benefit and understanding, new readers obtain the appropriate back issues of *Home Computer Magazine* containing *LOGO Times* articles.

LOGO Listings

As you enter LOGO statements, the last thing you do at the end of every statement is to press [ENTER] on the TI and IBM (the key with the ↵ symbol), or [RETURN] on the Commodore 64 and Apple. This signals the system to begin a new line. In our typeset listings, single LOGO statements may carry over from one line to the next without ending. The end of a LOGO statement is marked with a curved arrow (↵) to indicate that you press [ENTER] or [RETURN] at that point.

Notice

LOGO Times is actively soliciting articles. Manuscripts should be typed double-spaced, and accompanied by a cassette tape or disk if containing any lengthy procedures or graphics.

Send all materials to:

LOGO Times Editorial Dept.
Home Computer Magazine
1500 Valley River Dr., Suite 250
Eugene, OR 97401

All mail directed to the Letters-to-the-Editor column (Letters on LOGO) will be published in accordance with the conditions set forth on *Home Computer Magazine*'s masthead page.

Logo Contents — Volume 4, Number 1

What is LOGO?

LOGO allows this exploration of shapes from the bottom up, starting from single commands typed in at the keyboard. Because of this immediate response to commands at the lowest level, LOGO enthusiasts say it has *no threshold*.

In addition to giving you an immediate response, LOGO also allows you to gather your discoveries together and name them. This lets you think of them as units. For instance, if you type:

TO SQUARE

you'll enter LOGO's edit mode. When you follow that with the commands for a square and then close off that unit by typing END, you have created a structure that will draw a square. Now, to get LOGO to draw that shape, you only have to return to the main LOGO screen and type SQUARE. This structure, beginning with TO and finishing with END, is called a *procedure*. Procedures are the basic programming structure in LOGO. If you now type TO TRIANGLE, follow that with the above commands for the two legs of a triangle and END, you have another procedure. Type SQUARE and then TRIANGLE. The turtle will draw the simple house on the screen.

But instead of merely typing these procedure names one after the other, you can organize them into yet another procedure. If you type TO HOUSE, followed by SQUARE, TRIANGLE and END, you have a procedure (HOUSE) that uses, or *calls*, other procedures. After you type HOUSE, LOGO first draws the square according to the commands in SQUARE, and then it makes the triangle from the commands in TRIANGLE.

Other languages are oriented towards program lines and line numbers; BASIC is a good example. Programs in those languages transfer from one section of the program to another according to line numbers, and program segments run from line number *xxx* to line number *yyy*. That tends to obscure the structure of the program and makes programs harder to read and understand. LOGO does away with BASIC statements such as a GOSUB 2000 (what's 2000?) and instead calls procedures by their names—NEWHOUSE or SKYSCRAPER or MOONRISING or whatever.

In addition to building from the bottom up, LOGO helps you build from the top down. Suppose, for instance, that you wanted to draw a cityscape. You could type in TO CITY. You know that to draw the city on the video display, it would probably be easiest to start at the edge. So you type in EDGE after TO CITY. Your city will have houses, so you can type in HOUSES. When you follow that with END, you've defined a procedure that calls two other procedures. But wait aren't those two other procedures undefined? Yes, and if you try to run CITY before you define them, LOGO won't know what to do. But you now know what you want to do and what the general structure of your program will be.

Continued on p. 128

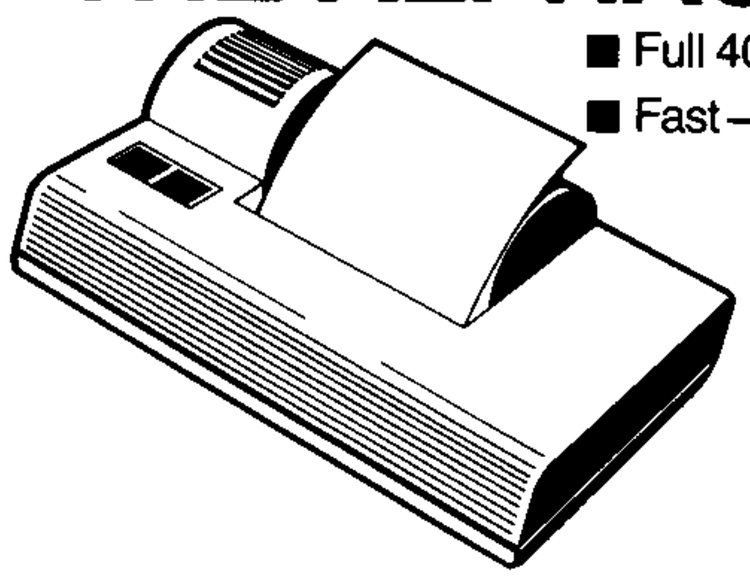
TI 99/4A PRINTER CONNECTION

THE LOW-COST ALTERNATIVE TO THE TI PERIPHERAL EXPANSION BOX

With the Alphacom 42 or 81 printers, it's a snap for your customers to expand their TI 99/4A into a full-function computer system. And all for less than the cost of TI's peripheral box!

All they need is this specially designed intelligent interface cable that connects the Alphacom printer to the TI 99/4A computer in one simple step.

THE ALPHACOM 42

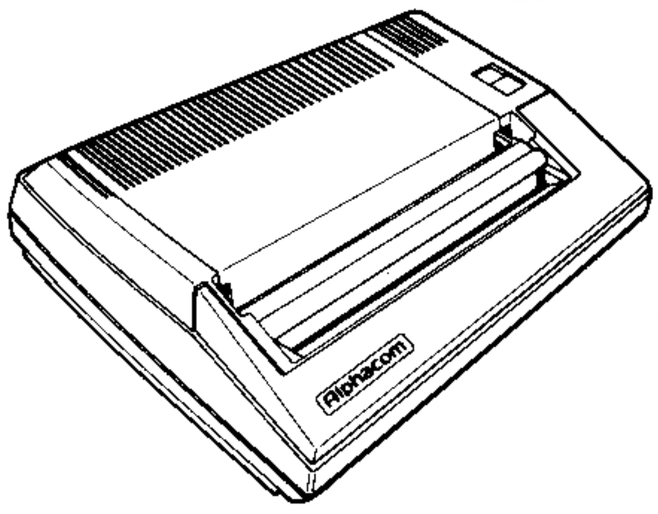


- Full 40-column width
- Fast — 2 lines per second

JUST $99.95

THE ALPHACOM 81

- Full 80-column width
- Fast—100 characters per second

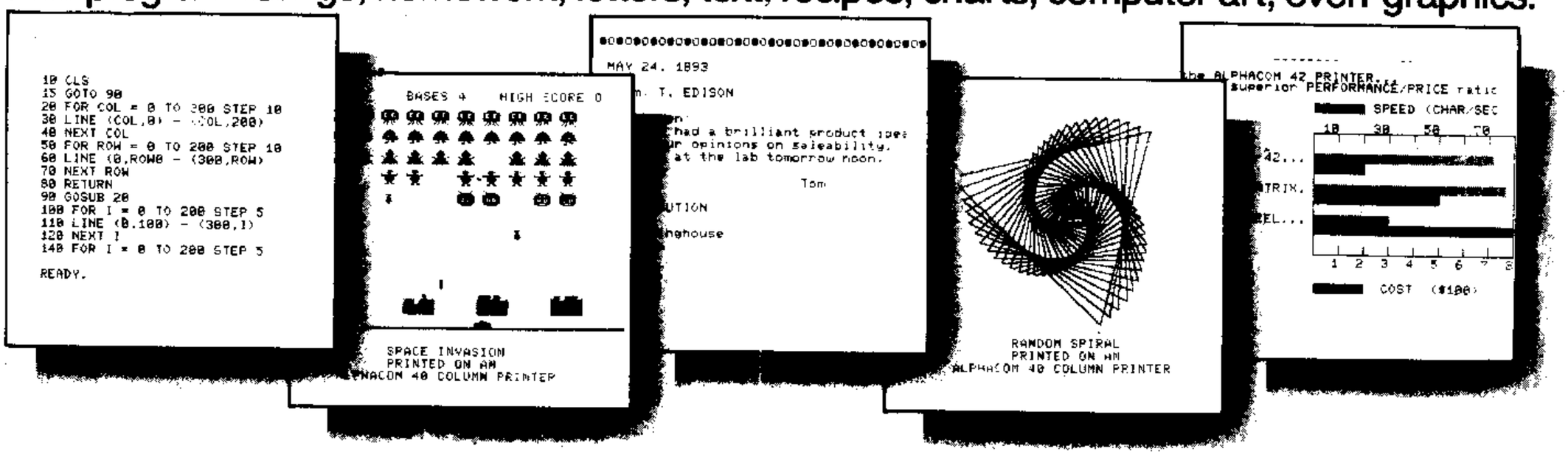


JUST $169.95

THE IDEAL PRINTERS

- Advanced thermal technology for quiet operation.
- Reliable, no maintenance required.
- 90-day limited warranty.
- Complete with paper, power adapter and easy-to-follow instructions.
- The cable retail price is right: $**39.95** for the Alphacom 42 printer interface cable; $**59.95** for the Alphacom 81 printer interface cable.
- Interface cables also available for Commodore VIC 20, Commodore 64, Atari 400/800/1200XL.

Print program listings, homework, letters, text, recipes, charts, computer art, even graphics.



The BACH Company

POST OFFICE BOX 51178 • PALO ALTO, CALIFORNIA 94303

SHOWROOM: 2500 OLD MIDDLEFIELD ROAD • MOUNTAIN VIEW, CALIFORNIA 94043

CALL TOLL FREE
800-227-8292

California
800-248-BACH (248-2224)
except 415 area code use 494-1995

What is LOGO? . . . from p. 122

You might want to tackle EDGE first. One solution is:

```
TO EDGE
PENUP
CLEARSCREEN
TELL TURTLE
RIGHT 90
FORWARD 120
RIGHT 90
END
```

Then to get your houses, you could define HOUSES as follows:

```
TO HOUSES
HOUSE
HOUSE
HOUSE
HOUSE
HOUSE
END
```

Let's modify the original SQUARE procedure as well, by removing the third line, RIGHT 180. (The two commands of RIGHT 90 in EDGE accomplish the same thing.) That will give you five houses—enough for a small city. Or will it? When you type HOUSES, you get part of a complex polygon: a semicircle of five interconnected houses. Between houses, you want to move the turtle to the location for the next house. You can do this with a procedure called, say, MOVE. Now HOUSES reads:

```
TO HOUSES
HOUSE
MOVE
HOUSE
MOVE
HOUSE
MOVE
HOUSE
END
```

And what should MOVE look like? When the turtle moves, you don't want it to draw—the houses should be separate. So you have it pull its pen up. Next you want it to turn right 60 degrees.

Now comes the actual move: FORWARD 5. To get the house right side up, you have to turn the turtle in the right direction: LEFT 90. MOVE has the following structure:

```
TO MOVE
PENUP
RIGHT 60
FORWARD 5
LEFT 90
END
```

The turtle is finally ready to draw the next house on the block.

This process of breaking the larger problem of CITY into EDGE, HOUSES and MOVE allows you to work on smaller, clearly-identified portions of the big problem. One of the developers of LOGO, Seymour Papert, calls these small portions "mind-sized bites." Each is a structure with a name; complete in itself, yet small enough that the whole of it can be grasped readily.

If you look back on the procedure CITY, one thing stands out: The procedure HOUSES is awkward, with its long string of repetitive commands. But what's worse, it gives you only five houses. What if you want four? Or six? Or two? Do you have to change it every time? Or is there an easier way?

There is an easier way. LOGO procedures allow you to give the information they need within the procedures themselves. If, for example, you wanted a city with six houses, you could alter the procedure to start its definition with this line:

```
TO CITY :N
```

N is a variable name; the colon indicates to CITY that it is to use the value stored in the variable. The name of the procedure now includes the information that the procedure needs more data before it can begin work. If you type:

```
CITY 6
```

LOGO recognizes this as the procedure with its value. In this case, you don't have to give N a value and then call the procedure using N. The procedure expects its name to be followed by a value, and it recognizes 6 as a value it can use. These bits of information that procedures need in order to work are called their *inputs*.

The next step should have the procedure draw the six houses, but to do this, you'll have to redefine HOUSES. Each time it runs, it should draw a house, move, draw a house, move, and so on, until it has put six of them on the

screen. You can get the procedure to do this by redefining it as:

```
TO HOUSES :N
IF ( :N = 0 ) THEN STOP ELSE
   HOUSES ( :N - 1 )
HOUSE
MOVE
END
```

The second line of the procedure first checks to see if the value of N is zero; if it is, then it stops. IBM LOGO does not recognize THEN and ELSE. Instead you simply place brackets around the instructions like:

```
IF ( :N = 0 ) [STOP] [HOUSES ( :N - 1 )]
```

If N isn't zero, then HOUSES does something interesting: It calls itself. This process, called *recursion*, is one of the most powerful features of LOGO. In essence, LOGO makes a copy of the procedure it is presently in, and then runs that copy. It continues until it is told to leave the copy—in this case, by STOP or END. Then it runs the previous copy, until it is told to leave. It then runs the previous copy, until. . . , and so on. In this case, HOUSES calls itself with a value one less than the original value of N. It will continue to subtract 1 from N and call itself until N equals zero. Then it will begin to draw houses and move. You'll have to alter CITY to read:

```
TO CITY :N
EDGE
HOUSES :N
END
```

and then run it with a number—a 6, for instance.

By adding some PRINT statements to the procedure, you can see part of the recursion take place:

```
TO HOUSES :N
PRINT :N
IF ( :N = 0 ) THEN STOP ELSE
   HOUSES ( :N - 1 )
HOUSE
MOVE
PRINT :N
END
```

The first PRINT statement prints how many more times HOUSES is going to call itself. The second prints the number of houses the procedure has finished drawing.

If you want to see it run slowly, you can insert a delay after the first PRINT statement by having the program repeat an action—say raising and lowering its pen—50 times.

```
REPEAT 50 [PENUP PENDOWN]
```

Recursion finds many applications in many areas. Factorials, for example, or the Fibonacci series—two common mathematical functions—are readily calculated with recursive procedures.

If LOGO has a serious weakness, it is that individual statements can quickly become difficult to read if they become complex. At one point, the Texas Instruments LOGO manual notes that "parentheses help considerably in enabling the human eye to see the pattern [of a statement, and] unless you are very practiced, you should not write a complex expression without parentheses for fear of not being able to read it the next day." It's always helpful to group portions of a statement in parentheses, as in the example above.

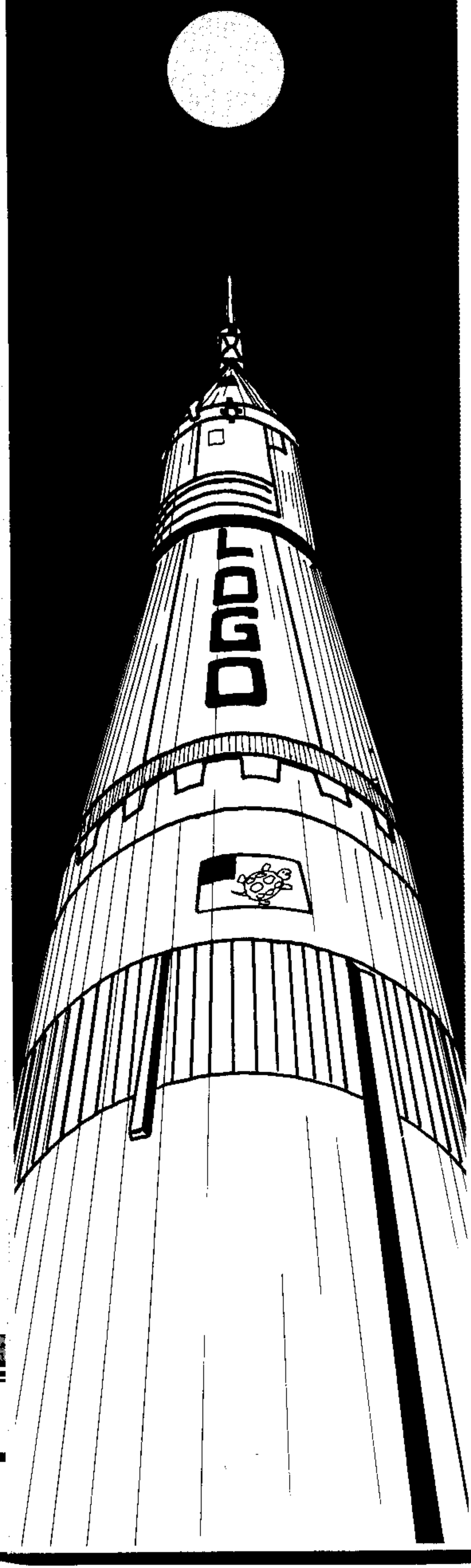
Beyond its graphics features, LOGO offers some other very powerful features. LOGO is a *list processing* language, with special commands for dealing with structures known as *lists*. LOGO uses brackets to mark off lists, which can be comprised of many things. These are some LOGO lists:

```
[A B C D E]  a list of letters
[1 2 3 4 5]  a list of numbers
["N "M "O "P "Q]  a list of variable
   names
[:N :M :O :P :Q]  a list of variable values
[[A B C D E] [1 2 3 4 5] ["N "M "O "P "Q]]
   a list of lists
```

LOGO has a number of built-in, or *primitive*, procedures that process lists. (Another article in this section, "Lyrical LOGO," uses these list processing procedures to begin development of a procedure to write poetry.)

This richness of the language shows off the other end of the LOGO spectrum: It doesn't take much to begin using LOGO, but it is infinitely extensible. You can write any procedures you need;

Continued on p. 137



LOGO SHOOTS FOR THE MOON

by Henry Gorman, Jr.
and the HCM staff

An important element of the LOGO philosophy is the idea that an intricate problem is composed of a hierarchy of smaller problems. Finding a problem's solution is, therefore, merely a matter of solving the smaller component problems. It is apparently easier to manage a large amount of information by subdividing it into smaller units than it is to attack the whole block *en masse*. For example, it is easier and faster to shop for groceries with a shopping list arranged hierarchically by type of food (dairy products, produce, cereals, beverages) than with a random list of items.

It is important that this structured approach to programming be developed through actual problem-solving experience. Students who have only textbook knowledge of the concept may develop unclear structures, which are worse than none at all. LOGO programming is a task which must be learned by doing. Often students can write structured programs without being able to verbalize what they are doing because their knowledge is active and procedural rather than passive and formal.

TI LOGO is tailor-made for the structured approach. Its procedures hide most of the program's details, revealing only the bare bones—the logical structure—of the program. The language also allows problems to be solved by programming with very little structure. Students can quickly compare programs with varying levels of structure and see the advantages of the more highly structured programs as they work on their own projects.

The Apollo Project

APOLLO is a student-written program depicting the lift-off, voyage, and touchdown of the Apollo space mission. One of its procedures, shown below, demonstrates a common beginning programmer's mistake: the tendency to strive for dense, short code. In LOGO procedures, clarity is more important than conciseness. I have, therefore, reworked the student's procedure to add a bit more structure and to make it more modular.

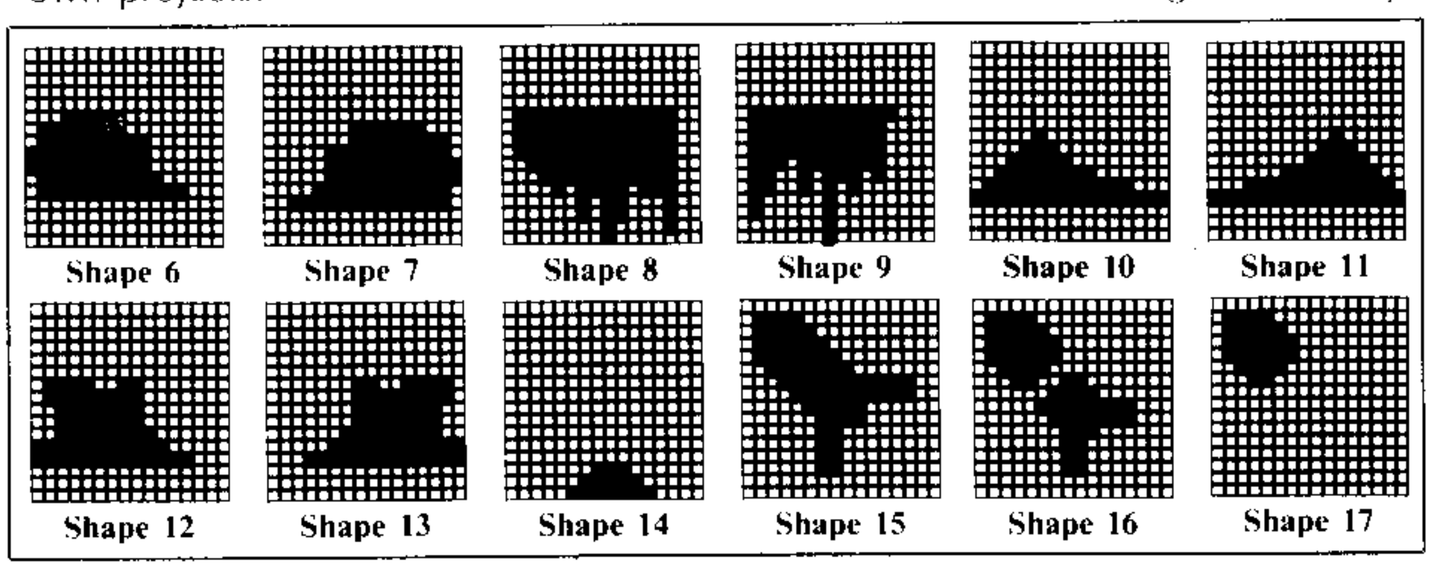
Student Version

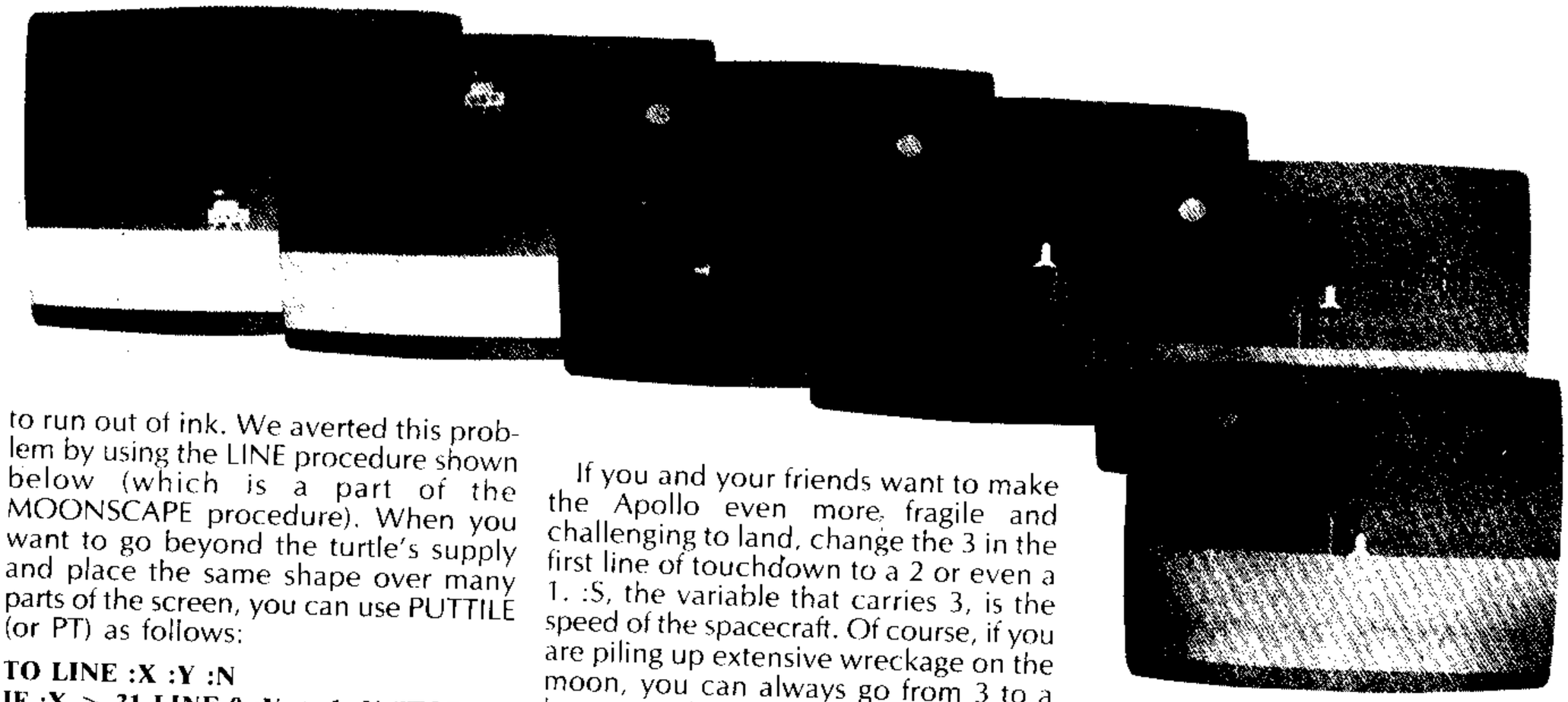
```
TO PREP
VANISH
EARTH
PT 101 15 11
PT 101 15 13
PT 102 15 12
TELL 0
CARRY 3
SC 1
SS 0
SH 0
HOME
TELL 1
CARRY 4
SC 15
SXY -80 80
END
```

Revised Version

```
TO PREP
VANISH
EARTH
MOON
GANTRY
ROCKET
END

TO MOON
TELL 1
CARRY 4
SC 15
SXY -80 80
END

TO GANTRY
TELL TILE 101
SC 1
PT 101 15 11
PT 101 15 13
PT 102 15 12
END

TO ROCKET
TELL 0
CARRY 3
SC 15
SS 0
SH 0
HOME
END
```

To run the program, you key in APOLLO. If you want to clear the sprites and characters from the screen, type in VANISH. The APOLLO program presents a common LOGO problem: The LOGO turtle has a limited number of tiles to use for drawing, so it's easy





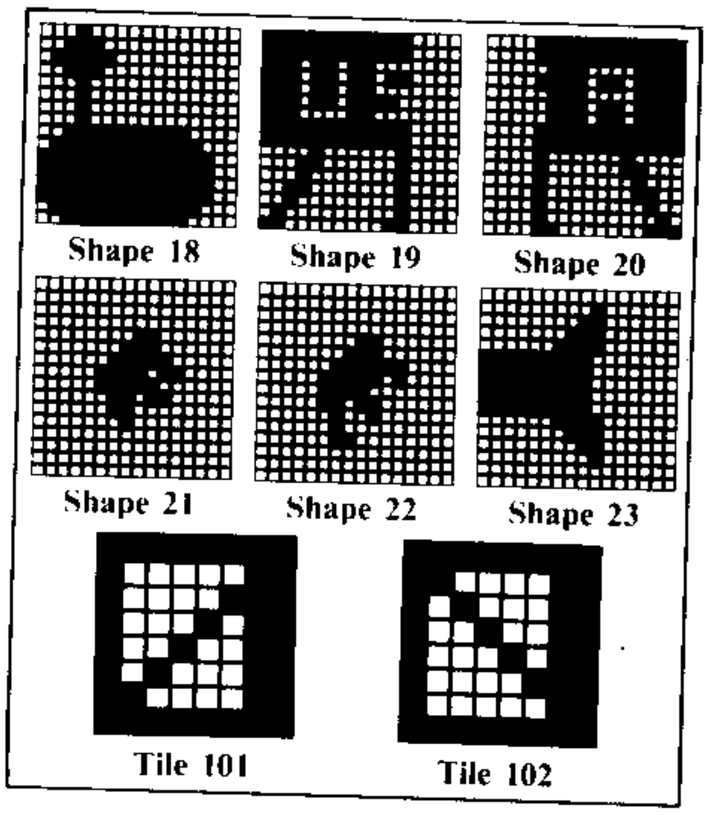
to run out of ink. We averted this problem by using the LINE procedure shown below (which is a part of the MOONSCAPE procedure). When you want to go beyond the turtle's supply and place the same shape over many parts of the screen, you can use PUTTILE (or PT) as follows:

```
TO LINE :X :Y :N
IF :X > 31 LINE 0 :Y + 1 :N STOP
IF :Y > :N STOP PT 1 :X :Y LINE :X
+ 1 :Y :N END
```

This procedure fills the screen from one side to the other, from row :Y to row :N (rows run down the screen 0-23). :N is the number of the last row filled. The procedure places the first tile in column :X of the first row (:Y). The 1 after PT stands for tile 1, but any tile number—or even a variable—could be substituted, as long as it's designed as a tile. It's also possible to rewrite the procedure to fill different blocks of the screen—from top to bottom, for example.

When the Apollo spacecraft settles toward the lunar surface, you will need to take over control from the computer to make the soft landing. Each time you press and release the space bar, you will trigger a burst from your braking rockets.

If you and your friends want to make the Apollo even more fragile and challenging to land, change the 3 in the first line of touchdown to a 2 or even a 1. :S, the variable that carries 3, is the speed of the spacecraft. Of course, if you are piling up extensive wreckage on the moon, you can always go from 3 to a larger number.



If you want to make additional landings, you can just type in the command LAND, and you will quickly find yourself hovering over the lunar surface.

Stepping Through a Program

When you want to double-check these modifications, you may run into a slight problem. Stepping a program (running it line by line) can be a helpful debugging device, but this facility is not built into the LOGO system. Although TI LOGO doesn't have a STEP primitive, you can write one:

```
TO STEP :A
PRINT [PRESS ENTER TO RUN THE
NEXT LINE.]
PRINT SE [NOW STEPPING THE PRO-
GRAM] :A
CHECKOFF BUTFIRST TEXT :A
END
```

Continued on p. 134

C64

Here are some suggestions for making a working version of APOLLO.

1. To start, you will need your LOGO disks and a formated disk with enough space on it (32 blocks free) to hold your completed program. The first step is to load the LOGO language.

2. Next, load the program SPRED from your utility disk (type READ "SPRED). SPRED will enable you to enter the graphics shapes that APOLLO needs. These shapes are divided into two packages for APOLLO, each of which is stored on disk and called up when it's needed.

When SPRED is loaded, you are ready to enter the shapes in the first sprite package, which will be called PREP. Make sure you enter the first package first, because we will carry over shapes 1 and 4 from the first package to the second package. . . that way you won't have to enter them twice. (With some modification the clouds of smoke from the takeoff serve well as the dust clouds of the lunar landing.)

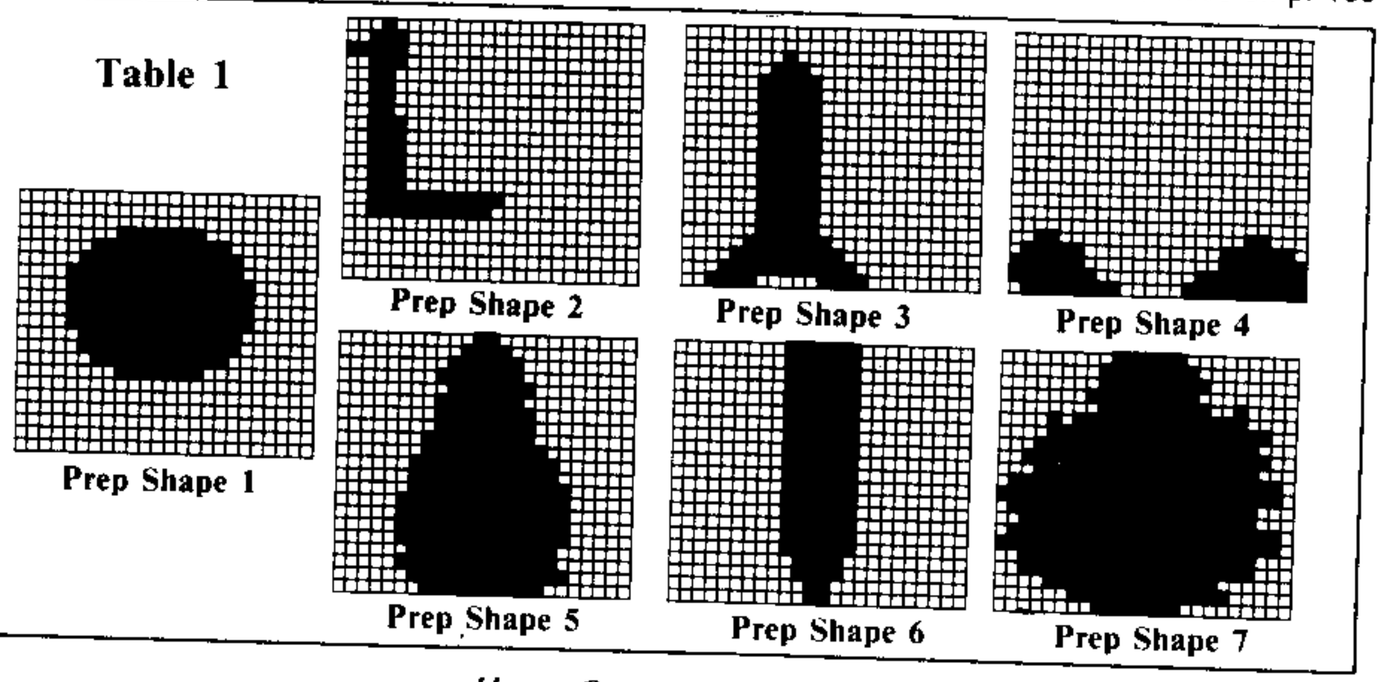
In Table 1 you will find the first package of shapes, numbered 1 through 7. The first sprite you'll "talk to" is sprite 1 (sprite 0 is the turtle and can't be changed), so type TELL 1, and then type EDSH (edit shape). This will put you in the Sprite Edit mode. You will see the large box that will hold sprite shape number 1. Hold down the [SHIFT] key, press the [CLR] key, and you will have a clean sprite box to fill. (Look in your LOGO manual at pages S-10 to S-13 if you need any help entering the sprite shape.) After you've entered the shape, press the [STOP] key and you will return to TOPLEVEL, ready to enter the next shape.

When you have entered all of the shapes, put the disk that you intend to use for your APOLLO program into your drive, type SAVESHAPES "PREP, and the package will be saved.

The second package of sprite shapes is called ORBIT and can be found in

Continued on p. 133

Table 1



LOGO Apollo . . . from p. 131

```
TO CHECKOFF :STUFF
CALL RC "B
IF FIRST :STUFF = [ ] STOP
IF FIRST :STUFF = SE :A [ ] STEP :A
PRINT SE [NOW RUNNING ] FIRST
:STUFF
RUN FIRST :STUFF
CHECKOFF BUTFIRST :STUFF
END
```

To STEP a program, first load the program, and then key in (or load from cassette or diskette) the two procedures above, STEP and CHECKOFF. The STEP and CHECKOFF procedures will be a part of your program even while you SAVE and RECALL. To use the STEP procedure, assume your program is called READ. Give the command STEP "READ. Note that CHECKOFF is called from within STEP, so the one command makes both procedures run. STEP can be modified to allow for the STEPPING of programs inside of STEPPED programs, or to allow for programs with variable input.

Now that we have explored the steps for structured programming as well as a structure for stepping programs, you are ready to run your modified APOLLO. Just keep in mind that all those cliches of the "divide and conquer" ilk apply when programming in LOGO. You must take many tiny steps before you can take that "giant step for mankind" from your APOLLO craft. HCM

TI-LOGO — TI-99/4A

```
TO APOLLO
PREP
BLAST
LIFT
ATMOSPHERE
STRATOSPHERE
ORBIT
LAND
END
```

TI-LOGO — TI-99/4A

```
TO ATMOSPHERE
TELL 5
REPEAT 48 [FLAME ]
TELL 5
SXY 0 (-72 )
TELL 0
SC 15
SXY 0 (-60 )
TELL [0 5 ]
SH 0 SS 3
END

TO STRATOSPHERE
CS
CB :BLUE
TELL 1
CARRY 4
SC 15
SXY -80 80
TELL 5
REPEAT 103 [FLAME ]
TELL 0
SXY 0 (-60 )
TELL 5
SXY 0 (-72 )
TELL [0 5 ]
SH 0
SS 3
CB :BLACK
END

TO ORBIT
SECTION
EJECT
END

TO LAND
TELL [0 1 2 3 4 5 ]
VANISH
CB 1
MOONSCAPE
TELL 0
CARRY 18
SC 4
SXY -20 95
TELL 1
CARRY 19
SC 4
SXY -23 80
TELL 2
CARRY 20
SC 4
SXY -16 80
TELL [0 1 2 ]
SH 180
CALL 3 "S
SS :S
CHECK
END
```

LOGO Apollo . . . from p. 133

TERRAPIN LOGO — C-64

```
TO DOWN :N
 IF :N < 48 THEN STOP
 CURSOR 20 22
 PRINT CHAR :N
 REPEAT 900 [ ]
 DOWN :N - 1
END

TO IGNITION
 TELL 4 PU
 LT 90 FD 4 RT 90
 BACK 59
 PC 1
 ST
 REPEAT 900 [ ]
 LT 90 HT FD 10 ST RT 90
 BIGX
 REPEAT 300 [ ]
 MESSAGE
 HT FD 28 BIGY ST
END

TO MESSAGE
 TEXTCOLOR 1
 CURSOR 16 22
 PRINT [IGNITION]
END

TO COUNTDOWN
 REPEAT 900 [ ]
 CURSOR 19 22
 TEXTCOLOR 1
 PRINT "10
 REPEAT 900 [ ]
 CURSOR 19 22
 PRINT CHAR 0
 DOWN 57
 CURSOR 20 22
 PRINT CHAR 0
 TEXTCOLOR 6
END
```

TERRAPIN LOGO — C-64

```
TO LIFT
 MOTION.1
 MOTION.2
 MOTION.3
 MOTION.4
END

TO MOTION.1
 REPEAT 29 [TELL 3 FD 1 TELL 6 FD 1 PC 8 PC 2]
END

TO MOTION.2
 TELL 5 HT RT 90 FD 12 LT 90 SMALLX ST
 REPEAT 13 [TELL 5 FD 2 TELL 3 FD 2 TELL 6 FD 2 PC 8 PC 2]
END

TO MOTION.3
 TELL 7 HT RT 90 FD 12 LT 90 SMALLX ST
 REPEAT 13 [TELL 3 FD 2 TELL 6 FD 2 PC 8 PC 2]
END

TO MOTION.4
 REPEAT 20 [TELL 3 FD 4 TELL 6 FD 4 PC 8 PC 2]
END

TO MOTION
 TOPLEVEL
END

TO ORBIT.START
 PC 10
 BK 120 RT 90 FD 140
 LT 135 ST
END
```

TI-LOGO — TI-99/4A

```
TO EARTH
TELL TILE 1
SC 14
LINE 0 14 23
END

TO MOON
TELL 1
CARRY 4
SC 15
SXY -80 80
END

TO GANTRY
TELL TILE 101 SC 1
PT 101 15 11
PT 101 15 13
PT 102 15 12
END

TO COUNTDOWN
WAIT 60
PT 49 25 8
PT 48 26 8
DOWN 57
END

TO IGNITION
TELL [2 3]
SC 15
SXY -13 (-3)
TELL 2
CARRY 6
TELL 3
CARRY 7
SXY 13 (-3)
WAIT 30
END

TO MESSAGE
PT 73 20 8
PT 71 21 8
PT 78 22 8
PT 73 23 8
PT 84 24 8
PT 73 25 8
PT 79 26 8
PT 78 27 8
END

TO SECTION
TELL [0 5]
SS 3
SH 315
TELL 5
CARRY 0
TELL 0
CARRY 15
SXY 0 (-65)
TELL 5
SXY 3 (-68) WAIT 400
END
```

TI-LOGO — TI-99/4A

```
TO BURST
REPEAT 10 [WAIT 5 TELL 2 CARRY 12 TELL 3 CARRY 13 WAIT 5 TELL 2 CARRY 6 TELL 3 CARRY 7]
END

TO FLAMES
TELL [2 3]
SC :WHITE
TELL 2
CARRY 10
TELL 3
CARRY 11
TELL 4
CARRY 14
SC :RED
SY -3
REPEAT 3 [BURN :WHITE :YELLOW]
TELL 4
REPEAT 5 [BURN :WHITE :RED]
END

TO MOTION
TELL [2 3 4]
CARRY 0
TELL 5
CARRY 8
SC :RED
HOME
SY -11
TELL [0 5]
SH 0
SS 2
TELL 5
REPEAT 3 [PICKUP]
END

TO MOONSCAPE
TELL TILE 1
SC 15
LINE 0 17 23
END

TO CHECK
WAIT 15
TEST RC?
IFT CALL :S - 2 "S CALL RC "B
CALL :S + 1 "S
TELL [0 1 2]
SS :S
TELL 1
TEST YCOR > 90
IFT OUTOFFUEL
IFF TEST YCOR < -23
IFT TOUCHDOWN
IFF CHECK
END
```

Continued on p. 136

TERRAPIN LOGO — C-64

```
TO BLAST
 COUNTDOWN
 MESSAGE
 IGNITION
 CLEARTEXT
 BURST
END

TO LAUNCHPAD
 BK 58 LT 90
 FD 17 RT 90 ST
END

TO BURST
 TELL 3
 FD 1 REPEAT 300 [ ]
 TELL 6
 PC 2
 BK 63 LT 90 FD 4 RT 90
 ST
 TELL 5
 BIGX PC 1
 LAUNCHPAD
 TELL 7
 BIGX PC 1
 LAUNCHPAD
 TELL 3
 REPEAT 6 [FD 1 REPEAT 300 [ ] ]
 LIFT.MESSAGE
 TELL 3
 REPEAT 16 [FD 1 REPEAT 200 [ ] ]
 CLEARTEXT
END

TO LIFT.MESSAGE
 TEXTCOLOR 1
 CURSOR 12 22
 PRINT [WE HAVE LIFTOFF]
 REPEAT 800 [ ]
END
```

TERRAPIN LOGO — C-64

```
TO LANDED
 IF :VELOCITY > 4 THEN CRASHED
 DUST.2 ST DUST.3 ST
 PRINT [CONGRATULATIONS, COMMANDER]
 HT
 REPEAT 1800 [ ]
 TELL 5 HT BK 20 BIGX BIGY ST
 TOPLEVEL
END

TO IGNITION.EXPLOSION
 TELL 4 PU
 LT 90 FD 4 RT 90
 BACK 59
 PC 1
 ST
 REPEAT 90 [ ]
 LT 90 HT FD 10 ST RT 90
 BIGX
 REPEAT 30 [ ]
 HT FD 28 BIGY ST
END

TO ORBIT
 RESET 7 FULLSCREEN
 READSHAPES "ORBIT
 TELL 1 IEFT 75 FD 120 ST PC 1
 BIGX BIGY ST
 TELL 3 FD 3
 ORBIT.START
 TELL 2
 ORBIT.START
 MOTION.7
 MOTION.8
 MOTION.9
END
```


HCM

LOGO Apollo . . . from p. 135

TI-LOGO — TI-99/4A

```
TO LIFT.MESSAGE
PT 87 20 7
PT 69 21 7
PT 32 22 7
PT 72 23 7
PT 65 24 7
PT 86 25 7
PT 69 26 7
PT 76 20 8
PT 73 21 8
PT 70 22 8
PT 84 23 8
PT 32 24 8
PT 79 25 8
PT 70 26 8
PT 70 27 8
END

TO FLAME
WAIT 5
CARRY 9
WAIT 5
CARRY 8
END

TO TOUCHDOWN
IF :S > 3 CRASH
TELL [0 1 2]
SS 0
END

TO VANISH
TELL :ALL
CARRY 0
SXY 126 96
SH 0
SS 0
SC 0
CB 7
CS
NOTURTLE
END

TO OUTOFFUEL
VANISH
PRINT [YOU ARE OUT OF CONTROL]
PRINT [AND OUT OF FUEL]
PRINT [I'M SORRY]
END
```

TI-LOGO — TI-99/4A

```
TO DOWN :N
WAIT 60
PT 32 25 8
PT :N 26 8
IF :N < 49 STOP
DOWN :N - 1
END

TO BURN :FIRST :SECOND
WAIT 5
SC :FIRST
WAIT 5
SC :SECOND
END

TO PICKUP
WAIT 5
CARRY 9
WAIT 5
CARRY 8
END

TO FIRE
WAIT 5
CARRY 22
WAIT 5
CARRY 21
END

TO CRASH
VANISH
PRINT [YOU HAVE CRASHED]
PRINT [A BILLION DOLLAR CRAFT]
END

TO LINE :X :Y :N
IF :X > 31 LINE 0 :Y + 1 :N STOP
IF :Y > :N STOP
PT 1 :X :Y
LINE :X + 1 :Y :N
END

TO ROCKET
TELL 0
CARRY 3
SC 15
SS 0
SH 0
HOME
END
```

TI-LOGO — TI-99/4A

```
TO EJECT
TELL [0 5]
SS 3
SXY -50 (-15)
TELL 0
SC 15
WAIT 100
CARRY 16
WAIT 100
CARRY 17
TELL 5
CARRY 21
SC :RED
TELL 3
SXY -50 (-20)
CARRY 23
SC 15
SH 180
SS 1
TELL 5
REPEAT 25 [FIRE]
END

TO PREP
VANISH
EARTH
MOON
GANTRY
ROCKET
END

TO BLAST
COUNTDOWN
IGNITION
MESSAGE
BURST
FLAMES
END

TO LIFT
MOTION
LIFT.MESSAGE
END
```

HCM

What is LOGO? . . . from p. 129

after they're written, they're treated by the system exactly like the LOGO primitives. Neither has precedence. LOGO advocates call it a system with no threshold and no horizon.

In addition, LOGO's error messages are not cryptic and curt. They may not always be entirely clear, but they point you to your problem more closely than those in some other languages. The LOGO system was developed with the understanding that we would make programming errors, and that the system should automatically help us find the causes of our errors. For instance, if you type CITU instead of CITY, LOGO will respond:

TELL ME HOW TO CITU

If you forget to give HOUSES an input, a number that it can use in its processing, LOGO responds:

TELL ME MORE

In LOGO, you can look at CITY, fix your error and run the procedure again immediately. This is the *interactive* nature of LOGO.

LOGO is a very clear language to work with because of these features: its low threshold; its use of procedures that divide a program into comprehensible, self-contained segments; its friendly, informative error messages; its unlimited horizon; and its interactive system. HCM

Have No Fear . . . from p. 109

characters in length, but if it is less than six, it must be filled out to six with blanks. The name is given using the TEXT Directive, and must be added in the location where we told the computer the information about our program would be (location >7FE0).

```
7020 71A6 AORG >7FE0
7FE0      TEXT 'TEST  '
7FE6 0000 ■
```

And finally, we add the starting point of the program (in our case, >7D00):

```
7FE6 7D00 DATA >7D00
7FE8      END
```

Then type END and exit the Assembler. Choose the Run option of the Mini Memory menu, type TEST and press [ENTER]. The asterisk will start moving across the screen.

Just one other thing: If, in the first line of our program there is a label—for example, 7D00____ NT LI R0,1—then these two lines would be equivalent:

```
7FE6      DATA >7D00
7FE6      DATA NT
```

It will be best to read the procedure step-by-step several times to understand it thoroughly, and work on the computer as you go along. Ultimately, you will see that it is a logical procedure.

Here is a summary of the steps to follow in making a REF/DEF table entry:

a) Use AORG to get to the First Free Address of the Module. Check the value there.
b) Check the value in the Last Free Address of the Module in the same way.
c) Subtract the value in >701C from the value in >701E. If the number is eight or larger, proceed.
d) Change the value in >701C to the value at the end of your program using the DATA Directive.
e) In the same way, change the value in >701E to a value eight bytes smaller than its present value.
f) Jump to the value you added in part (e) and write the program name (six characters long, including blanks) using the TEXT Directive.
g) At your ending location, use DATA to tell the computer the starting place of your program (either a memory location or a label).
h) Exit the Assembler using the END Directive and run the program.

In this third part of the series, I have tried to give you some more background information about this new language you are learning, to help you understand the advantages and limitations of the *Line-by-Line Assembler* more thoroughly. I have also shown how to save your program by name, using a very simple program as an example. You will notice that all the programs we have seen so far have been quite simple and straightforward, with a minimum of instructions. I selected these programs on purpose to keep everything as clear as possible until you had more knowledge before launching into more complicated ones. HCM

.....Texas Instruments TI-99/4A — COMPUTERS, COMPONENTS AND SOFTWARE.....

TEX-COMP™

TI USERS SUPPLY COMPANY "YOUR LUBBOCK CONNECTION™"

America's Number One TI computer retailer

TI-99/4 PRODUCTS AT PROFESSIONAL PRICES

AN IMPORTANT MESSAGE FOR OWNERS OF TI-99/4A COMPUTERS

As many of you are already aware, TEX-COMP has been selling and supporting the complete TI-99/4 home computer line since its introduction in late 1979. As a result of our prompt service and competitive prices, we have become the leading independent TI home computer retailer in the country. As an "independent" retailer we are not affilated with any other TI sales entity such as distributors, hardware co-ops, or premium firms and we therefor only obtain our TI products through legitimate sources. We are a "direct" TI retail account which means that we have met TI's high standards for this status.

As a direct TI account, TI has asked them to assist in liquidating much of the TI home computer product that has been returned unopened by many of the mass merchandisers. We are also buying the TI inventory of the many TI sales organizations that have dropped the TI product line or have gone out of business (or both). We have recently bought out the stock of Tronics, Byte Industries, Blatt Distributors, California Radio Distributors, Olympic Sales, Galaxy Computer Stores, G.A.M.E.S. and J.C. Penny. As these product arrive, we will be announcing availability on our ***HOT LINE*** *(818) 366-6631 . We recommend that orders be placed by credit card as personal checks take up to three weeks to clear and items could be sold out in that time. You will only be charged for what is actually shipped. New third party items will be arriving on a continuous basis so we recommend checking our HOT LINE often.*

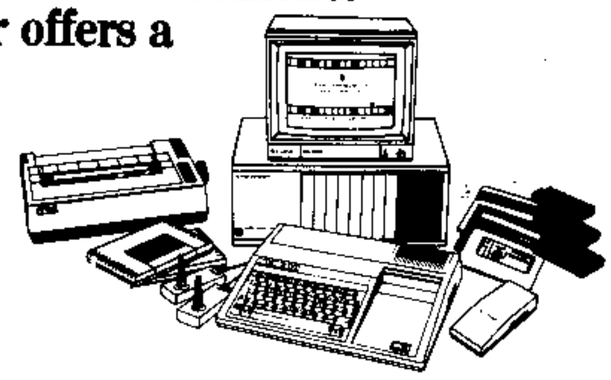
We only will carry quality lines of TI compatable hardware and software so you can look to TEX-COMP as so many have in the past, as your primary source for 99/4A support. As you can all well imagine , things have been very hectic in the weeks following TI's announcement of its leaving the home computer field and we at TEX-COMP want to thank all of you for your understanding for any delays in order processing or fullfillment during this period. We also want to extend our apology to those whose orders for the PHP 4000 system could not be filled due to TI being unable to fill the orders we and other dealers placed in good faith.

The important thing to keep in mind is that it appears that the 99/4A with its huge installed base will continue to be supported by third party producers at a far better level than ever before. You own the finest value in a home computer as to both price and quality and it will provide you and your family with many hours of enjoyment, enrichment, and satisfaction for years to come.

TEX-COMP INTENDS TO BE AROUND TO SEE THAT IT DOES.....

TEXAS INSTRUMENTS

The one to start with is the one to stay with. The Texas Instruments Home Computer offers a big choice of accessories to let you expand and customize your system.



TEXAS INSTRUMENTS

TEX-COMP • P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •

SEND FOR THE LATEST TEX-COMP CATALOG OF TI -99/4A PRODUCTS AND ACCESSORIES. ONLY $2.00 (cash or check only) We include a $5.00 merchandise certificate with each order

The data base processing of the program happens in lines 115 to 170. We OPEN the file (which we call file 2) on device #1 (the cassette), and tell the VIC that it is a *read only file* (that's the meaning of the 0 in the secondary address).

We then INPUT the special code numbers and recipe names one at a time and check them. If the VIC finds a match, it stores that information in the previously DIMensioned arrays by incrementing the counter variable A.

Once the tape is read, the program closes the file and tells us how many recipes it matched. Then it moves into the random selection process.

Lines 180 through 205 generate as many random numbers as recipes we asked for at the beginning of the search (unless we asked for more than were found). Then it prints one less than the actual number found. Taking one less random number significantly increases the speed of selection.

The remainder of the program prints out each recipe's name and code number. Finally the program displays MORE?. Typing a Y for Yes takes you back to the beginning of the program. Hitting any other key will end the program.

The *Make a Tape* program begins with a title display and mini menu, from lines 50 through 60. Line 65 determines which path was selected. If you push any button except 1 or 2, you will be sent back to line 60 to try again. Lines 70 through 90 open the tape file and allow you to input the data. They also then output your data onto the tape via the PRINT# statements.

The Add to Existing Tape option begins at line 95. It is similar to the section which reads in info from the *Menu Planner*, but instead of testing each entry, it just reads it into memory. Later (lines 115 to 160) these same entries are returned to another tape file, but the file is left open, so you can add more recipes. Again, when you type END, it closes this new file and ends the program.

The *Menu Match* program uses the same principles as the other two. The real work is done in lines 95 through 125. Here is where the recipe's name is read in and checked for a match.

In order to speed up execution time, line 115 first looks to see if there are fewer unchecked characters in the name than in the key word. If they are less, then there is no possibility of a match and that recipe is rejected.

The program is written in three parts to leave as much of the VIC's memory free as possible. Minor modifications will allow the parts to be merged into one neat package if you have a memory expander, or a Commodore 64. Although *Menu Planner* is a simple and specific system, it could easily be improved and adapted to make your VIC-20 a small but useful data base management system.

Now if anyone out there finds a way to have the VIC do the ironing...

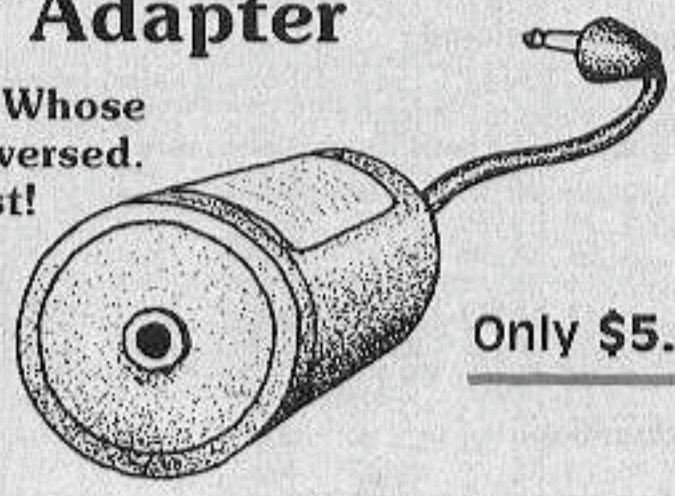
HCM

99'er.WARE™
P.O. Box 5537
Eugene, Oregon 97405
Tel. (503) 485-8796

The TEX-SETTE™ Adapter

For Those Cassette Recorders Whose Remote Jack Wiring is Reversed. Cassette Compatibility At Last!

If the TI-99/4A will not control your cassette recorder's motor (automatic start/stop), we have the solution for you.



Only $5.95

DUST COVERS

Features:

- Equipment Protection
- Handsome Appearance
- Custom-Fit
- Antistatic Treated
- Quality Construction

A	10" Color Monitor Cover	$10.95
	13" Color Monitor Cover (now shown)	$12.95
B	Peripheral Expansion Box Cover	$12.95
C	TI's 99/4 Matrix Printer Cover (same as Epson MX-80 Cover)	$ 9.95
D	Cassette Recorder Cover (1 size fits up to 10" x 6")	$ 4.95
E	TI-99/4(A) Console Cover	$ 8.95
F	Speech Synthesizer Cover	$ 3.95
G	Peripheral Box Cover (Specify: 32K Memory Expansion RS232 Interface, or Disk Controller)	$ 5.95
H	Thermal Printer Cover	$ 8.95
I	Disk Memory Drive Cover	$ 5.95

Dealer Inquiries Invited

► THIS ORDER FORM MAY ALSO BE USED FOR: ◄
The Best of 99'er (p.163), *99'er Directory* (p.166), and Simon's Saucer (p.148)

NAME _____ PLEASE
ADDRESS _____ PRINT
CITY _____ STATE _____ ZIP _____

Check enclosed ☐ MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my: ☐ VISA ☐ Master Card ► _____ Expiration Date

Account No.

Tel. No. _____ Signature _____

QTY	ITEM(S)	PRICE	SHIPPING	AMOUNT
	Best of 99'er	$19.95	$1.50 ea.☆	
	Best of 99'er On Tape	$35.00	$1.00 ea.☆	
	Book and Tape Package	$49.95	$2.50 ea.☆	
	Finder Binder (no tapes)	$11.95	$1.50 ea.☆	
	Finder – Binder w/6 blank tapes	$16.95	$1.50 ea.☆	
	Finder-Binder w/12 blank tapes	$21.95	$2.50 ea.☆	
	12 Blank Tapes	$14.00	$2.00 ea.	
	Texsette Adapter	$5.95 ea.	.50 ea.	
	Simon's Saucer	$9.95	$2.00 ea.☆	
			Subtotal	

Qty	ITEM(S)	PRICE	SHIPPING	AMOUNT
	Back Issues (Circle Issues Desired) Volume 1 No. 6, Nov '82, Dec '82, Jan '83, Feb '83, Mar '83, Apr '83, May '83, Jun '83, Jul '83, Aug '83, Sept '83, Oct '83, Nov '83 (Handling incl.)	$3.95 ea.	Foreign Surface add $2.00 ea. Foreign Air add $4.00 ea.	
	Dust Covers A B C D E F G H I		.50¢ ea. cover	
	99'er Directory (Handling incl.)	$5.00		

☆Foreign surface add $3.00 for shipping.

Foreign Air Inquire

Subtotal	
Total Order	
Handling	$1.50
TOTAL PAYMENT ENCLOSED	

Enclose check, money order or credit card information to:
99'er ware, Dept M1
P.O. Box 5537
Eugene, OR 97405
or
Use VISA/MasterCard and Order Toll-Free
1-800-828-2212
In Oregon, Alaska, and Hawaii telephone
(503) 485-8796
Please allow 10 days for your order to be processed.

Addendum to *99'er Directory of Commercially Available Software, Peripherals, and Accessories*, Summer/Fall 1983 Edition

Program Title	Mfr. No.	Medium	Req'd Accs.	Opt. Accs.	Description	Sugg. Retail
Home Management/Personal Finance						
Electricity Manager	FU020	Dsk	DD&C, X-BASIC, 32KMExp		Monitor/budget energy	34.95
Household Inventory	FU020	Dsk	DD&C, X-BASIC		Record home assets	34.95
Med Alert!	FU020	Dsk	DD&C, X-BASIC		Accident prevention	34.95
Amortization Schedule	FU020	Cass/Dsk	CR&C/DD&C, X-BASIC, RS232		Calculate mortgage	49.95/54.95
Business/Professional						
Accounts Payable	FU020	Dsk	DD(2)&C, X-BASIC, RS232, Printer, 32KMExp		Payable functions	149.95
Accounts Receivable	FU020	Dsk	DD(2)&C, X-BASIC, RS232, Printer, 32KMExp		Receivable functions	149.95
Billing	FU020	Dsk	DD(2)&C, X-BASIC, RS232, Printer, 32KMExp		Billing functions	149.95
General Ledger	FU020	Dsk	DD&C, X-BASIC, RS232, Printer	2nd DD&C	General ledger functions	149.95
Inventory Management	FU020	Dsk	DD(2)&C, X-BASIC, RS232, Printer, 32KMExp		Maintain inventory	149.95
Mailing List	FU020	Dsk	DD(2)&C, X-BASIC, RS232, Printer, 32KMExp		Versatile list	99.95
Payroll	FU020	Dsk	DD(2)&C, X-BASIC, RS232, Printer, 32KMExp		Print payroll checks	149.95
Trendline	IM010	Cass/Dsk	CR&C/DD&C	X-BASIC	Advanced stock market program for the 99/4A	27.95
Early Education						
Tell Me A Story	FU020	Cass	CR&C, X-BASIC		Encourage child's reading interests	19.95
Skill Builder 1	IM010	Cass	CR&C, Joysticks		Two skill games: Bingo Duel and Number Hunt	19.95
Continuing Education						
Wall Street Challenge	IM010	Cass	CR&C		Computer simulation of the stock exchange	19.95
Wildcatting	IM010	Cass	CR&C		Simulate a hidden oil deposit, which you try to find	19.95
Learning Morse Code	IM010	Cass/Dsk	CR&C/DD&C		Teaches the user the correct sound of each character in easy steps	24.95
Teaching and Administrative Aids						
Galactic Spelling	IM010	Cass/Dsk	CR&C/DD&C, TE-II, SpSyn		Self-teaching aid, tutor, game, and quiz	19.95
Word Processing						
Word Processor	FU020	Dsk	DD&C, X-BASIC, RS232, Printer, 32KMExp	2nd DD&C	Create, maintain, print	149.95
Direct Writer II Word Processor	IM010	Dsk	DD&C, X-BASIC, 32KMExp, RS232, Printer		32 to 132-character line length	69.95
Programming Aids						
Compactor Plus	IM010	Dsk	DD&C, X-BASIC, RS232, Printer		Programmer tool—compact, analyze, delete, resequence	29.95
Entertainment						
Air Missile Command	FU020	Cass	CR&C, X-BASIC		Protect city from aliens	19.95
All *Star Baseball	FU020	Cass	CR&C, X-BASIC		Two-player game	19.95
All *Star Bowling	FU020	Cass	CR&C, X-BASIC		1 to 8 players	19.95
Battleground	FU020	Cass	CR&C, X-BASIC		Calculate enemy's position and destroy	19.95
Casino Pack	FU020	Cass/Dsk	CR&C/DD&C		Blackjack/Craps/Slot	19.95/21.95
Challenge I	FU020	Cass	CR&C		Leap-frogging	16.95
Challenge II	FU020	Cass	CR&C		Nim/Tic-Tac-Toe	16.95
'Chutes & Sharks	FU020	Cass	CR&C, X-BASIC, 32KMExp		Parachutist's fall	19.95
CIA Adventure	FU020	Cass/Dsk	CR&C/DD&C, X-BASIC, 32KMExp		Find missing ruby	19.95/21.95
'Copter Capers	FU020	Cass	CR&C, X-BASIC		Rescue from burning building	19.95
Cosmic Drift	FU020	Cass	CR&C, X-BASIC		Action game	19.95
Death Station	FU020	Cass	CR&C, X-BASIC		Try to break death spell	19.95
Doomcastle	FU020	Cass	CR&C, X-BASIC		Recover sapphires	19.95
Dr. Nuttier	FU020	Dsk	DD&C	SpSyn, TE II	Mind-blowing answers	21.95
Dragon Game	FU020	Dsk	DD&C, X-BASIC, 32KMExp, EdAss, Joysticks		Action game	24.95
Fly Snuffer	FU020	Cass	CR&C, X-BASIC, Joysticks		Snuff out flies	19.95
Flying Saucer	FU020	Cass	CR&C, X-BASIC		Action game	19.95
Fowl Play	FU020	Cass	CR&C, X-BASIC		Help hen across road	19.95
Galactic War	FU020	Cass	CR&C, X-BASIC		Battle the alien craft	19.95
Getcha!	FU020	Cass	CR&C, X-BASIC, Joysticks		Action game	19.95
Hopper	FU020	Cass	CR&C, X-BASIC		Jump frogs across river	19.95
The Hustler	FU020	Cass	CR&C, X-BASIC, Joysticks		Eight ball/rotation	19.95
Ice Caverns of Xen	FU020	Cass/Dsk	CR&C/DD&C, X-BASIC, 32KMExp		Reasoning ability	19.95/21.95
Laser Battle	FU020	Cass	CR&C, X-BASIC, Joysticks		Action game	19.95
London Blitz	FU020	Cass	CR&C, X-BASIC, Joysticks		Shoot down bombers	19.95
Mars Marauder	FU020	Cass	CR&C, X-BASIC, Joysticks		Navigation skills	19.95
Mars Rover	FU020	Cass	CR&C, X-BASIC		Action game	19.95
Memory Match	FU020	Cass	CR&C, X-BASIC		Make most matches and win	19.95
Minefield	FU020	Cass	CR&C, X-BASIC		Step on mine & explode	19.95
Moonbus	FU020	Cass	CR&C, X-BASIC		Defend pyramid, shoot aliens	19.95
Moonduster	FU020	Cass	CR&C		Try landing on moon	19.95
Monster Craze	FU020	Cass	CR&C, X-BASIC		Action game	19.95
Newton's Revenge	FU020	Cass/Dsk	CR&C/DD&C, MiniMem	Joysticks	Catch falling apples	24.95/24.95
Newton's Revenge	FU020	Dsk	DD&C, X-BASIC, 32KMExp/EdAss	Joysticks	Catch falling apples	24.95
Sam Defense II	FU020	Cass	CR&C		Three difficulty levels	19.95
Starship Concord	FU020	Cass	CR&C, X-BASIC		Real-live action game	19.95
Starship Concord	FU020	Cass	CR&C, X-BASIC, 32KMExp		Real-live action game	19.95
Viral Vendetta	FU020	Dsk	DD&C, X-BASIC, 32KMExp/EdAss		Protect cells from virus	24.95
Wall Street	FU020	Cass	CR&C		Realistic simulation	19.95
Wall Street	FU020	Cass	CR&C, X-BASIC		Realistic simulation	19.95
Word/Numberscope	FU020	Cass	CR&C, X-BASIC		Enhance speed reading	19.95
Tournament Brick Bat	IM010	Cass	CR&C, Joystick		Action-skill game	14.95
Strategy Pack I	IM010	Cass	CR&C	Joysticks	Two strategy games: Roman Checkers and Frame Up	19.95
Mind Master	IM010	Cass	CR&C		Strategy game in which computer designs hidden problems & reports the results of each guess	14.95
Demon of the Planes	IM010	Cass/Dsk	CR&C/DD&C, X-BASIC		Graphic adventure into the astral plane	19.95
Mr. Mouse	IM010	Cass/Dsk	CR&C/DD&C, X-BASIC	Joysticks, SpSyn	Talking game in X-BASIC	19.95
Asteroid Destroyer	IM010	Cass/Dsk	CR&C/DD&C, X-BASIC	Joysticks	Arcade game	19.95
Shuttle Command	IM010	Cass/Dsk	CR&C/DD&C, X-BASIC	Joysticks	Destroy the Russian Attack Military Satellites	19.95
Quimbee	IM010	Cass/Dsk	CR&C/DD&C, X-BASIC		Game of skill and chance	19.95

CC = Command Cartridge, CR&C = Cassette Recorder & Cable, DD&C = Disk Drive & Controller

Due to a misinterpretation of information from Wycove Systems, Ltd., their three Wycove Forth programming aids were listed in the September *99'er Directory* as N/A. The programs were and are available.

The telephone numbers listed for C. A. Root Associates in the September *99'er Directory* should be disregarded. To contact C. A. Root Associates, call (206) 941-6984.

Company	Mfr. No.
FUTURA Software, P. O. Box 5581, Fort Worth, TX 76108, (817) 732-1687	**FU020**
Image Computer Products, Inc., P. O. Box 3761, Cherry Hill, NJ 08034, (609) 667-2526	**IM010**

COMING

TO A NEWSTAND NEAR YOU

A GRAPHICS EXTRAVAGANZA

- More Exciting Tips for 3-D Apple Graphics!
- ANIMATING your PCjr
- *Surprising* Shapes on the VIC-20
- Quick 'n' Easy C-64 Graphics
- Gaming: Fight Killer Robots & Bloodthirsty Cannibals
- LOGO Expands with Data Files
- Customizing PCjr's Keyboard
- Sprites Alive! on the 99/4A
- A TI Construction Project

Game Reviews! New Products! Industry News!
and much, much more!

HOME COMPUTER™ magazine

NEXT ISSUE

FACEMAKER

A Review
by Sharyn Lyon
HCM Staff

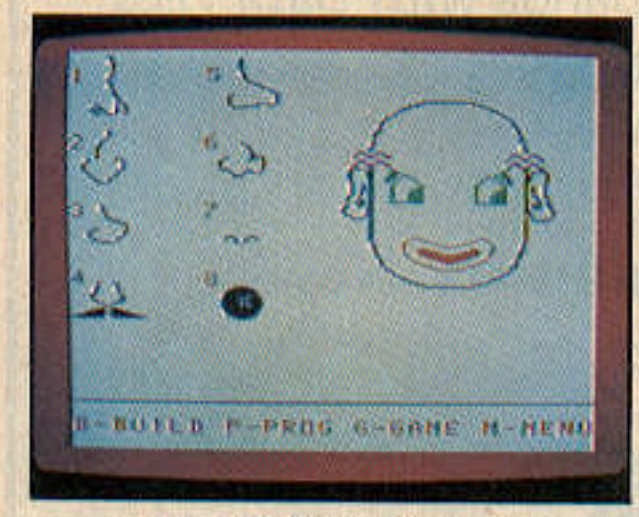
Name:	**Facemaker**
ProgramType:	**Educational game**
Machines:	**Apple, Commodore 64, TI-99/4A, Atari**
Distributor:	**Spinnaker Software Corp. 215 First Street Cambridge, MA 02142**
	Texas Instruments P.O. Box 10508 Lubbock, TX 79408
Price:	**C-64 cartridge and Apple (II+,IIe only) disk $34.95 TI cartridge $29.95**

	Poor	Fair	Good	Excellent
Performance				
Documentation				
Ease of Use				
Engrossment				
Graphics				

All of us have, at some point in our childhood, heard the phrase, "Don't make faces like that! Your face will freeze that way!" That prediction usually flew off the lips of some well-intentioned adult attempting to persuade children to stop their contortions. Now there is *Facemaker*, and the prospect of hours of perfectly harmless and temporary facial transformations.

Using *Facemaker*, children aged 4-10 can entertain themselves for hours while exercising their imaginations, learning how it feels to control a computer, and stretching their memories.

The game starts with a catchy tune, a blank face, and a menu of three options. The user can choose to Build a Face, Program a Face, or Play a Game. You don't have to take up these options in any particular order: If you want to play the game or program a face, you can do it without first building a face.



TI build-a-face option screen

Build a Face

There is an implied hierarchy in this trio, however; so let's start by picking the Build option. This selection has a menu of its own that lists the parts you can use to create your face: Mouth, Eyes, Ears, Nose, and Hair. Next to each word is a picture that makes it possible for non-readers to play on their own with almost no adult intervention. Building a face with five facial parts in eight different styles gives a child all the fun of Mr. Potato Head without any pieces to break or lose. One feature I did miss from the old vegetable version was the selection of hats. Maybe *Facemaker* is missing a good bet by not including this topper.

TI main menu screen

Program a Face

When you and your imagination have settled on a face you'd like to see in action, it's time to try your hand at programming your face. Without losing your creation, you can go back to the first menu and select Program a Face. Each part of your face can perform its own action, and each of the six actions has its own tonal accompaniment. In other words, your face not only smiles (S), frowns (F), sticks out its tongue (T), winks (W), wiggles its ears (E), and cries (C)—it also sings! (Although this addition may seem like a mere frill at this stage, the action/music combination will work like a mnemonic device when you get to the Play a Game option.) To program your face, simply type in the appropriate letter and watch the face react. If you reach the maximum "program" length of 25 actions, the face will perform the actions you assigned automatically, without your having to press any keys.

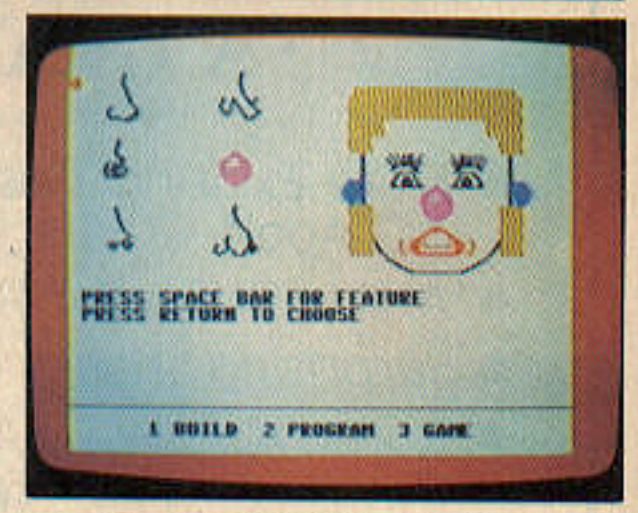
Play a Game

When you think you can keep all these actions in your head, try the Game option. The game does not have instructions, except in the TI version, but it does take you through a trial run to demonstrate the response it's after. In this memory game the face will perform an action. If you type in the letter that stands for that action, it will repeat the action and add a new one. This process continues through up to 25 actions. If you are a whiz, and can keep all 25 actions straight, a special reward comes your way!

You Can't Tell One from the Other Without a Program

There are minute differences among the three different machine versions that we looked at. The main difference among them is in the graphics programming. My personal favorite was the TI version because of the variety of faces possible and because there was a pattern, an inherent order, that satisfied my left brain. To children, who love discovering patterns in things that seem to be without order, this was an appealing feature. I watched a group of children play the game, and they were so delighted with themselves when they discovered that all the option 8 features made a sensible face instead of a silly one, that they felt compelled to share their discovery with anyone who happened to be within shouting distance!

The Apple version has precise, colorful, and varied graphics. The faces range from strangely grotesque to appealingly funny. The Apple version also puts color to good use, but its slowness of execution is a bit annoying. Each time you branch to a different mode, you must wait through two screen re-paintings before you can play the new option.



Commodore menu screens

The Commodore 64 version was the most disappointing, principally because its graphics were so imprecise. For example, of the 8 noses, 2 looked very similar, only one was in color, and the other 5 were more like squiggles than noses.

But let's *face* it—even with these minor flaws, *Facemaker* is a program the entire family will enjoy. It's a fun way to learn a little about the computer keyboard and to feel the excitement of programming the computer to do what you tell it to do. It exercises both your memory and your imagination. HCM

HCM Program Bug

DeBUGS on DisPlay

Bugs—problems that cause programs not to function properly—are an inevitable consequence of programming. Programs are complex systems that are difficult to test exhaustively. And occasionally bugs slip through even the best testing procedures. When such a bug in one of our programs comes to our attention, we print a correction in this column so that our readers can correct their programs as soon as possible.

The author of the article **Public Investigator** (August, 1983) informs us that two errors slipped by him and onto the master tape he sent us. Lines 930 and 950 should be altered as follows:

```
930 FOR I=I TO Q
950 NEXT I
```

This adds enough "@" signs to finish the response string if a user stops in the middle of the questionnaire.

As we noted before, our version of **Success Formula** wasn't too successful. The author has sent along the following changes necessary to make it work better:

```
2920 RESTORE 5830
2960 RESTORE 5835
```

and

```
5835  DATA  7,6,10,10,2,16,5,20,10,
      22,3,24,4,25,5
```

Our thanks to David M. Douglas and Oris Bud Davis for straightening us out.

Robert Schenk, author of **Grisly Adventure** (October 1983), has just written to say that a reader has informed him of a bug in the program. Line 2850 should read as follows:

```
2850 IF (I<>G) + (P<>F) THEN 2880
```

"As written," he writes, "it is imposible to kill the bear by shooting either right or left. The only way to kill the bear is to shoot it from above or below.

"Also, I have had many calls and letters form people who got a data error in line 270. They do not realize that their problem is not a mistake in the **READ** statement in 270, but in a **DATA** statement at the end of the program. To isolate the mistake, one can type in **PRINT B,A$** when the error occurs and the computer will tell one the last value that the program read correctly. Perhaps you can pass this along to your readers because they will have similar errors in many of the other programs you publish."

We're glad to publish Robert's debugging hint, and thank him for the correction to **Grisly Adventure.**

October's gremlins must have slipped into the wizard's keep. In your **Escape from Wizard's Keep**, if you get into the room with no exit, you really get trapped. Instead of a chance to play again, you got * SYNTAX ERROR IN 970, and the Extended BASIC screen. Line 970 should read:

```
970 CALL COLOR(13,1,1)
```

If you tried to modify **The Poor Man's Program Loader** by Rick Rothstein (Letters to the Editor, p. 7, November 1983) according to the directions accompanying it, you probably got as befuddled as we did when we tried. The directions for modifying the program names LOADER and CAT should refer you to lines 230 and 250 respectively.

HCM

Apple Pie

Lines 200-270 calculate the four corners of the top of the box. There are two coordinates for each corner. We don't need to calculate the bottom four corners of the box because they will always be directly under the top four corners. Notice that A is the only variable used to find all four points of the top of the box. This is because the value of PI represents one complete rotation around an imaginary circle, and the other points of the box can be represented as positions on the circle by adding fractions of PI to A. If you add .25*PI to A, you get a point one-quarter of the way around the circle from A. The corner opposite A is .5*PI+A, and .75*PI gives the corner three-quarters of the way around the top of the box.

The size of the box is specified by multiplying COS and SIN by a chosen number, in this case 10 or 20. The number added to this total is the offset, and is used for screen positioning. The actual box is drawn in line 310, using the points which were calculated earlier. Line 300 sets the color, which is white in this listing.

If you don't erase your old box before creating a new one, you will soon end up with a white blob on the screen. The HPLOT statement in line 290 will draw the box created in the previous pass in black, effectively erasing it from the screen. Line 280 sets the line color to black. In line 320 the coordinates from the new box are saved so that *it* can be erased on the next pass, before a new one is drawn.

Animation Plus

The first time I worked with graphics on the Apple, I looked for a facility to redefine the character set and discovered an even better way of dealing with graphics shapes. Often when you are designing a game, the line-drawing commands aren't quite suitable. That's where the Apple DRAW statement comes in handy. This statement lets you draw complex shapes on the screen without having to redefine characters or draw every line. In addition, you can easily expand the shape with the SCALE= function or rotate it with the ROT= function. By defining several shapes and using these functions you can create animation that rivals "sprite" (smoothly moving screen objects) capabilities. In this program I created a simple plus sign, then expanded and rotated it at the same time. The result is interesting, and quite easily accomplished, as you can see by the size of the listing.

The FOR–NEXT loop in lines 180-190 reads the data in line 200 and places it in memory beginning at address 7676, which is just below the high resolution page 1.(High resolution has two pages for memory storage, but can display only one at a time.) This data sets up a shape table to create the plus sign. The first four numbers in the table are the table

```
100 REM  **********
110 REM  * ANIMATION *
120 REM  **********
130 REM  BY WILLIAM K. BALTHROP
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 1.1
160 REM  APPLESOFT
170 HGR
180 FOR X = 7676 TO 7686
190 READ A: POKE X,A: NEXT X
200 DATA 1,0,4,0,46,62,62,60,44,0
210 POKE 232,2: POKE 233,29
220 HCOLOR= 3
230 FOR X = 1 TO 63 STEP 2
240 ROT= X
250 SCALE= X / 2 + 1
260 XDRAW 1 AT 140,96
270 XDRAW 1 AT 140,96
280 NEXT X
290 GOTO 230
```

index. The first value indicates the number of shapes defined in the table. The second number is not used. The third and fourth numbers specify the offset from the start of the index to the first shape data. In this case the offset for the only shape is 4 since there are four bytes before the shape data starts. If you add another shape to the end of this one, you will need an entry in the index. The first value will have to be changed from 1 to 2, and the third value will have to be changed from 4 to 6. Then the start of the new table must be inserted between the offset of the first table and the start of the first shape table. The new DATA line would look something like this:

DATA 2,0,6,0,13,0,44,46,62,62,60,44,0,. . .(new shape data)

Before the shape table can be used, the address of its index must be POKEd into memory (line 210); otherwise, the computer won't know where the table is. The table can be put anywhere in free memory.

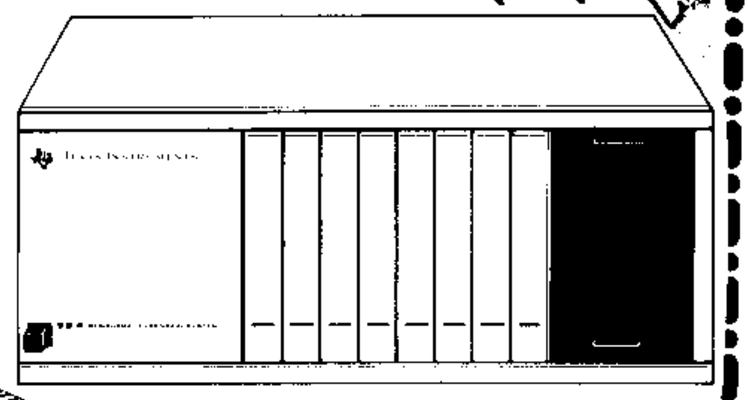
The HCOLOR= function in line 220 sets the color for the DRAW and XDRAW commands, as well as the HPLOT command. The ROT= function controls rotation. The loop in line 230 increments up to 63, representing one full rotation. Values from 64 to 127 will cause one rotation, and any number over 255 will cause an error. The SCALE= function in line 250 sets the scale of your shape with only one command.

Lines 260-270 are the statements responsible for placing the shape on the screen and then erasing it. The XDRAW statement is used because it will cause the shape to automatically erase itself the second time it's drawn in the same place. Try leaving line 270 out of the program, and you will see the results. Then slip it back in, sit back, and enjoy the show. The old spirograph was never quite like this.

HCM

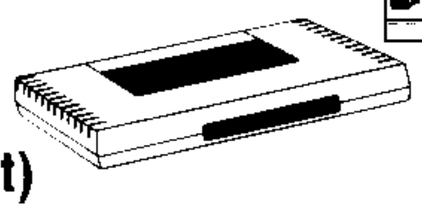
DHEIN'S

CONTINUES TO SUPPORT TI-99/4A OWNERS



NOW IN STOCK

PERIPHERAL EXPANSION BOX AND ALL PERIPHERALS

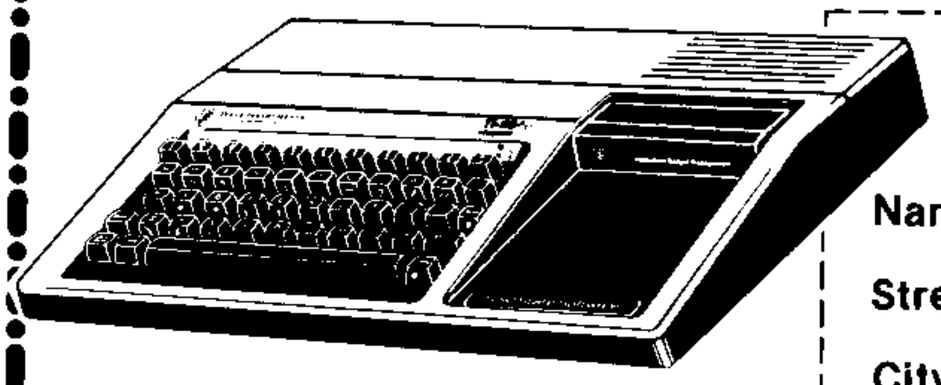


(Check for availability as they are going fast)

MOST TI CARTRIDGES, TAPES AND DISKS STILL AVAILABLE

LOTS OF THIRD PARTY PRODUCTS NOW IN STOCK — MUCH MORE TO COME!

SEND COUPON OR POSTCARD FOR **FREE** COMPLETE DESCRIPTIVE LISTING



COUPON--MAIL TODAY

FREE COMPUTER BULLETIN

Name _____

Street _____ State _____

City _____ Zip _____

The best combination of Price, Service and Quality is TRUE VALUE.

More than just a name, it is our way of doing business.

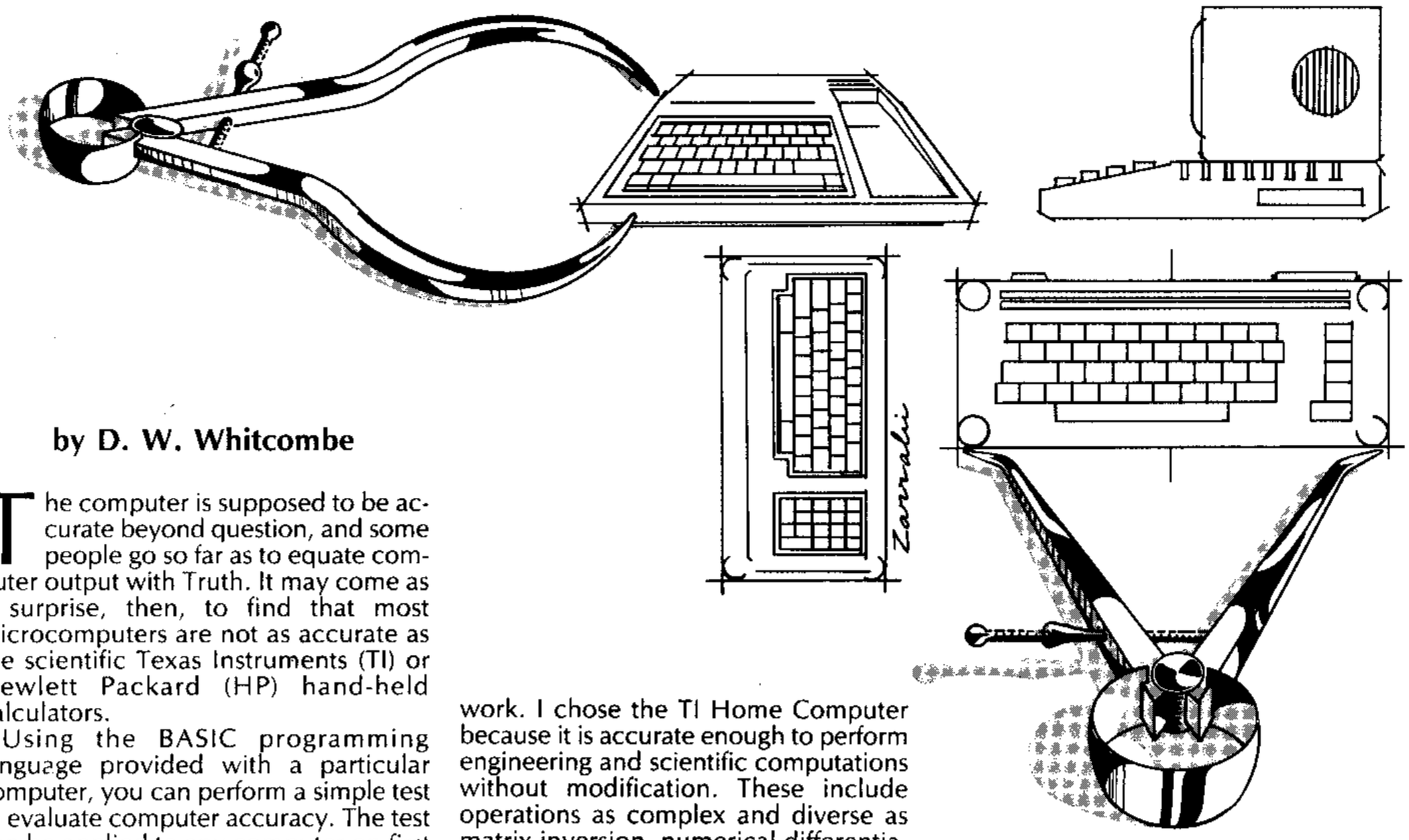
DHEIN'S

True Value HARDWARE STORE ®

(319) 236-3861

7 W. Airline Hwy.
Waterloo, IA 50701

Microcomputer Accuracy



by D. W. Whitcombe

The computer is supposed to be accurate beyond question, and some people go so far as to equate computer output with Truth. It may come as a surprise, then, to find that most microcomputers are not as accurate as the scientific Texas Instruments (TI) or Hewlett Packard (HP) hand-held calculators.

Using the BASIC programming language provided with a particular computer, you can perform a simple test to evaluate computer accuracy. The test can be applied to any computer on first inspection. It evaluates the following forms for X = .01 to 1.0 in small steps:

1. ATN(TAN(X)) – X
2. TAN(ATN(X)) – X
3. LOG(EXP(X)) – X
4. EXP(LOG(X)) – X
5. (X ^ 10) ^ .1 – X

The results are as follows:

COMPUTER	SIGNIFICANT FIGURES
Control Data Cyber	15
TI-99/4,TI-99/4A,TI CC-40	13
TI 59 (hand-held)	11
Apple,(1) Commodore,VIC-20	9-10
TRS 80 Model 1	7
TRS Model 100	7
IBM PC, COMPAQ	7
Epson HX-20	7
Atari 1200 XL	7
Sanyo MBC-1000	6

(1) The Apple gives 9 figures on 4*ATN(1).
NOTE: The Cyber and the TI provide numerical round-off protection so that glaring errors within the precision of the software do not occur.
[NOTE: The IBM PCjr, with Cartridge BASIC, is capable of accuracy to 16 significant figures with variables defined as double precision.—Ed.]

I have three home computers: one at home for myself, one for my wife and son to play games with, and one at work. I chose the TI Home Computer because it is accurate enough to perform engineering and scientific computations without modification. These include operations as complex and diverse as matrix inversion, numerical differentiation and integration, satellite orbit computation, Monte Carlo simulation using random numbers, and filter and optimization analysis.

"Now we have home computers—and we expect them to do nearly everything that the large computers do."

I performed similar computations, but with smaller programs, on hand-held calculators such as the TI 59 or the HP 41. The TI 59, for example, was programmed for 3D satellite orbit computation including two orbit adjusts, using closed-form formulas. This program filled the storage on the TI 59. If you wanted to add any embellishments (e.g., a powered flight vehicle stage or more orbit adjustments), then you would need a computer's larger memory. Significantly increased storage is also desirable because there is a tendency for a program to grow to many times its original size as needs change with time.

The TI-99/4A fulfills this requirement for increased storage without sacrificing accuracy. It provides a minimum of 13 significant figures, and 10 of these are displayed in TI BASIC programming. All 13 figures can be displayed when Extended BASIC is used.

The Apple and the Commodore versions of BASIC provide only 9 to 10 significant figures and are not protected against round-off errors. For example, these computers give an incorrect answer of 1.65 when the following BASIC program is RUN:

```
10 PRINT INT ((1.655 + .005)*100)/100
```

The TRS 80, IBM PC, COMPAQ, Epson, and Atari computers use a BASIC supplied by Microsoft that provides about 7 significant figures. Some of these BASICs use double precision, which provides a 16-digit display. However, most of the mathematical functions are good to only 8 figures, even in the double precision mode. (It is considered undesirable for any computer to print or display more than one incorrect digit, if this can be avoided.) To see for yourself, run the following program on any of these computers:

```
10 PRINT 4*ATN(1)
```

Then compare the answer with 3.141592653589793. The Commodore and Apple print the answer out to 9 figures; the TI prints 10 and the IBM prints 7. You could obtain the 16-digit figure with the proper software accompaniment, and Apple users are fortunate in having a large selection of software alternatives available. In general,

Continued on p. 193

LASER-TARGETED TO THE RIGHT HOME COMPUTER MARKET...

Focused only on
Apple, Commodore,
IBM and Texas Instruments

5 MARKETING STRATEGIES FOR MAXIMUM REACH, IMPACT, AND RESPONSE

1 Pinpoint The High-Growth Segment Of The Home Market.
The home computer expansion of the 80's will be dominated by 3 brands: Apple, Commodore, and IBM. Advertisers who focus on this group will realize the largest gains.

2 Avoid Too Broad Or Too Narrow An Advertising Medium.
Indiscriminate advertising in both "general interest" and "machine-specific" publications is an inefficient strategy that quickly depletes your advertising budget.

3 Place Your Ad In The Strongest Editorial Environment.
Commercial messages stand out best when surrounded by facts—not fluff! Ads that keep company with the right balance of useful, stimulating articles will be noticed repeatedly.

4 Capture An Overlooked Universe Of Proven Computer Enthusiasts.
The over two million TI Home Computer owners represent an explosive buying potential. These proven enthusiasts will continue to shop for new products and many will migrate to larger, expandable home computer systems.

5 Simplify Your Media Buy With *Home Computer Magazine*.
No other magazine gives you more for your advertising dollar than *Home Computer Magazine*. It gives you instant access to the Apple, Commodore, IBM, and TI home markets without having to pay for readers you don't need or ad duplication you don't want. Our popular programs, colorful graphics, valuable reference articles, and in-depth product reviews provide readers with a magazine that is eagerly read, carefully saved, and constantly referred to. We invite you to examine a copy of *Home Computer Magazine*, and measure it against other print buys.

ONCE YOU COMPARE — THERE'S NO COMPARISON.™

HOME COMPUTER™ magazine

Contact our Ad Department: 1500 Valley River Drive, Suite 250 • Eugene, OR 97401 • Tel. (503) 341-1050

Microcomputer Accuracy . . . from p. 185

however, the appropriate software for the other machines does not exist, although it is possible to "fix" the IBM PC for accuracy by compiling the BASIC program using the BASIC compiler and running the compiled program from DOS. [On the IBM PCjr, Cartridge BASIC provides 16-digit accuracy.—Ed.]

Necessary Precision

Just how much accuracy is required of the home computer for scientific and engineering work? This is a difficult question because the answer depends on the type of computations that have to be done. Ten to fifteen years ago, I used a slide rule for all my personal computing. But I didn't use the slide rule to integrate rocket trajectories or invert 6-order matrices. These computations were saved for the IBM 704 or the Univac 1103. Then along came the HP 35, SR 52, HP 65, TI 59, HP 42, etc. These hand-held calculators changed the work habits of many scientists and engineers who came to expect the same accuracy from them as they got from the large IBM or Univac scientific machines. But these calculators were slow and limited in their capacity to hold large programs. Hence, they did not lend themselves to numerical integration or Monte Carlo simulation using random numbers. Now we have home computers—and we expect them to do nearly everything that the large computers do.

"Most computer users do not require any more accuracy than 6 or 7 significant figures."

Most scientific constants are known to 6 significant figures, and this sets the minimum standard for accuracy in computation. There is also some justification for extending this standard of precision to 8 digits since some physical measurements (e.g., angular velocity of the earth) are known to this accuracy. Computational errors continually arise from incorrectly entered data or faulty program algorithms. In addition to these obvious sources of error, two other conditions can lead to loss of accuracy in computing. These situations call for a standard of accuracy higher than 6 or 8 significant figures.

"Accurate" Numbers

The first problem results from the finite size of the computer's *accumulator* (a part of the Central Processing Unit where numbers are processed). The accumulator is only 8 decimal figures or 5 bytes (40 binary bits). Accuracy is lost whenever a large accurate number is added to (or subtracted from) a small accurate number in an accumulator. (An accurate number uses all the computer's significant digits.) Accuracy is also lost when two accurate numbers that are nearly equal are subtracted. As an example, try the following computation on your calculator or home computer:

$$(1E8 + 1.23456789) - 1E8 = ?$$

IBM's response is 1.2345678, TI's is 1.2346, and Apple and Commodore return 1.25. If the result you get is 1.23456, then your computer's addition/subtraction is very accurate. If you get zero, you should be aware that you must program your calculations to minimize this loss in accuracy. In some cases, it is possible to evaluate the above result using a power series expansion that saves several terms which can be accurately evaluated. Fortunately, this problem arises only during addition and subtraction and is of little concern when the input numbers are of the same magnitude and precision.

Byte Noise

Extended "number crunching" can also lead to a loss of accuracy in computing. Errors in the internal math functions (SQR, SIN, EXP, LOG, ATN, etc.) and the accumulation of inaccurate numbers lead to computation "noise." In estimating the effect of noise on computation accuracy we shall assume that the two noise sources are approximately equal. Then the computation error may be regarded as a noise error, which increases as the square root of the number of operations, N, and:

$$E_f = N \times E_I$$

where E_f and E_I denote final and initial errors.

You can get a feeling for the number of computer operations involved by considering a typical ten-minute powered flight trajectory integration. This involves about 100 computations for the integration step of 6 variables and an additional 100 computations for printout. If the step size is 2 sec, then approximately 60,000 operations or computations (+, −, ×, ÷, SQR, SIN, ATN, etc.) are required.

Now, if the computer has 8 significant figures and the interval math functions are evaluated to this accuracy, then the likely initial noise error is:

$$E_I = 10^{-8}$$

If we now run the trajectory integration program of 60,000 operations, then the error in the result E_f is estimated as:

$$E_f \approx 2.4 \times 10^{-6}$$

This computation assumes that all numbers are about the same size so that we don't have the loss in precision of the first type discussed.

The TI-99/4A can perform 1 million computations (operations) and still provide 10-digit accuracy. Many of the available computers will provide only 6-digit accuracy after 100 computations. But even with the TI-99/4A, it is occasionally necessary to program within the limits of the central processor. For example, evaluate:

$$B(n, K) = \left(\frac{1}{2^n}\right)\left(\frac{n!}{(n-K)!K!}\right)$$

for $n = 300$, $K = 200$

The term n! is pronounced "n factorial," and it is equal to:

$$1 \times 2 \times 3 \times \ldots \times (n-1) \times (n).$$

A TI computer is able to evaluate n! only up to n=69. IBM, Commodore and Apple can only go to n=33.

"Don't be so awed by your machine that you accept its every response with unquestioning trust simply because it is a computer."

Most computer users do not require any more accuracy than 6 or 7 significant figures. This is sufficient for most business, real estate, word processing or spreadsheet applications. Accountants may require 9 or 10 significant figures along with round-off protection because an error of even one cent is not acceptable. Serious engineering and scientific calculations clearly need more than 6 or 7 significant figures and round-off protection.

This discussion should motivate you to re-evaluate the degrees of accuracy you require and to investigate the capabilities of your own machines. Ideally, you should face these questions before you purchase your computer; otherwise you may find yourself constantly burdened with special programming techniques to avoid losses in accuracy. Of course other factors, such as sound capability, graphics features, and cost, must also be considered when selecting a computer. Just remember that all computers have limitations. Don't be so awed by your machine that you accept its every response with unquestioning trust simply because it is a computer. HCM

Future Is Now . . . from p. 167

The Future

As this hardware/software combination catches on in the marketplace, we can soon expect to see new software designed for Apples with the *SuperSprite* board in slot 7. These products will be state-of-the-art in every way and equal to or better than the graphics you see on any system.

One of the truly appealing features of *SuperSprite* is that the old Apple graphics do not become totally obsolete. They are compatible with the improved system. This new development in Apple graphics has accomplished a quantum leap to enhance an already fine system. Finally, we have a way to update, rather than replace, our Apple graphics and end up with arcade-quality results, with very natural-sounding speech thrown into the bargain. HCM

Don't let price get in the way of owning a quality printer.

Adding a printer to your computer makes sense. But deciding which printer to add can be tricky. Do you settle for a printer with limited functions and an inexpensive price tag or buy a more versatile printer that costs more than your computer? Neither choice makes sense.

Here's a refreshing option—the new, compact STX-80 printer from Star Micronics. It's the under $200 printer that's whisper-quiet, prints 60 cps and is ready to run with most popular personal computers.

The STX-80 has deluxe features you would expect in higher priced models. It prints a full 80 columns of crisp, attractive characters with true descenders, foreign language characters and special symbols. It offers both finely detailed dot-addressable graphics and block graphics.

And, of course, the STX-80 comes with Star Micronics' 180 day warranty (90 days on the print element).

The STX-80 thermal printer from Star Micronics. It combines high performance with a very low price. So now, there is nothing in the way of owning a quality printer.

*Manufacturer's suggested retail price.

star™
micronics·inc
THE POWER BEHIND THE PRINTED WORD.
Computer Peripherals Division
P.O. Box 612186, Dallas/Ft. Worth Airport, TX 75261 (214) 456-0052

Now with more computer connections.

The new STX-80 printer for only $199.*