PROOF OF PURCHASE

Software Giveaway with Purchase of this Magazine

See back cover

HOME COMPUTER magazine

FOCUSING EXCLUSIVELY ON • APPLE • ATARI • COMMODORE • IBM • TEXAS INSTRUMENTS

Vol. 5 No. 6

$3.[illegible] in USA
$4.50 in Canada

FREE SOUNDSHEET
Of Computer-Orchestrated
Synthesizers & Drum Machines

FlashMail

Computer-Help For ZIPier Letters

Featuring A Ready-To-Use Electronic Postman for:

- Database Construction
- Systematic Search & Sorts
- Personalized Mass Mailings

Special Apple, Atari, Commodore, IBM, & TI Software:

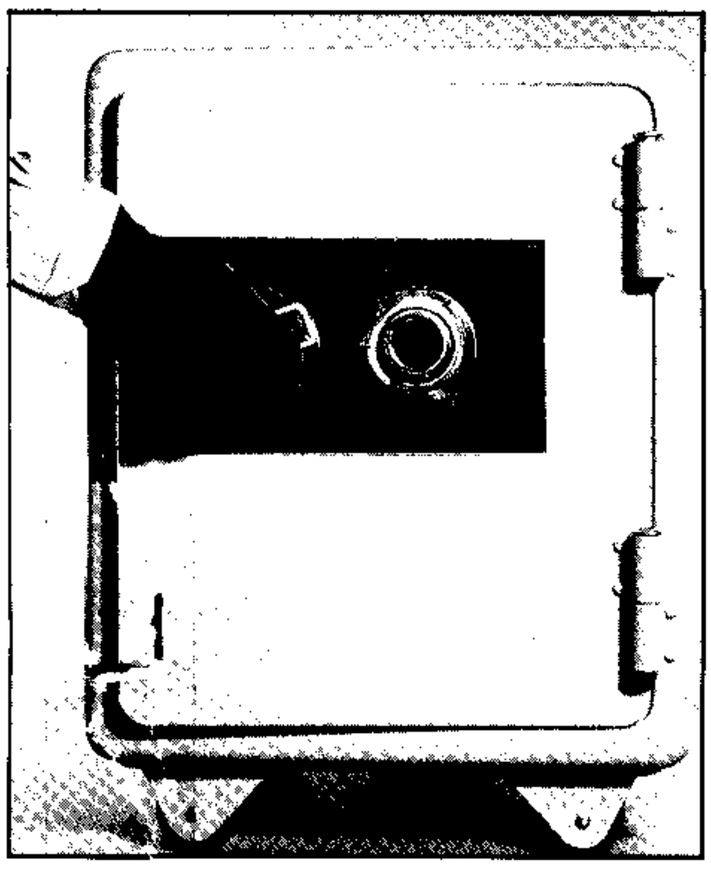
- Assembler For A Song
- Living Cell Bio-Simulation
- File-Shuffling Management Tricks
- Serf City Econo-Fun

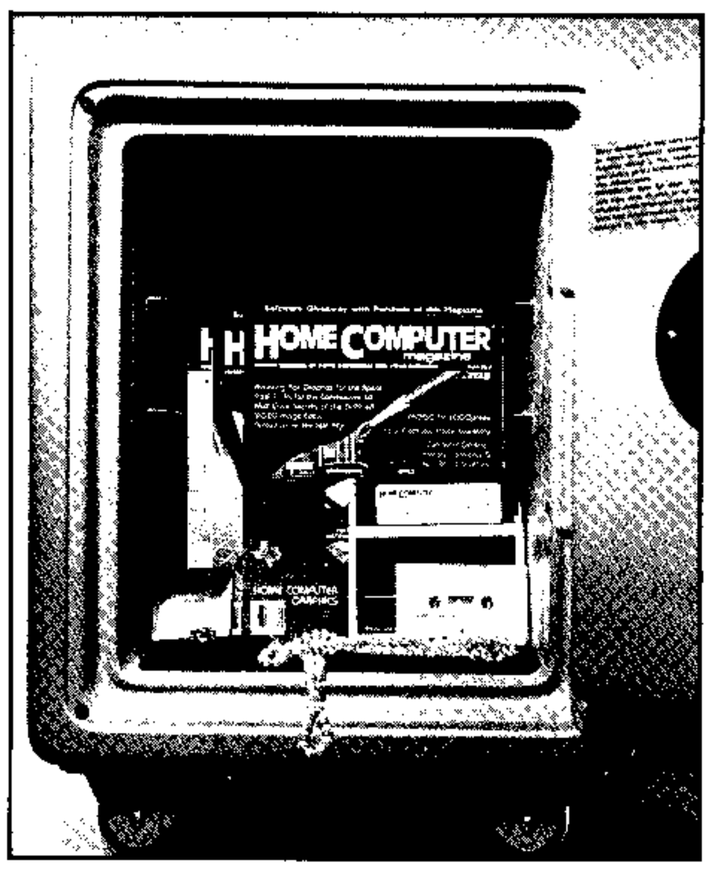
—Reviews Galore:

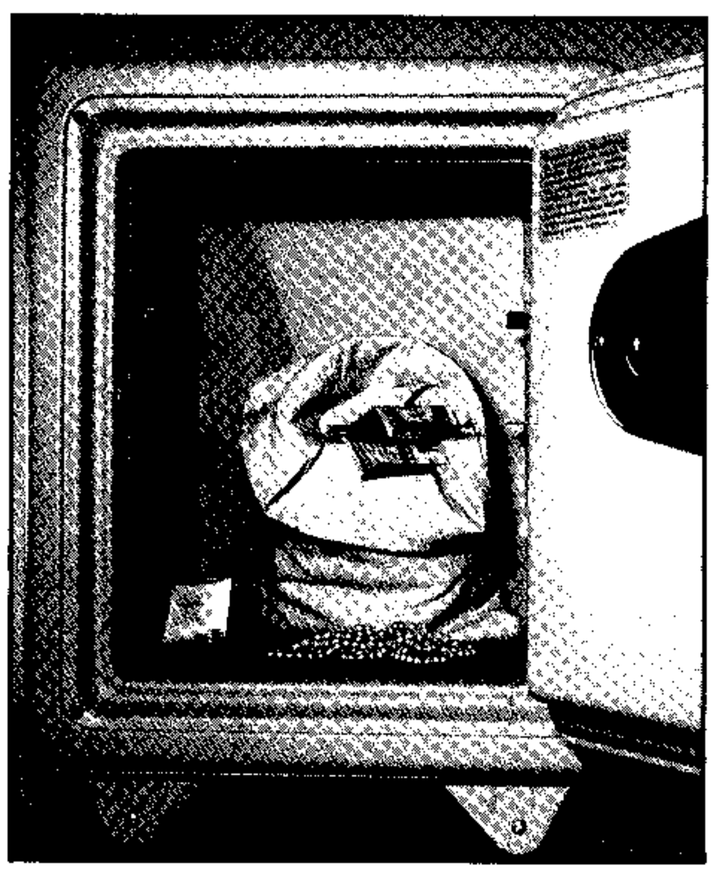
★ Financial Recipe Round-Up
★ Speed-Reading Illusions
★ Down-To-Earth & Blue-Sky Edu-Tools
★ Major Memory Expanders

Contains 52 Type-In Programs!

0 12509 06
477465



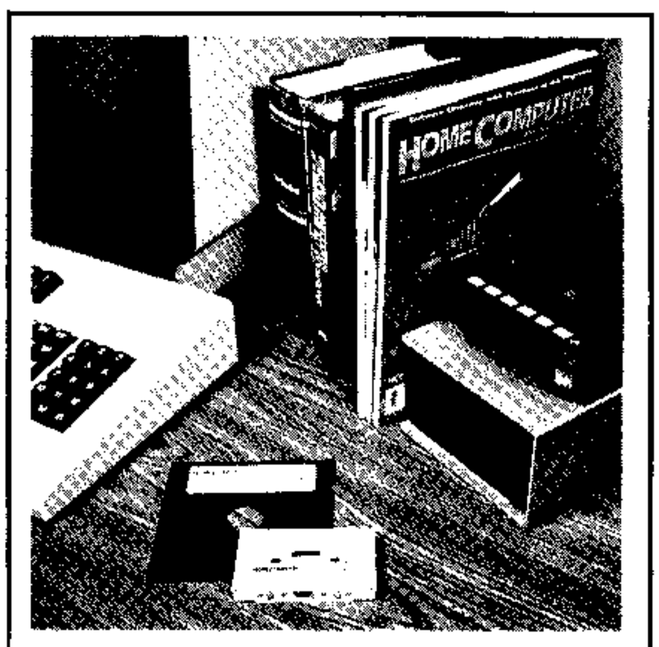




MISSING ANY VALUABLES?

If you're missing any back issues of HOME COMPUTER™ magazine you're missing more than you'll ever know . . .

Having each issue of *Home Computer Magazine* readily at hand provides you with direct access to a valuable reference library of home computer knowledge —unequaled anywhere!



A valuable reference library of each *Home Computer Magazine* issue is the One Essential Peripheral™ for your home computer.

Back Issues of HCM's program service —ON DISK™ or ON TAPE™ are also available.

Collect all programs from each magazine issue on a ready-to-RUN quality floppy disk or cassette tape available in separate versions for Apple, Atari, Commodore, IBM, and Texas Instruments.

ON DISK™ and ON TAPE™ are the convenient, accurate and affordable ways to save hundreds of typing hours.

SPECIAL CLOSE-OUT SALE!

ALL 4 BIG ISSUES SHOWN ON THE OPPOSITE PAGE
(Vol. 4 No. 1, Vol. 4 No. 2, Vol. 4 No. 3, Vol. 4 No. 4)
DELIVERED TO YOUR DOOR FOR ONLY $5.
USE REGULAR BOUND-IN ORDER FORM TO WRITE-IN SPECIAL SALE ORDER.
No phone orders less than $10 accepted. Limited-Time Offer—While Supplies Last.

"Safeguard" Your Home Computer Knowledge— Order Valuable Back Issues Today!

To Order, Use Bind-In Card at Center of Magazine.

HCM BACK ISSUES

. . . and discover what's in store for you— an extraordinary resource value!

VERSIONS SUPPORTED:

Machine	Media
APPLE II Family (A)	ON DISK™
Atari (At) (coverage commenced with issue 5.5)	ON DISK™/ON TAPE™
COMMODORE 64 (C)	ON DISK™/ON TAPE™
IBM PC/PCjr (I)	ON DISK™
TI-99/4A (T)	ON DISK™/ON TAPE™

*** = No ON TAPE™ available, even if normally supported**
TX = Extended BASIC programs only
PCjr = Available for PCjr only

Apple owners: Please note that ON DISK™ Media for *HCM* 4.1-4.3 is in DOS 3.3 format only, and all Apple programs beginning with *HCM* 4.4 are in ProDOS format. All programs will RUN on a 64K Apple II + (with Applesoft BASIC in ROM), an Apple *IIe*, or an Apple *IIc*.

Apple & IBM "clone" owners: Some *HCM* programs may not RUN (without modification) on your machines, because of differences in hardware and/or BASIC interpreters.

Issue 4.1:

Premier Issue * Uncle Larry's Fiddle Tunes * Electronic Sheet Music * Music In Mini Memory * PCjr: A Look Inside the Peanut's Shell * 66 Keys to Graphics Success: A Primer for the Commodore 64 * Have No Fear: Assembly Language Won't Byte, Part 3 * Porsches and other Pipedreams: Computer Assisted Savings * 3Dlle: Apple Graphics in Three Dimensions, Part 1 * Biting into Your Apple * Don't Be A SlowPOKE * Down Memory Lane: Don't let programmable characters gobble up your memory * Easy As Pie: Apple programming for intricate works of art * Microcomputer Accuracy * What is LOGO? * Lyrical LOGO * LOGO Shoots for the Moon: A lesson in structured problem-solving * Product Reviews * Flak Attack * Slots * Meltdown * Challenging the Tower of Hanoi * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A * Product News * Group Grapevine, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Flak Attack (A,C,I,T)
Air-to-ground battle game
Applesoft 3D (A)
Apple graphics in three dimensions
Tower of Hanoi (A,C,I,T)
An ancient brain teaser
Saving (A,C,I)
Computer-assisted savings plan
LOGO Poet (A,C*, I)
Recursion frees the poet in your console
LOGO Apollo (C*, T)
A lesson in structured problem-solving
Slots (T)
An intriguing Las Vegas simulation
Uncle Larry's Fiddle Tunes (C, I, T)
Play ten beloved fiddle tunes
Music Magic (TX)
"Joy to the World" in harmonious BASIC
Music Assembler (T)
Assembly language simplifies composition
Autosprite (C)
Routines to keep your graphics lively
Meltdown (TX)
Debug the reactor and save the world

Issue 4.2:

Graphics * Sea of States * San Francisco Tourist * Building Your Character: A Graphics Editor for the VIC-20 * Quick Pixel Tricks: A Graphics Editor for the C-64 * Follow the Bouncing Ball: On the rebound with graphics fundamentals * 3Dlle: Apple Graphics in Three Dimensions, Part 2 * Double Your Color, Double Your Fun: Sprites try on a layered look * Musical Mystery Words * Matrix Muncher * Elementary Addition and Subtraction for the VIC-20 * IBM Animation: Controlling the pallet on the PCjr * Jr. Sounds Off: Access Jr's Special Sound Enhancements * The Electronic Home Secretary * Files in LOGO * LOGO Spans the Generation Gap: A review of Commodore LOGO * FROGO: LOGO Invades the Arcade * Product Reviews * Tablut * Cannibals * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A * Product News * Group Grapevine, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Cannibals (A, C, I, T)
Livingston Stew, I presume?
FROGO (T)
A logical LOGO learning lesson
The Home Secretary (A, C, I, T)
Address & inventory recordkeeping
LOGOFILES (A, C*, I, T)
Access your DATA files in LOGO
Sea of States (C, TX)
State Capitals and dive for booty
Tablut (C, I, TX)
14th-century strategy revisited
Matrix Muncher (C)
Solve unknowns simultaneously
Graphic Editor (C)
Pixel tricks create easeful graphics
Mystery Words (A, I)
Reading between the treble clefs
PCjr Animation (PCjr)
Exploring Junior's graphic modes
Applesoft 3-D lle (A)
Edit your 3-D graphic shapes

Issue 4.3:

Productivity * Snap-Calc: A Homespun ready-to-use spreadsheet * Bars and Plots: Create colorful graphic charts of your records * Elementary Addition and Subtraction for the 99/4A and C-64: A powerful children's learning tool * Spider Graphics: Spin a colorful web on screen * Convertible for Comfort: Automatically convert your machine-language programs to DATA statements * Programming: The Name of the Game: Designing your own game—a complete tutorial * Colorfun on your VIC-20 * Product Reviews * Binary Forest: Branching out with LOGO * LOGO Flakes: Creative explorations with snowflake designs * Robochase * Cyber-Cipher * Wild Kingdom * Speeder * Boolean Brain * Missile Math * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A * Product News * Group Grapevine, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Snap-Calc (A, C, I, TX)
Home sweet spreadsheet
Robochase (A, C, I)
Run from the rampaging robots
Spider Graphics (A, I)
Spin a myriad of rainbow filaments
Boolean Brain (A, I)
A graphic Adventure inside computers
Wild Kingdom (A, C, I, TX)
Flee ferocious felines
Missile Math (A, C, I, T)
Launching grade-school arithmetic
Binary Forest (A, C*, I)
Branching-out with leafy LOGO trees
Bars & Plots (T)
Color your chart factfully
Cyber-Cipher (T)
Decode correct color combinations
Elem. Addition & Subtraction (C, T)
BASIC preschool arithmetic skill-builder
LOGO Flakes (T)
Snowflakes in June? This must be LOGO
Convertible for Comfort (C)
Machine Language DATA auto-conversion

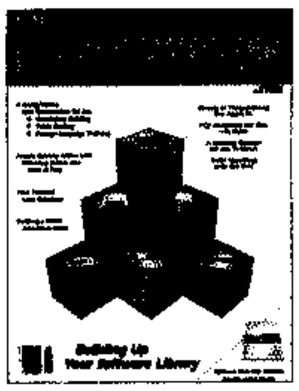
Issue 4.4:

Computer Sports * IIc: The Core of a New Machine * On the Home Court: Computer Sports Simulation * Razzle Dazzle: Quick Graphics Magic for the 99/4A * Simon Sez: Plug in 114 new BASIC commands to the Commodore 64 * Tax Deduction Filer: A complete tax recordkeeping program convinces you that makes tracking of deductions a breeze * Kaleido Computer: Creating a myriad of mosaic designs on your home computer * Multiplan Medium, Part 8 * Have No Fear: Assembly Language Won't Byte, Part 4 * The RS-232 Interface: Understanding Your Link to the Periphery * One for the Money, Two for the Slow—Adding a Second Drive to the PCjr * Missionary Impossible: A Logic Puzzle in LOGO pits you against hungry Cannibals * Product Reviews * Boolean Brain * Stadium Jumping * Market Madness * Elementary Addition and Subtraction: An arithmetic tutor (for Apple and IBM PC and PCjr systems) * HCM TECH NOTES: Apple, C-64, IBM and 99/4A * Product News * Group Grapevine, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Boolean Brain (C, TX)
A graphic Adventure inside computers
Tax Deduction Filer (A, C, I, TX)
SAVE-ing with your tax deductions
Market Madness (A, C, I, TX)
Exciting Stock market simulation
Stadium Jumping (A, C, I, T)
Horsing around an Olympic Stadium
LOGO Spreadsheet (A, C*, I, T)
And you thought LOGO was kidstuff
Missionary Impossible (A, C*, I, T)
Watch out for Cannibals with LOGO
Elem. Addition & Subtraction (A, I)
BASIC preschool arithmetic skill-builder

And There's More On The Next Page



Issue 4.5:

Building Up Your Software Library * Quiz Construction Set: Create a Quiz or Take a Quiz—a must for students and teachers * Personal Loan Calculator: Find out where your interest lies * Jumping Ahead With Game Programming: A complete game programming tutorial includes a program example * Sketch-64: Joystick graphics with just a flick of the wrist * Simon Sez: New string-related commands explained * Razzle Dazzle: Character manipulation on the 99/4A * Division Tutor: Teaching BASIC math learning skills * Putting The Puzzle All Together: Apple IIc Programming Considerations * Bird Brain * Slither * LOGO Clones: TI Graphics In a Turtle-Shell * Build A LOGO Adventure, Part 1 * Product Reviews * HCM One Liners * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A * Product News * Group Grapevine, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Bird Brain (A, C, I, T)
Keep your fishing feathers dry
Division Tutor (A, C, I, TX)
Expand elementary math skills
Personal Loan Calc (A, C, I, T,)
Find out where your interest lies
Sketch-64 (C)
Use a joystick to draw graphics
Quiz Construction Set:
Quiz-Make/Quiz-Take (A, C, I, T)
Complete tutorial with file examples
Peg Jump (A, C, I, T)
Learn BASIC game programming
Slither (A, C, I, T)
A maze of snake-like proportions
LOGO Clones (T)
TI-Graphics in a Turtle-Shell
LOGO Adventure (A, I, C*)
Pt. 1: Creating interactive fiction

Issue 5.1:

Thought Processing: A New Frontier In Home Computing * The Organizer: Store and organize your thoughts * Orbital Defender * Quiz-Print/Quiz-Print Tutorial: This educational enhancement is a tool for use with your Quiz Construction Set (see *HCM* 4.5) * Electronic Backgammon: A modern version of an ancient game of skill * Razzle Dazzle: Screen patterns with graphics characters on the 99/4A * Kors-Elf: An Arcade Typing-Tutor Game * Personal Loan Calculator: Find out where your interest lies * Apple Seedlings: A ProDOS Date-Setting Utility * IBMpressions: Create a beautiful pie chart * Build A LOGO Adventure, Part 2 * LOGO Sailing: A Premier Yachting Event * Simon Sez: Composing music is simple * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A * Product News * Group Grapevine, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Orbital Defender (A, C, I, T)
Split-second battle decisions
Electronic Backgammon (A, C, I, TX)
Pit your pips against the computer
Kors-Elf (A, C, I, TX)
An arcade typing adventure
The Organizer (A, C*, I, TX*)
A versatile Thought Processor
Quiz-Print (A, C, I, T)
Format printouts of your quizzes
Apple Seedling (A)
BASIC utility dates ProDOS files
LOGO Adventure (A, C*, I)
Pt. 2: Creating interactive fiction
Merging Files (C)
Experienced hackers only!
Personal Loan Calc (T)
Find out where your interest lies
Razzle Dazzle (T)
Wormwood your character graphics
LOGO Sailing (T)
Turtles race for the America's Cup
IBMpressions (I)
Create a beautiful pie chart

Issue 5.2:

Number Crunching: The Building Blocks of All Computing * It Figures: An equation calculator that'll crunch your numbers accurately * Evacu-Pod: See if you can rescue all the miners in this challenging space game * Switch 'n' Spell: Electronic anagram brain teasers to puzzle over (for children, and adults) * Laserithmetic: Strut your math skill with this space fantasy edu-game * Organizer Reports: An enchancement to print-out your organized thoughts (see The Organizer *HCM* 5.1) * Razzle Dazzle: Tinker with musical sounds, or Play it Maestro! * What is CP/M?: Learn the Basics of Control Programming for Microcomputers * Apple Seedlings: Sorting out your ProDOS Catalog * Commodore Hornblower: Discover what's inside the Commodore 64's SID chip * IBMpressions: Create 3-D surface drawings in BASIC * Field & Screen: A tutorial for using a Data Base System—correctly * Product Reviews * HCM One Liners * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A * Product News, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Evacu-Pod (A, C, I, TX)
Miner rescue in space
It Figures! (A, C, I, TX)
A mighty equation calculator
Laserithmetic (A, C, I, T)
Blast aliens with your math skills
Organizer Reports (A, C*, I, TX*)
Print your organized outlines
Switch 'n' Spell (A, C, I, T)
A spelling aid that's fun to boot
Apple Seedlings (A)
Sort your ProDOS catalogs
Commodore Hornblower (C)
Inside the SID chip
IBMpressions (I)
3-D surface drawing in BASIC

Issue 5.3

Computerized Budgeting: Featuring a ready-to-use budget processor (Budgetron) * Honing your Geometry skills (Geometrix) * LOGO Adventuring (Build A LOGO Adventure, Pt. 3) * Survive a nuclear plant disaster (Over-Reaction) * Guard the seaways with nuclear submarines (Torpedo Alley) * Turtles race with Zeno's theory (Achilles and the Turtle) * Apple Seedlings: Character graphics on the hi-res screen * Commodore Hornblower: Select waveforms and envelopes from SID * Razzle Dazzle: Multi-layered animation with TI sprites * IBMpressions: Blending sign waves into complex patterns * MAC-ROs: Expanding BASIC on Macintosh * Speeding Up a BASIC Program * Product Reviews * HCM One Liners * Group Grapevine * Product News, * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Budgetron (A, C, I, T)
Budget your income and expenses
Geometrix (A, C, I, T)
Sharpen your geometry skills
Over-Reaction (A, C, I, T)
You're at a nuclear plant's controls
Torpedo Alley (A, C, I, T)
Keep the enemy's ships at bay
Achilles & the Turtle (T)
A LOGO demonstration of Zeno's Theory
LOGO Adventure, Pt. 3 (A, C*, I)
Apple Seedlings (A)
Character graphics in hi-res
Commodore Hornblower (C)
Waveforms & envelopes from SID
Apple Tech Note (A)
Key-in checking routine
IBM Tech Note (I)
Selective keyboard input
Commodore Tech Note (C)
Merging programs from disk
TI Tech Note (T)
A full-screen editor

Issue 5.4

Time Management: Computer-Assisted Efficiency Comes Home * Run-Day-View: Let your computer streamline your day * Trig-Trix: Use the triangle as a measuring tool * Archeodroid: Participate in a future archeological dig * Mine Over Matter: Hone your business skills in this simulation of uranium mining operations * MAC-ROs: Create custom graphic shapes on Macintosh * IBMpressions: Create your own computer windows * Razzle Dazzle: Explore sound-on-sound recording with this 3-track recorder program * Apple Seedlings: Use a pie chart as a visual aid * Commodore Hornblower: Change filters on the SID chip * Algorithm-A-Tricks: Create invisible ripples * Speeding up a BASIC Program: Part 2 * Build a Logo Adventure: Part 4 * Product Reviews * HCM One Liners * Home Computer Industry Journal * Product News * HCM TECH NOTES: Apple, C-64, IBM, and 99/4A, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

Run-Day-View (A, C, I, T)
Organize your daily schedule
Trig-Trix (A, C, I, T)
The triangle makes measurement easy
Archeodroid (A, C, I, T)
Join a future archeological dig
Mine Over Matter (A, C, I, T)
Manage a uranium mine
LOGO Adventure, Pt. 4 (A, C, I)
Apple Seedlings (A)
Creating pie charts
Commodore Hornblower (C)
Changing filters on the SID chip
Apple Tech Note (A)
Exiting error routines
Commodore Tech Note (C)
Merging programs from tape
IBM Tech Note (I)
Using special character graphics

NOTE:
Programs for the IBM PC/PCjr will run on the Tandy 1000 with modifications specified on page 130.

Issue 5.5

Memoranda Processing: Computer-Assisted Correspondence Made Easy * NanoProcessor: Explore your computer's language * Electronic Typewriter: Make your computer act like an electronic typewriter * TI Card-Trix: Deal out organization on 3x5 file cards * The Plains of Salisbury: Help King Arthur save Camelot * Vital Signs: Control cardiovascular functions with this simulation * Apple Seedlings: Blend frequencies with your Apple * Commodore Hornblower: Create special SID sound effects * IBMpressions: Explore 3-D shadow graphics * Atari Atrium: Make your own "big" sounds with this sound-on-sound recorder * Razzle Dazzle: Crunch numbers down to the 10th decimal * Algorithm-A-Tricks: Learn the mathematical secrets behind Vital Signs * Soundbytes: Merge music and electronics * Product Reviews * HCM One-Liners * Home Computer Industry Journal * Product News * HCM TECH NOTES: Apple, Atari, C-64, IBM, and 99/4A, and much, much more!

CONTENTS: ON TAPE™ & ON DISK™

NanoProcessor (A, R, C, I, T)
Learning your computer's language
Electronic Typewriter (A, R, C, I)
The computer typewriter
TI Card-Trix (T)
Organize with file cards
The Plains of Salisbury (A, R, C, I, T)
Battling for Camelot
Vital Signs (A, R, C, I, T)
A "hearty" lesson
Apple Seedlings (A)
Blending frequencies
Commodore Hornblower (C)
Sound effects on SID
IBMpressions (I)
Shadowing 3-D graphics
Atari Atrium (R)
Recording sound-on-sound
Razzle Dazzle (T)
Calculate quick and easy

Atari users please note that coverage in HCM didn't commence until this issue. Also commencing with this issue, programs for the IBM PC/PCjr run as is on the Tandy 1000.

FOR NEW READERS

The Plain & Simple Truth About HOME COMPUTER™ magazine

Chock Full of Valuable Software & How-To Articles Without Filler

Every issue is a software "horn of plenty" with dozens of type-in-and-RUN programs printed in an easy-to-read listings format. Our programs are also available on inexpensive disks or cassettes for those who prefer the convenience of ready-to-RUN software. Step-by-step tutorials round out each issue, providing the solid facts you need without fluff or filler. Thus, each issue functions as an excellent reference work, as well as a valuable software source.

No Outside Advertising

Freed from the pressures of servicing *advertisers*, we concentrate on serving our *readers*. Each issue provides uninterrupted editorial flow and graphic layouts for better comprehension—plus unbiased product reviews which focus on true strengths and weaknesses, wherever the chips may fall . . . And we don't have to worry about losing advertisers because of publishing software in the magazine that is "too good." Consequently, we can provide the best free software available anywhere.

Focused on the 5 Hot Home Brands

We are 5 system-specific magazines under one wrapper—not a sprawling, "general interest" publication which attempts to cover too wide a field, only to spread itself too thin. The other side of the coin to this focused approach is the knowledge you gain from being exposed to the many tips, ideas, and techniques we provide for 4 of the 5 systems you may not even have. You'll learn more about your Apple, Atari, Commodore, IBM, or Texas Instruments home computer from this one magazine than from a host of more limited sources.

A Balanced Mix For a Perfect Recipe

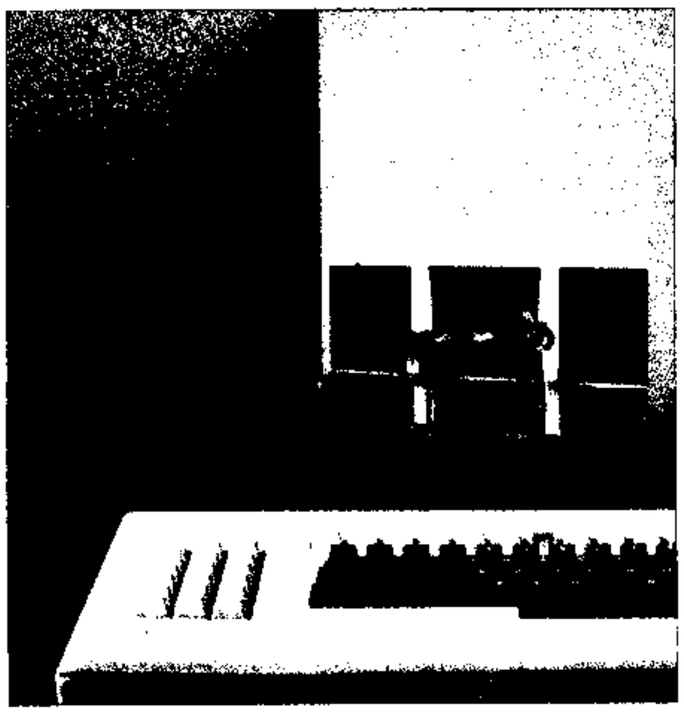
In each issue we strive for a perfect balance of productivity, entertainment, education, utilities, and computer literacy—serving the needs of novice and pro alike. Every issue is a full-course meal, with a smorgasboard of tasty dishes for all palates. Whereas other computer magazines may dish out lumps of "editorial indigestion," we serve up a satisfying blend—one digestible byte at a time.

—Welcome to Our World of Home Computing

Home Computer Magazine (ISSN 0747-055X) is published ten times per year by Emerald Valley Publishing Co., P.O. Box 70288, Eugene, OR 97401. 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 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 $3.50, and $4.50 in Canada. Foreign subscription payment should be in United States funds drawn on a U.S. bank. Second-class postage paid at Eugene, OR 97401, and Columbia, MO 65201.

POSTMASTER: Send all address changes to **Home Computer Magazine**, P. O. Box 70288, Eugene, OR 97401. 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 Editor or 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, programs, 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

Zappp! With the crack of a lightning bolt, the power of electronic mailing comes to the home computer. And to the home user, this new power may come as a pleasant surprise—like the pop-up greeting card depicted on our cover. Especially in the Holiday season, most of your correspondence may be as personal as this card. But when your mailing list grows, your computer can apply the speed of electricity to turn that lengthy list into a whole stack of personalized greetings, ready for the mailbox. Such useful software is the "hallmark" of *Home Computer Magazine*—so, look to us. We deliver the mail!

As electronic tools have accustomed us to quick—if not instant—transfer of information nearly anywhere on the globe, we have come to *expect* this magical ability as part of everyday life. Meanwhile computers have helped speed every level of communications—from the latest video transfer, right on down to one of the earliest forms of civilized correspondence: the mail.

In this issue, we present the *Electronic Postman* himself—a software enhancement to last issue's *Electronic Typewriter*. This modern hero will strike through all the tedium of personal "mass mailing" (just in time for the Holidays!) and still convey your personal touch.

To communicate your personal *power*, take charge of *Serf City*, an economic strategy game set in medieval times. This two-player simulation—pitting one feudal kingdom against another in a race for greater prosperity—will surely test your Machiavellian skills. (Remember: In *Serf City*, "hang ten" means to hang ten serfs!)

Communication is basic to life itself—as you can see in *Cell Mates*, a software simulation of a dynamic living cell. Here, you take the part of the nucleus, communicating life-preserving messages to our internal symbiotic partners: the organelles. As you learn to direct this cellular dance, its very complexity may surprise and awaken you to another great mystery of life.

To communicate a complex idea is one thing—to make it simple enough to understand is another. For, as you will be able to learn from this issue's *NanoAssembler* (a companion to *NanoProcessor* of last issue), simplicity underlies all complexity. Using a simple model of a computer's workings to simulate its complex *works*, this second *Nano* article shows how to use a human-like "assembly" language to communicate in terms the machine can understand.

After communicating our main *HCM* "software instructions," we dedicate specific program/tutorial columns to each computer brand. In this issue, *Apple Seedlings* creates three-dimensional waves on a simulated pond's surface; *Atari Atrium* breeds sine-wave hybrids; *Commodore Hornblower* addresses SID in BASID; *IBMpressions* grows a binary forest; *Razzle Dazzle* evokes some startling sound effects from the 99/4A. And, for all machines, *Problems In Productivity* brings one of our star programs from the past into the arena of everyday life.

In our review section, we try to communicate the real truth about specific products in the computer marketplace. How much hassle is it to add more memory? How speedy is this speed-reading course? Can your computer help you cook up simple financial advice? Can software programs teach you the secrets of the earth? Or will they help you reach for the stars?

Electronic communication may seem so commonplace to us that it takes something special to reveal the magic of electronic media—something as special as music. So as you enjoy our regular editorial fare, take time also to explore this issue's special inserts: the *Cornucopia Sound Sampler* and *Cornucopia Catalog* of electronic musical instruments. We think this is sound advice.

In the Electronic Age, the most powerful instrument—for any kind of communication—is your computer. And at *Home Computer Magazine*, we always aim to keep your home machines well in tune.

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

On Screen

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

"I've been discussing the part computers are starting to play in personal music creation and performance —now, I get to literally put my music where my mouth is."

Good news. We have a special audio-visual surprise for you. As you know, over the last couple of issues I've been discussing the part computers are starting to play in personal music creation and performance—now, I get to literally "put my music where my mouth is." As narrator of the special bound-in Cornucopia Sound Sampler, I'm proud to introduce the exciting new field of computer-orchestrated electronic music into your livingroom. And to enhance this educational experience, we've prepared an informative Cornucopia Catalog of musical product offerings. This 16-page supplement and our 6-minute stereo recording are the first steps in providing the crucial information that will help home computer users like yourselves to participate in this new electronic music making.*

For best results when playing this stereo recording, I suggest that you first place a regular LP record on your turntable platter, and then lay our Soundsheet on top of it. May I also suggest that you read this issue's *Soundbytes* column; it should help put the relationship between home computer usage and music participation into better perspective.

Once you've played our Soundsheet on the best stereo system you can muster up, please audition the recording for whomever you think may be interested. Don't forget to show them the catalog too.

Anyone entering the world of musical electronics will need a source of timely, accurate, and reliable information. *Music & Electronics*, our new sister publication, is that indispensable source. We're now in the midst of a major drive to gain subscribers to this unique magazine, and we need your help. Of course, we're also hoping that many more of our present *HCM* readers will choose to join us on this exciting new journey.

Producing and distributing a recording and catalog like this takes a great deal of money. We'd like to think that it's worth it. So, won't you please drop me a line telling how you like (or dislike) our efforts. If you'd like to see more of these special inserts, and have ideas for other state-of-the-art products you'd like to see carried, please let me know. Producing a publication like *Home Computer Magazine* without revenue from outside advertising isn't an easy task. If our readership does, in fact, support our Cornucopia product offerings, it makes it that much easier for us to increase the quality of this magazine's content.

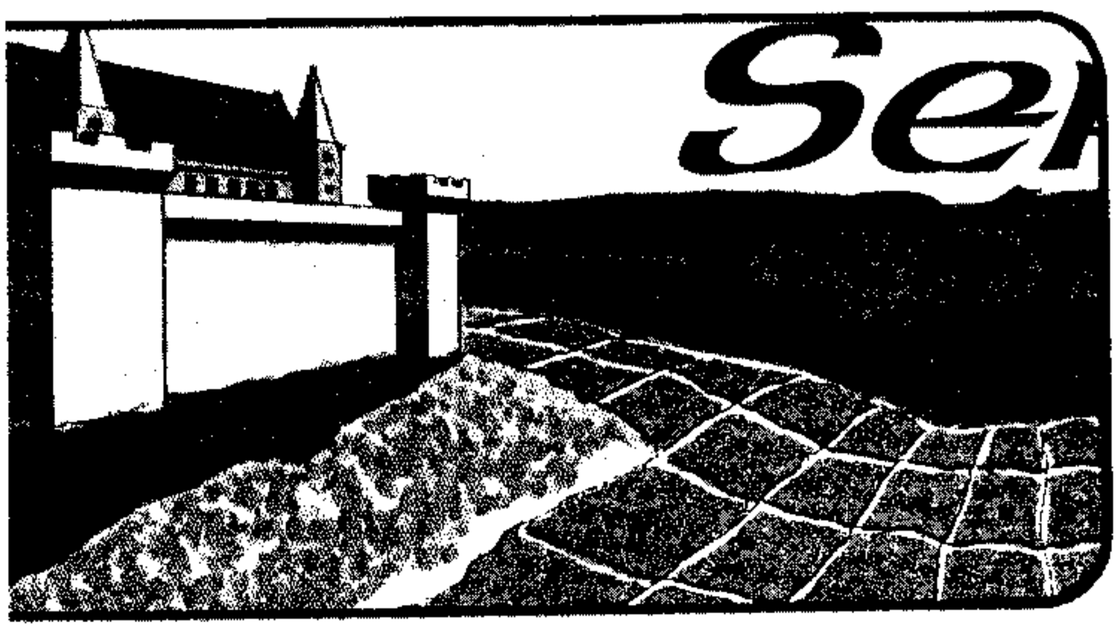
***NOTE TO HCM READERS OUTSIDE THE U.S.**
Because we are presently unable to ship product orders from outside of the continental U.S., the Cornucopia Catalog is not included in non-U.S. distribution. However, Canadian readers will receive the bound-in stereo recording to provide them with a "sound" reason for subscribing to the audio-visual version (magazine plus companion audio cassette) of Music & Electronics.

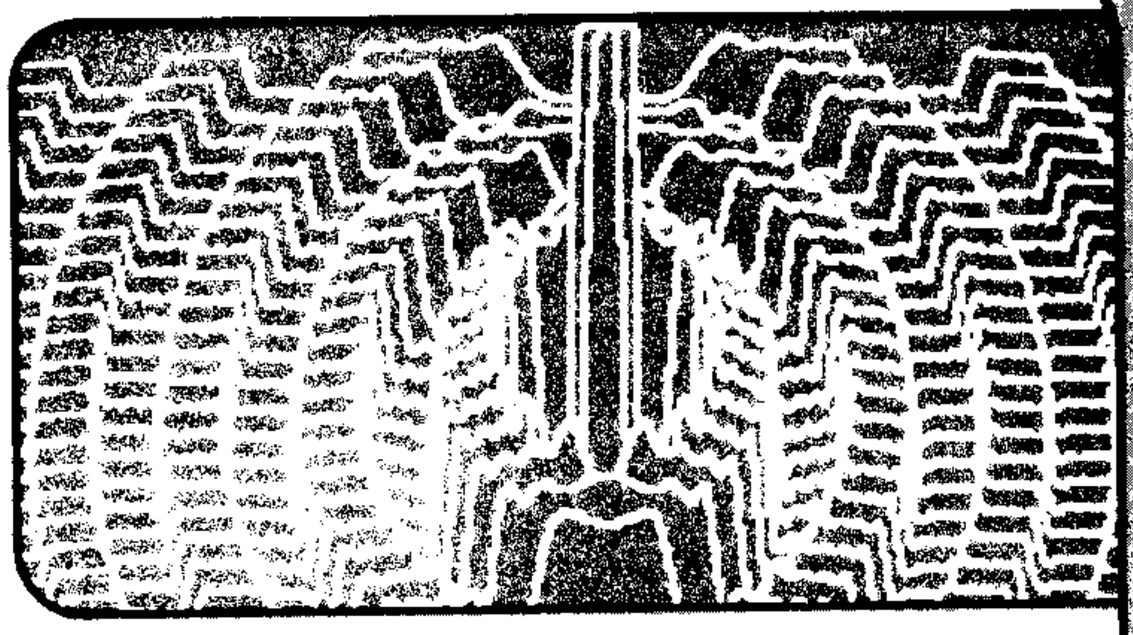
On another score (pun partly intended), this issue features some special enhancement software and the premiere of our *Problems In Productivity* column. I'm especially pleased to have had the opportunity to participate in the creation of such a unique, challenging—and badly needed—magazine feature. This new column is a direct result of the reader feedback I asked you for a few issues ago. This kind of exchange is what the reader-publisher relationship is—or should be—all about. And I, for one, would like to see a lot more of it. So please keep your feedback mail coming in. It's crucial in the process of "tuning" *HCM* to your changing needs.

You will notice that in this issue, two of our featured programs are "software enhancers"—separate utilities that add valuable functions to our previous stand-alone programs, turning them into even more powerful, versatile packages. *Electronic Postman* and *NanoAssembler* are important milestones in our continuing quest to bring you the best possible software within the constraints imposed by magazine-delivery economics. If you like this approach, be sure to let us know as soon as possible—quite a bit of lead-time is necessary to design and implement multi-issue software packages.

Although the reported marketing machinations surrounding Atari's new 16-bit 520 ST system continue to perplex us, we are nevertheless pleased over the preliminary response to last issue's expansion into the Atari 8-bit world—the 800, 800XL, and 130XE. It's evident from early feedback that *HCM* has a lot to offer the Atari user—valuable material not available elsewhere for so reasonable a cost. It's also obvious that, so far, we've only just scratched the tip of the iceberg in making the large Atari user base aware of *HCM*'s new coverage. I hope that many of you will take the initiative to help us get the word out. And while you're at it, please don't forget to tell all your friends with IBM-compatibles (such as Tandy, Zenith, etc.) and Apple-compatibles (Franklin) that *HCM* now also serves them too. Because we don't have the "big bucks" for expensive ad campaigns, our subscription marketing effort depends on your help. We're here today because of your past support. So, please continue to lend us a hand—come grow with us.

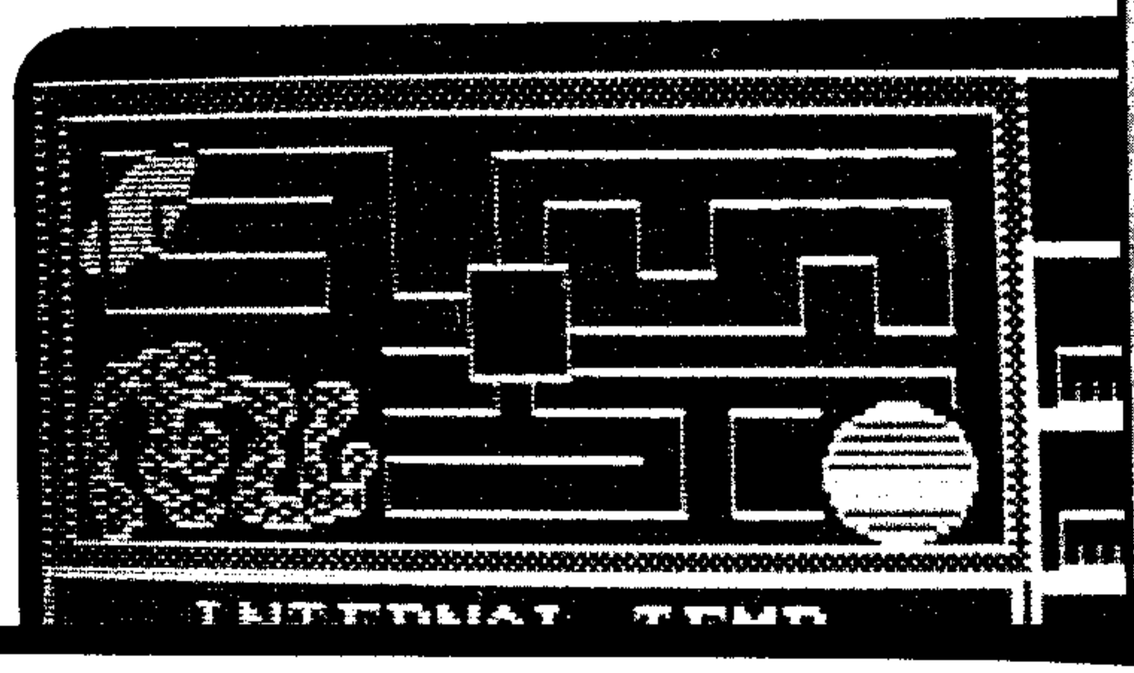
HOME COMPUTER magazine





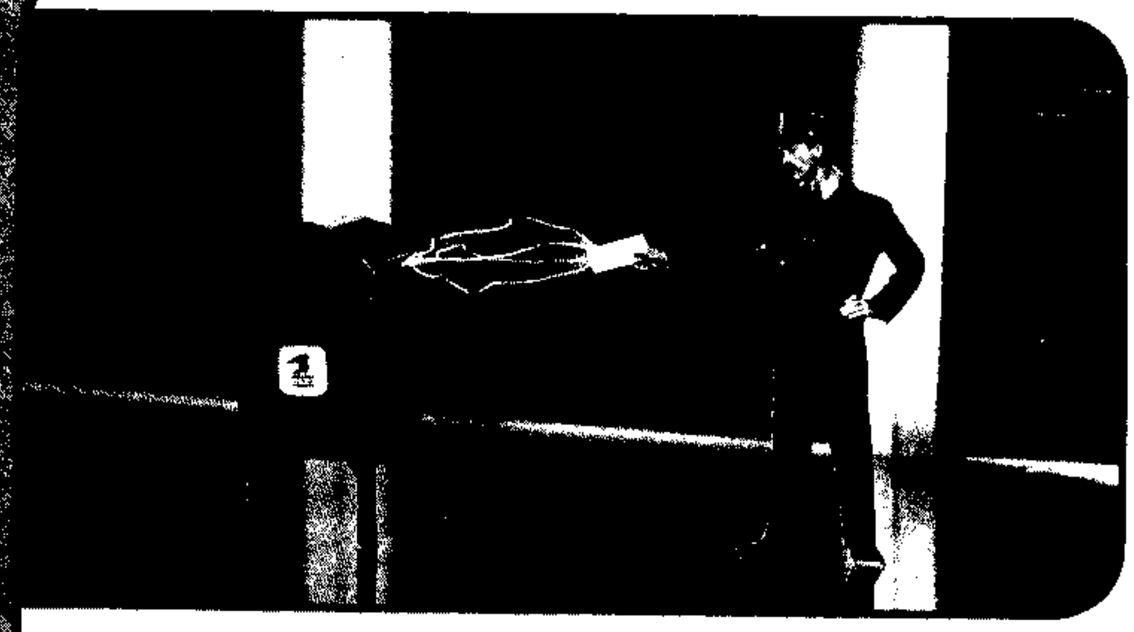
FEATURES

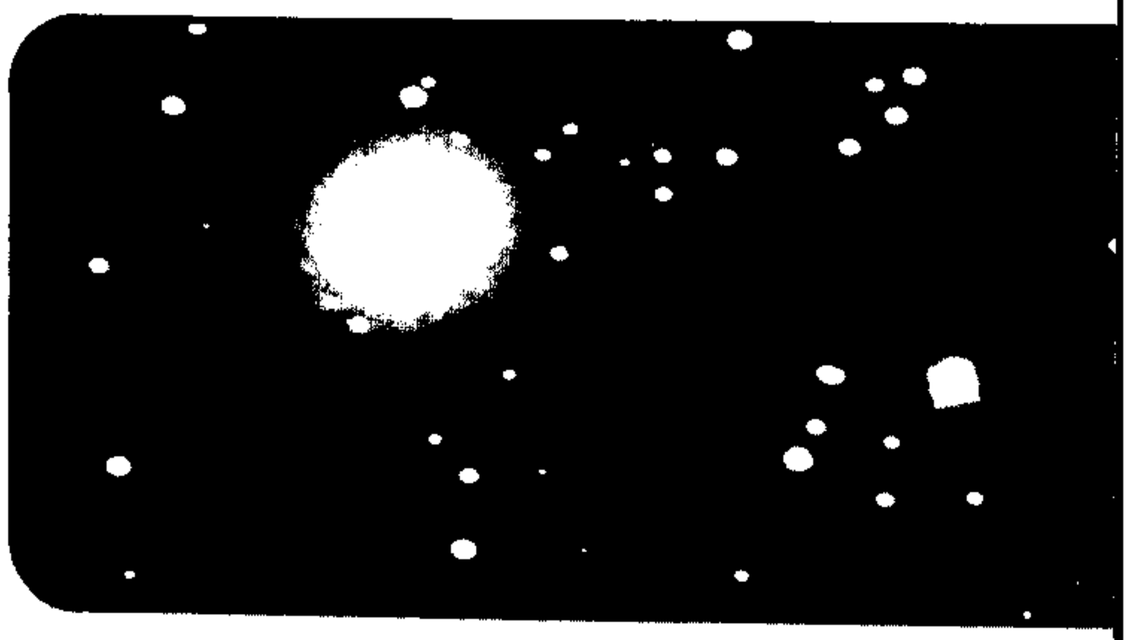
PRODUCT REVIEWS



CONTENTS

VOLUME 5 NUMBER 6





DEPARTMENTS

MACHINES SUPPORTED

Machine	Requirements	Media
APPLE II family	At least 64K RAM & Applesoft BASIC	ON DISK
FRANKLIN ACE	At least 64K RAM	ON DISK
ATARI 800, 800XL, 130XE	-None-	ON DISK/ON TAPE
COMMODORE 64	-None-	ON DISK/ON TAPE
COMMODORE 128	Must be in 64 mode	ON DISK/ON TAPE
IBM PC	BASICA	ON DISK
IBM PCjr	Cartridge BASIC	ON DISK
TANDY 1000	GW-BASIC Version 2.02	ON DISK
TI-99/4A	TI BASIC or Extended BASIC	ON DISK/ON TAPE

SPECIAL NOTES

Apple Owners: Apple ON DISK media for *HCM*, Vol. 4, No. 1-Vol. 4, No. 3 is in DOS 3.3 format only. Beginning with *HCM*, Vol. 4, No. 4, all Apple programs are in ProDOS format. All programs **RUN** on Apple II+, Apple *IIe*, or Apple *IIc* computers.

Franklin Owners: Beginning with *HCM*, Vol. 4, No. 4, all Apple ON DISK media is in ProDOS format only. Booting ProDOS on a Franklin requires the following steps:

1. Boot ProDOS. When the system hangs up, press [RESET].
2. Type 265B:EA EA and press [RETURN].
3. Type 2000G (insert no spaces between the last zero and the G) and press [RETURN].

See *HCM*, Vol. 5, No. 4, page 13 for more information.

Tandy 1000 Owners: Starting with *HCM*, Vol. 5, No. 5, all of our IBM PC programs run on the Tandy 1000 without modifications. Programs prior to Vol. 5, No. 5 may need minor changes as explained on page 130 of *HCM*, Vol. 5, No. 4.

Errata

Atari Computer Users: Due to an oversight in our software that translates program format to typeset listings, line 390 of the *NanoProcessor* in *HCM* Vol. 5, No. 5 mistakenly contained an English Pound symbol (£). This character should have been a backslash (\), which you produce by pressing [**SHIFT**] + on the Atari computer's keyboard.

Attention Subscribers:

To avoid missing any valuable issues, please check the address label for maturity of your subscription—so you can be sure to renew before it expires.

Letters to the Editor

Atari Saves A Reader
Dear Sir:

Today I received your magazine and much to my surprise, I found that you have added Atari to the micros covered. I must tell you that I was an old TI subscriber who had upgraded to the Atari and though I found you a source of good info and programs when I had the TI, I was going to let my subscription lapse—not now!

Enclosed please find a check for $5.95 for the Atari disk with the programs from Vol. 5, No. 5. If you can produce the great data, info, and programs that I know you have had in the past, not only will I renew, but I will talk to some of my friends and show them what you are all about.

Keep up the good work, and I and a million (or more) other Atari owners will flock to buy your magazine. I especially am fond of your tutorials and short programs which are usually both instructive and entertaining.

Alex Pignato
Oceanside, Long Island, NY

Alex, your letter is typical of the positive response we have received to our new Atari coverage. We are already beginning to see the vanguard of that flock you speak about. We want all the Atari users out there to know that the roost is ready at Home Computer Magazine. So, fly on over!

Better Sequential Saver
Dear Sir:

I am a proud owner of a Commodore 64 and enjoy HCM very much. In last month's issue, Mike Poole explained how he used the C-64 merge program to convert sequential files to a usable program. I think I have come across an easier way to make a sequential file a usable program. Just type LOAD''PROGRAM NAME,S'',8. The sequential file is loaded into memory and can be saved as a program file.

You can also save a program as a sequential file by putting (,S) at the end of the file name when saving a program. This serves as a form of protection from those who don't know about this trick. If you put (,U) at the end of a file name when saving a program, the program will be listed as a user file in the directory. The program can then only be loaded with the ,8,1 ending. I hope this information is useful to some HCM users because it has been very useful to me.

Bert Palmer
Address unknown

Well Bert, saving a program with the ,S extension does not actually change the format of the file—it is still a program file. This method does, however, place the SEQ identifier next to the file. Thus, even though the directory contains the SEQ, it is possible to load a program file saved in this manner because the file is still in program format. Your method does not work with program files that were listed to disk as sequential text files (see "HCM Bug-Out" for details on how this is done), and therefore not in tokenized format. The program must be in tokenized format for the LOAD statement to work properly. As for program protection, using the extensions ,S and ,U is a great little trick. Who would think that a "SEQuential file" was actually your latest programming masterpiece? And better yet, who would think to LOAD it? Thanks for the tip Bert.

Creative Cursoring On Apple II
Dear Sir:

Did you ever wish that you could change the Apple's cursor without any special routines? Well, I have. On an Apple, screen memory is larger than the actual screen. I guess it was easier to just make it an even 1-K. The leftover bytes are either unused or are used for special functions. Location 2043 (decimal) is the value of the cursor (its ASCII value + 128). If you POKE this location with the correct number, it changes the cursor to whatever you want. For example,

```
POKE 2043,ASC("_")+128
```

will produce a blinking underline cursor. This cursor remains until you POKE 2043 with a different number or press [CONTROL] [RESET]. If you put the actual ASCII value here, the character will be in a flash mode in addition to the normal cursor blink. Am I the only one who didn't know this, or is this an important find?

Donald W. Scott, Jr.
Address unknown

Donald, you've found something that we haven't seen documented before—but don't try to get it to work on an Apple without 80-column capability. The location you are POKEing is one that is part of text-screen scratch-pad RAM used by expansion slot 3 or the auxiliary slot on the Apple IIe. Apparently, this location is used by the 80-column card to determine the cursor character, even when the computer is not in 80-column mode, which explains the little trick you've uncovered.

Who Adds Most To PCjr?
Dear Sir:

As an owner of the PCjr, I have several questions that are in urgent need of reliable answers.

1. There are five sources (Tecmar, Quadram, Impulse, Racore, Legacy) of second-drive enhancements with memory expansion for the PCjr, but what are the special features of each? How does each of them increase the PCjr's compatibility?
2. Can the PCjr utilize the 8087 math-coprocessor? (For example, the Micro Way 8087 math-coprocessor for PCjr.)
3. Is there any other internal modem besides the one sold by IBM that the PCjr can use?
4. How is Racore's DMA controller board utilized by the PCjr? Does it make the PCjr more compatible?
5. Because of the PCjr's lack of DMA, intense I/O and downloading of large files are close to impossible. Would the DMA controller board by Racore eliminate such problems? Also would it allow the disk drive and the CPU to work simultaneously?

Wen J. Chen
Flushing, NY 11355

Over several issues, we have reviewed the Tecmar jrCaptain (Vol. 4, No. 4), Legacy II (Vol. 5, No. 2), the Quadram Expansion (Vol. 5, No. 4)—and, in this issue, both the Microsoft and IBM PCjr memory expansion systems. In this last article, you will find a chart comparing the features of all these products. We are currently examining the Racore version of the previously reviewed Quadram product and will publish a review in an upcoming issue. We are also looking at the Impulse product and the Racore DMA controller as future review items. In addition, our technical staff is researching the feasibility of installing an 8087 in the PCjr—look for an HCM report in an upcoming issue.

TI LOGO II Cartridge Sought
Dear Sir:

I would like to buy the TI LOGO II cartridge. I have been looking for a place to buy the cartridge, but cannot find one. Could you help me out? I would greatly appreciate it. Thank you.

Roman L. Schmitz
Malone, WI 53049

Roman, we've traced down a few sources for you. TI LOGO II is currently available from: TexComp (P.O. Box 33084, Granada Hills, CA 91344) for $20.00; Unisource (P.O. Box 64240, Lubbock, TX 79464) for $69.95; and Triton Products (P.O. Box 8123, San Francisco, CA 94128) for $19.95. Other local or regional sources may also carry this product.

Disenabling Break Key?
Dear Sir:

To disenable the break key on the IBM computer, this program works:

```
10 POKE 108,64:POKE 10,1:POKE 110,112:POKE 111,0
```

I tried this program on the Tandy 1000—it did not work. Another way to trap the break button on the IBM is this program:

```
10 KEY 17, CHR$(4)+CHR$(7):KEY (17) ON:ON KEY (17) GOSUB 100
20 REM REST OF PROGRAM HERE
90 REM
100 PRINT "BREAK BUTTON INOPERABLE":RETURN
```

This did not work on the Tandy 1000 either. I hope you can solve my problem. Thank you.

Darren Eichorn
Hatboro, PA 19040

Darren, the BREAK key on the IBM is function key 17, and is therefore redefinable as you have detailed. The Tandy machine does not allow redefining of the function keys above 12. Consequently, no similar fix that we know of will work on that machine. Are there any Tandy programmers out there who've been able to perform this feat of magic?

Program Prompts Purchase
Dear Sir:

I bought a computer simply because of your program "The Organizer." I had wanted "Think Tank," but knew it would only run on Apple or IBM PC machines and, in addition to its price (approximately $125), I felt the overall system costs of these two machines were above my means at this time; but when I saw the "The Organizer" would run on the Commodore 64, I then bought a second-hand 64 (and your software)! I have not yet tried it out, so my enthusiasm may be premature, but at least I've gotten my feet wet.

I would like more technical things and fewer games. Also separate loose-leaf documentation that we could order from the magazine, for ease of use and safety (cheap of course). I appreciate the documentation in the magazine, but the magazine will be hard to use everyday and I will have to make a photocopy.

I notice there seem to be quite a few programs exclusively for the Apple or IBMs. I hope Commodore doesn't discontinue the 64, I would like to see more 64 programs in your magazine.

The program listings in your magazine are helpful even if I don't key them in. If you print updates on the programs I use, I will have to be able to find the lines easily.

Thanks for innovation!

Lin Sinclair
Alameda, CA 94501

Thank you, Lin, for the compliments and advice. Actually, if you look more closely, you should find that our magazine content is evenly balanced between all the machines we cover. You can check this out yourself by counting through the Program Listings Contents on page 71. As for technical material, almost everything we publish contains a wealth of technical information—e.g., our Programmer's Windows for all major programs (including games), plus Tech Notes, tutorials, and reviews of specialized products. We try to balance this material against lighter content for less tech-oriented computer users. A word about the "games" we do publish: We have recently shifted our approach to computer gaming away from the arcade to simulations with some educational—as well as entertainment—value. Your suggestion of providing separate software documentation is interesting and deserves consideration. Anyone else out there want to see this? Good luck with your new computer, Lin, and have fun!

MousePaint: Look But Don't Print?
Dear Sir:

I think your magazine is marvelous. We just recently purchased an Apple *IIc* computer and have bought two of your Home Computer Magazines (Vol. 5, No. 2 and Vol. 5, No. 3).

My question to you has to do with my printer and my mouse. It seems I can't get the MousePaint pictures I drew to print on my Star Gemini 10x printer with the PAC interface. We've tried everything and hope you can shed some light on the subject. Will you please explain what might be wrong? Is the problem in my printer, interface, or MousePaint disk?

I was wondering if you were planning on putting more arcade-style programs in your magazine in the future. I also think you should arrange your magazine into sections, like putting all the Apple software in one section, C-64 software in another section, and so on. This would really help me and probably all the other subscribers a lot.

Roger Thompson
Reeds Spring, MO 65737

In order to send a MousePaint drawing to a printer, you need an interface and printer combination that will dump the high-resolution graphics screen. From what you say, it doesn't sound like your interface-printer combination can do this. We've had excellent success using the Grappler + from Orange Micro (1400 N. Lakeview Avenue, Anaheim, CA 92807) and an Epson MX-80. (See "Mousing Around on the Apple II" in HCM Vol. 4, No. 4 for details.) As to your question about separating the magazine into sections: If you look at the issues following Vol. 5, No. 3, you will see that we have indeed phased in a system of marking the magazine in easy-to-identify content sections. Special "edge-tab" markers flag each major section. Documentation for all major programs (those with versions for all 5 computer brands we cover) appears in the first section marked "Software Instructions." This is followed by a section of "Product Reviews," then "Tutorials" (including separate columns for each computer brand), the "Programmer's Windows," and finally the "Type-in Listings" section. We hope this will be a very handy quick-reference guide for you and others who have requested similar tab-indexing.

To DeBug, Or Not To DeBug . . .
Dear Sir:

As a new subscriber to HCM I wish to tell you how pleased I am with your magazine. I purchased the first issue I ever saw at a magazine stand and was surprised at the amount of information I was able to understand and use. I am a newcomer to computers and have a TI-99/4A computer with a TI Expansion system, 32K memory board, RS232 interface and Extended BASIC. I am still not sure of how to make the most of this system, but am enjoying every moment at the keyboard.

After going through Vol. 5, No. 3, I called your 800-number and ordered a one-year subscription, and backorders of Vol. 5, Nos. 1 and 2 ON DISK. They have just arrived and after reading instructions on how to update the disks (I am referring to The Organizer program) and a little trepidation, I began. Then to my utter surprise, I discovered that the updates had already been done to Vol. 5, No. 1 ON DISK. Did I somehow miss some information as to these updates being made automatically, and if so, can you explain the procedures for when and if updates are needed?

I realize that Texas Instruments is no longer making the TI-99/4A, but I am sure that there are many other new users who might appreciate some articles that are geared for beginner users. I am referring to some of the lesser-used functions such as PEEK, INIT, LINK, DEF, etc. When do you use them, and examples in short programs? How about a LOAD program to automatically boot your disks?

Thanks for a terrific magazine which is giving me hours of pleasure.

Sylvia Kesten
Northridge, CA 91326

Glad to have you on board, Sylvia. After checking the introduction to our DeBugs section, we can understand your confusion. Sure enough, we do not specifically communicate this fact: ON DISK issues always contain the latest debugged—and enhanced—versions of each program as they are shipped. In addition, the most recent ON DISK issue will also contain an "update file" of the latest DeBugs, for those who received any previous issue before all its DeBugs appeared in print. These files can be merged with programs on the older, non-updated disk. If you want to ascertain whether your disk contains the latest updated programs or whether it requires merging in update files, compare the program's version number on your ON DISK issue to the version number of the latest DeBug as it appears in print. If the numbers are the same, your version contains all the latest DeBugs; you do not have to merge it with an update file.

We have—and will continue to—run articles that demonstrate specific "PEEK and POKE" kinds of programming methods. For the 99/4A, check out "Razzle Dazzle," a regular tutorial column that beginners can learn from.

Putting A Second Drive Into Junior
Dear Sir:

As a loyal reader of HCM for several years, I am pleased to see that you are still keeping readers up-to-date on happenings with the PCjr. As you probably know, most other magazines concerning the PCjr discontinued publication when production came to a halt. All PCjr material which I have read in HCM is always very helpful and beneficial to me.

Now I need special help! Shortly after I purchased my PCjr, I bought an Okidata printer. Next, I purchased the Tecmar jrCaptain. With these new additions, I have been able to utilize my PCjr for something other than playing games. My problem is this: I would like to purchase a second disk drive. I already own a clock/calendar, a printer port, and have expanded memory to 640K. I have shopped around, looking at the Legacy and other systems, and am very disappointed because I don't want to pay hundreds of dollars for a second drive which adds features that I already own! All I want is a second disk drive—nothing else!

Please help me locate a second drive for my Junior, where I don't have to pay for unnecessary features.

Stephanie Fox
Brick, NJ 08723

At HCM we have several PCjr's with second disk drives, Stephanie—and they are completely compatible with the Tecmar jrCaptain. (For specifics on how a second drive and the Tecmar product work together, see the letter from Ron Sutherland in HCM Vol. 5, No., 3.) We installed these drives ourselves using the method described in the article entitled "One For The Money, Two For The Slow: Adding A Second Disk Drive To The PCjr" in HCM Vol. 4, No. 4. You may wish to get a copy of this issue (see HCM Back Issues in the front of this magazine), and see if this solution fits your needs. The kit described in the article allows you to modify your PCjr disk controller card to support a second drive, and is available from our order department for $49.95 plus $3.50 for shipping and handling.

Seeking C-64 Interface
Dear Sir:

I have read your fine magazine since the Vol. 4, No. 1 issue, especially for the Commodore 64 articles and the Letters to the Editor column. This question is directed to that column: What interface(s) will work best with the Commodore 64 and the Gemini-10 (not the new 10x)? One that would give complete Commodore 1525 emulation?

Letters to the Editor CONTINUED

After many unanswered letters from printer interface companies, I'm hoping the answer to this question shows up in your Letters to the Editor column. Thanking you in advance.

Richard L. Wightman
Wellsville, NY 14895

There are so many good interfaces, Richard, we don't know where to begin. Cardco (300 S. Topeka, Wichita, KS 67202) has several quality interfaces that make the Gemini-10 emulate a 1525. If all you do is word processing, the ?/B or ?/PS Cardco interfaces should suit you fine. If you want something with graphic capabilities, the ?/G+ from Cardco is an excellent choice. Both The Connection from Tymac Controls Corporation (127 Main Street, Franklin, NJ 07416) and Super Graphix from Xetec (3010 Arnaold Rd. Salina, KS 67401) provide 1525 emulation with graphics.

Making Room For Mac

Dear Sir:

I have noticed that your last two issues of Home Computer Magazine have had a section called "MacRos" for the Apple Macintosh. After seeing these articles I have two basic questions for you. First, are you going to continue to support the Macintosh? Will you broaden the coverage? I hope you will, especially since the Macintosh is a member of one of the families you are supporting. (When IBM introduced its PCjr, you immediately started to support it!)

The second question deals with the MS-BASIC for the Macintosh. If you start to support the Mac, will you provide programs for it? I know that the MS-BASIC version for the Mac is similar to the MS-BASIC for IBM. If so, it would seem to be easy enough, even without color. Please consider what I have suggested. I really am interested in buying a Mac and I would like your magazine to cover it in the same style which you have covered the other computers. I am sure that other readers share my view—so please consider it.

Brian Neidig
Taylor, TX 76578

Brian, we have run several features for the Mac in recent issues to see what kind of response we would get from home users. We have to say that the response—though positive—was sparse. Your letter is one of the very few we have received from Mac users. This is not to say that—given more requests—we won't cover the Macintosh in the future. But to include Mac versions of all our major programs, we would need to make space in the magazine by deleting something else or increasing our total page length—a costly alternative in either case. There's also another consideration, Brian. Apple just recently canceled its long-worked-on MacBASIC. Our plea to the Apple for bundling the languages with all Macs is now moot. At its present price in the $150 range, we feel that Microsoft's BASIC for the Mac is too costly a package for a large enough user base to develop. Without the necessary numbers, an expansion to full Mac coverage just doesn't make economic sense.

99/4A Lock-Up

Dear Sir:

I own a TI-99/4A and have been having problems with the computer locking up. I have been told that it is normal for it to happen once in awhile, but it locks up much more than it should. The cartridge that locks up the most is Extended BASIC. When I plug in the Extended BASIC cartridge and then select choice #2 for Extended BASIC the screen will sometimes fill up with random symbols.

It also locks up with other cartridges. For example, in the middle of a chess game with the computer, the computer will sometimes lock up. This problem appears to be either in the cartridge plug-in or where the peripheral expansion system plugs into the speech synthesizer or possibly where the speech synthesizer plugs into the keyboard.

I have written many letters to different companies and nobody seems to be able to figure out my problem.

Any help you could give me will be appreciated.

Kip A. Dondlinger
Hutchinson, KS 67502

Kip, you have just hit on two of the 99/4A's trouble spots. The cartridge slot and the side port are very susceptible to wear and tear. The side port, in particular, can give users many headaches. To help minimize your own headaches, make sure that the console and expansion box cable are on a flat, level surface. Problems start arising if the expansion cable doesn't fit just right into the side port. The side port can also be a problem if your console moves around on the table top. Any time the console moves, the connection between the console and the expansion cable can weaken. If this is a problem, try putting double-sided tape under the feet of the console to keep it from slipping. Finally, the cartridge port is quite often the culprit in a lock-up. Extensive plugging and unplugging of cartridges would cause even the most expensive connectors to wear out in a very short time. Unfortunately, it would require some technical expertise to disassemble the cartridges and/or your console to clean the contacts or, in severe cases, to replace the contacts. The best solution to this problem is to minimize the number of times that you switch cartridges.

Ask And Ye Shall . . .

Dear Sir:

I don't own an Atari 800XL computer anymore. I sold it because it didn't stand up to my TI. Why don't you cover the Atari computers? There are more Atari 800s and 800XLs than PCjrs, not to mention Atari 400s, 600XLs and some 1200XLs. If you added Atari's line of computers to your magazine you would really see a jump in your subscriptions and sales. If space is a problem why not just drop the PCjr? (I know that will make a lot of people angry and upset, so let me apologize now to any offended person.)

Also you have HCM one-liners. Why not HCM graphics? There are many people who would love to show off their still and moving graphics programs. I feel you should let your faithful readers know the answers to these questions.

Frank DeCandia
Jersey City, NJ 07307

There's no need to drop a very fine machine like the PCjr, Frank. We have kept Junior and also added more pages for Atari coverage. As you can see from Mr. Pignato's letter (Atari Saves A Reader) above, we are already starting to reap the benefits of this new coverage.

In regard to your second question, our "HCM One-Liners" page generally includes outstanding graphics submissions. In addition, our specialty articles – like TI "Razzle Dazzle" and "Atari Atrium"—make perfect showcases for such programs. If you have an original program that you'd like to show off, send it to us. We welcome any article and program submissions—graphics or otherwise.

For IBM PC Sans Graphics Board

Dear Sir:

I recently bought a copy of your magazine at a newsstand thinking I would enjoy running several of the programs offered. I found, however, that I couldn't use any of the programs because my IBM PC is not equipped with a graphics card. I bought the computer for data management (text) and word processing, so a graphics card is unnecessary (and would cost about $500). Wouldn't it be possible with a program like Budgetron to have an alternative version for machines without graphics? If not, then your publication is of no use to someone like me, which seems a pity.

Thank you for your attention.

Katherine A. Kirk
Victoria, BC, Canada

Katherine, the architecture of the IBM PC is unlike other computers such as the Apple. The IBM BASIC interpreter only supports graphics commands when the graphics adapter is installed. We feel that color is an essential part of good home-user software. Because both the IBM PC and PCjr are such great color-graphics machines, we don't want to limit our readers to monochrome-only programs. Without the graphics interface installed, our programs would be limited to text only. Whenever a program runs best with the card, we consider it a system requirement. Budgetron is an excellent example: it will run without the graphic adapter card, however the bar-graph option results in an ILLEGAL FUNCTION CALL. If you don't select this option, the program will work just fine. After running some tests, we have determined that the following programs will run on the IBM PC without a graphics adapter—with limitations as noted.

Vol. 4 , No. 2: Electronic Home Secretary
Vol. 4 , No. 3: Snap-Calc and Missile Math (without space ship graphics)
Vol. 4 , No. 4: Basic Addition and Subtraction and Tax Deduction Filer
Vol 4, No. 5: Division Tutor, Personal Loan Calculator, Quiz-Make, and Quiz-Take
Vol 5, No. 1: Quiz-Print, all Organizer programs
Vol 5, No. 2: FNkey, Organizer Reports, It Figures!
Vol. 5, No 3: Geometrix (with limited graphics), Budgetron (no bar graph option)
Vol. 5, No 4: Run-Day-View, Trig-Trix (with limited graphics)
Vol 5, No. 5: Electronic Typewriter, Bug-Out
Vol 5, No. 6: Electronic Postman

Bug-Out Identifies Bugs
Dear Sir:

I own an Atari 800XL with a single disk drive, a tape recorder, and a very dotty matrix printer. I have previously owned four other computers, all of which were ignored, for the most part, by the publishing industry. It was really nice to see the support for my machine at last.

This reaction is tempered a little after a couple weeks of (intermittently) typing programs from your magazine! The first thing I typed in was the Proofreader, and I must admit that it worked. The second thing I undertook was the NanoProcessor and my experience was not so great.

The first problem arose when I tried to enter line 920, a DATA item with a clear and unmistakable Pounds-Sterling symbol. Atari X models have such a symbol; however, the use of it normally excludes the other graphic characters used in the NanoProcessor display. I eventually figured out that what you must have meant was a **SHIFT +** character used (by Atari mostly in other languages than BASIC) for mod division.

The second problem was rather more of a bug, and took a while to figure out. The program checks the location 764 for last key pressed, and branches accordingly. At least, it ought to; it doesn't. The problem seems to be well illustrated by the segment from line 1500 to 1580. The variable B contains the last-key-pressed identity from location 764; line 1520 tries to match it against the ASCII value of the control-panel keys. But the Ataris use three interlocking character set designations: ASCII; Internal code, which is the sequential position of the character in the shape table; and Keyboard code, which bears little resemblance to the other two. Keyboard code is tied to the row-column position of the keys, and it is a member of this set which is stored at 764. Here's a one-liner for you to illustrate the point:

```
1 FOR A=1 TO 256:A = PEEK(764):
?CHR$(A);" ";:NEXT A
```

PLEASE type it EXACTLY as shown. The continuous reassignment of the counter variable value within the loop keeps the thing going forever, or until you have had enough (use BREAK to get out). Press any key, alone or with either or both of CONTROL and SHIFT keys. Try the Atari key (inverse toggle) and the CAPS, ESC, and RETURN keys. Try using each key with and then without the CONTROL. Try any key presses you like. CTRL-1 which pauses all action and then unpauses it, and the BREAK key will not give unusual results.

But back to the program bug. I thought of looking for the keys that would give the result, in effect reassigning the control panel in the game. Almost immediately after that, I thought of altering line 1580 so that the correct connection with ASCII would result. Both of these approaches had to be rejected because not all ASCII values represent valid keycodes. Not even all 6- or 7-bit numbers are valid. As a matter of fact, it turns out that some of the values required for the program are not valid K-codes. The only workable solution I found was to rewrite the following two lines.

```
1520 READ ZQ: IF B=ZQ THEN 1550
1580 DATA 49,50,51,52,60,62,80,69,6,
     73,82,76,62
```

I have not yet had a chance to play with the program, to see whether or not it is completely debugged. At least these fixes get it started running.

I look forward to the next few issues of your magazine. I hope it lives up to the high promises made by issue 5.5, apart from these details.

Diane Hoffman
Oshkosh, WI 54901

We're glad to have such a knowledgeable Atari reader, Diane. Thanks for the comments. The first bug you noted was due to an oversight in our process for translating the program to our type-set listing format. The other bug seems to stem from an error you apparently made in typing in the program. Line 1460 does indeed PEEK location 764, but B does not contain the value from that location—it contains the ATASCII returned from the GET #2, B statement. Thus, that aspect of the program works correctly as published. Thanks, though, for the excellent discussion of the various ways Atari machines represent key input, as well as for the neat little "one-liner."

Word Pro From ProDOS To 3.3
Dear Sir:

I am having some problems with my Apple *IIc*. A few weeks ago, a friend of mine sent some documents he created with his word processor to me. The documents were on a ProDOS disk and were stored as text files. My problem is that most of the documents he sent to me that were stored as text files were stored under a sub-directory. My word processor can read text files when they are directly stored as text files, but not when they are stored under a sub-directory. My word processor translates the text files to files my word processor can understand. (The files are on a DOS 3.3 disk now because that is the only format my word processor can translate them from. I copied the files from the ProDOS disk. This procedure has always worked until now.) Do you know of any program or procedure that would allow me to store these text files directly onto the disk instead of under a sub-directory so I could print the files on my printer with my word processor? I need a solution really soon.

Also, I have heard about a Z-RAM board by Applied Engineering that would allow you to add 256 or 512K of additional memory and a Z-80 microprocessor to the Apple *IIc*. Their ad says that it comes with a RAM disk so programs could run 20 times faster. It also said that you could run regular Apple software and CP/M software without unplugging the board. Do you think you could review this product? It sounds very interesting.

Thank you very much for your help.

Pete Sison
San Diego, CA 92139

In order to convert a file from ProDOS to DOS 3.3 you need two disk drives and the ProDOS Convert utility from Apple. If you have these, the various sub-directories shouldn't give you any trouble in conversion—just specify the complete ProDOS pathname in the convert program. We are evaluating the Z-RAM card from Applied Engineering and will be reviewing it in a future issue.

Hard-Driving Junior?
Dear Sir:

Like thousands of others, I thoroughly enjoy your magazine. In particular, I have used your techniques to successfully add a second floppy drive to my PCjr. Now I wonder about adding a hard drive. With the rapid de-escalation in hard drive prices this would appear to be a valuable next step. Any possibilities of your experts telling us how to do that? I am sure a number of us would benefit greatly.

Warren T. Dent
Indianapolis, IN 46260

Warren, our experts are currently working on hard-disk drive solutions for the PCjr, so stay tuned to HCM for cost-effective means to expand your PCjr's capabilities. Also, if any other readers desire additional Junior enhancements, please let us know so we can work toward giving you what you need.

Terminal Tape Troubles
Dear Sir:

I recently purchased a modem through mail order for my Commodore 64. Somehow there was a mixup in my order and the company I ordered from sent me the terminal software on disk instead of cassette tape as I requested. I contacted the company and they told me that the terminal software was unavailable on tape. When I borrowed a friend's disk drive to load the program and try to transfer it to tape, I found that I could only access one line of the program which was a SYS command. Is there any way I can access the full terminal program and copy it to tape or will I be forced to purchase a disk drive to use my modem?

Matthew Correll
Grinnell, IA 50112

Buying a disk drive is not a bad idea Matthew—especially now that the price on the Commodore 1541 is so low—but it is not the only answer. Several fine terminal programs come on tape and even cartridge. The popular VIP Terminal from Softlaw is available on tape. The SkiWriter II cartridge from Prentice-Hall (Reviewed in HCM Vol. 5 No. 4) is both a word processor and terminal program. As for copying and modifying the program you have now, don't try. It may be copy protected, and the attempt may void its warranty. If what you have does not work with your current system, we suggest that you either buy a disk drive or return the package.

Mini-Amnesia On Mini-Memory
Dear Sir:

The Mini-Memory module for my TI home computer will not accept assembly programs from tape, or BASIC programs from the computer. I assume the battery has died.

Could you tell me if this can be remedied and if so, where do I send it? How much would it cost? How long would it take? Have I diagnosed the problem correctly?

Any information you could send me would be appreciated.

John Wood
Salem, OR 97306

John, it's good that you should bring up this subject. There are a lot of Mini-Memory car

Letters to the Editor CONTINUED

tridges out there with weak or failing batteries. There's a very good possibility that your Mini-Memory battery has finally gone to that great recharger in the sky. The symptoms you describe fit the dead-battery syndrome, though that doesn't rule out the possibility that the connector could be dirty or the cartridge misaligned so that the pins on the card edge don't fit correctly when the cartridge is inserted into the system. Fortunately, there is help on the way. We are currently tracking down the least-expensive sources for the battery, and will tell you how to replace it in next issue's "Home Computer Tech Note" for TI.

Personal Accounts-Payable?
Dear Sir:

I have an IBM PC and I am looking for a computer program that will do the following things. I was wondering if you would have this on any of the disks that are available through you, or if you might know where I could find it.

It is to manage my personal bills. I would like to be able to enter the company name, company address, or a code number for each company, subtract the payment made and have the balance show. The next month I would like to be able to subject the payment again and have the account balance show. It would work like an accounts payable file.

Thank you for your help.

Florence M. Glenn
Santa Ana, CA 92702

Florence, so far we have not published a program that will totally "fill the bill" for you. We are, however, developing such a program—one that will not only allow the kind of file you suggest, but also provide great flexibility in customizing other kinds of alphanumeric/calculating data files. Look for this software soon in an upcoming issue.

Looking For Artificial Intelligence
Dear Sir:

I will be a sophomore in high school next year and have acquired (through certain channels) a free hour each day which I spend working with Apple IIe computers. I also have a Commodore 64 at home and I am planning to buy the new C-128 when it arrives on the market.

My question is this: Do you know of any companies that specialize in literature and/or software on artificial intelligence programming? I intend to focus my future computer studies on this particular subject.

Thank you.

Philip G. Jones
Slater, MO 65349

Philip, your local library will probably have a few major books on Artificial Intelligence (AI), although such books are largely non-technical and are targeted for the popular market. Most famous among these is The Fifth Generation, *by Feigenbaum and McCorduck, which talks about Japan's push for AI. Others include:* Artificial Intelligence, *by L. Stevens,* The Thinking Computer, *by B. Raphael, and* Godel, Escher, Bach, *by D. Hofstader. If you want technical information and news of the current state-of-the-art, a newsletter published out of Scottsdale, Arizona, called "AI Trends," will fill your needs. You will find that most AI research these days, centers on "expert systems"—software that provides the kind of technical advice usually reserved for experts in a given field. For example, Hewlett-Packard employs an expert system on the assembly line to help diagnose defects in the structure of microchips. These programs use a combination of raw data bases and a "knowledge" base garnered by quizzing engineers on every conceivable "what-if" situation. Some of these professional techniques have trickled down to the microcomputer level—with programs like Personal Consultant for the TI-99/4A, Experlisp for the Macintosh, and the Knowledge Engineering System for the IBM PC. Human Edge Software offers a few such "decision making" programs for the Apple II family; but we don't know of any available for the C-64 or the C-128. (The C-128 should be compatible with AI software written for CP/M systems, however.) As these kinds of programs become more applicable to home use, we will certainly cover them in HCM.*

A Pound Sign For A Backslash
Dear Sir:

Imagine my surprise. I have never looked in your magazine, just at the cover. This time—Atari!

Your magazine seems much more imaginative than most of the others, especially in your programs and explanation sections. I see an excellent relationship growing.

However, is it just because you are new to the Atari field? Your NanoProcessor program in Vol. 5, No. 5 has a bug that can't be just a typo. Line 920 has a data statement which includes an English pound sign! I can access this symbol on my 800XL with the International Character function, but it reverts to a simple CONTROL 'H' graphics symbol on LISTing or RUNning. What did you have in mind?

Second, I can't get the program to work. I have CAREfully checked my listing but when I RUN the program I get an Input Statement Error (#8) on line 820. For some reason my computer will not accept the line 920 data statements. If I remove the offending statements (losing the switch pointer) the program runs for awhile until I enter the third line of the sample program where I get a String Length Error (#5) in line 1660.

Last, but not least, Atari Bug-Out doesn't DO anything! It runs, it asks for output device, it says READING, but it doesn't access my cassette file. Is it supposed to operate my cassette recorder? Your instructions don't say!

I know this letter looks hyper-critical, but it's not meant that way. I'm super impressed with your first Atari issue. I know it can only get better.

Norm Palosky
Alexandria, VA 22304

Thanks for the input, Norm, and welcome aboard. For the answer to the first problem you mentioned, see "Bug-Out Identifies Bugs" (above). As for the Bug-Out program, other readers have experienced no difficulty; we can only conjecture that you have made a typing error.

No ProDOS-Compatibles
Dear Sir:

I have three different Apple II + compatible computers (from two different manufacturers) that I have been using heavily for over two years. I haven't had the slightest compatibility problem with loads of different software and with different operating systems, including CP/M and Pascal. But ProDOS doesn't work. None of the ProDOS diskettes that you have sent me works. Neither does one recently purchased program which is on a ProDOS diskette. The disks boot up displaying the title screen for the ProDOS release, then the computer locks up. The only key that the computer will respond to after that is RESET, which causes more problems. (Note: Two of the computers have 16K RAMcards in slot 0, and the other has a 128K Saturn board.) Is it necessary to purchase a ProDOS preboot diskette? Or is there some multiple-letter code to type in? Or has Apple figured out a way to defeat the "compatibles" with this latest operating system?

If I'm out of luck with ProDOS, would it be possible for you to send me your Home Computer ON DISK software with DOS 3.3?

Mike Rosa
Peru, IN 46970

Mike, on the second page of the Table of Contents in this issue, there are instructions for running ProDOS programs on Franklin machines. You don't say which brand of Apple-compatible you have, so we can't guarantee this will work for you. In Vol. 5, No. 5 , a letter from Henry Curry provides another way of fixing a ProDOS disk for Franklin—by using a "Sector Editor."

Expansion Of Coverage Hailed
Dear Sir:

This is the first time I have ever written to any magazine and I just wanted to say thanks. I am an avid Atari user and will help support anyone who supports Ataris. The first time I saw your magazine was several months ago on the newsstand. I was disappointed to see that you did not include Atari in your coverage.

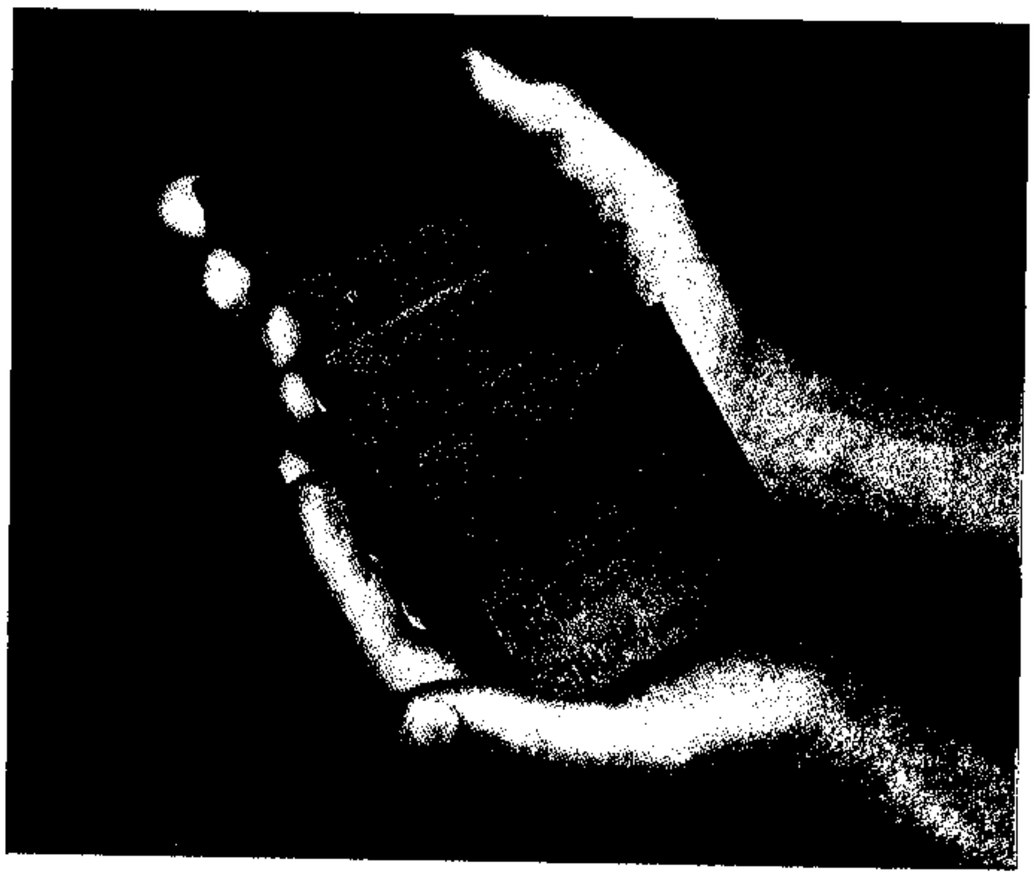
A couple of days ago, I was skimming through the magazine rack and saw issue 5.5 with the small banner announcing Atari coverage. I immediately bought the issue. A day later I mailed my subscription.

I have owned an Atari for about two years now. I started with the 400 and the 410 recorder, but now have an 800XL, disk drive, modem, and printer.

Again, thanks for the expanded coverage. You can expect me to be a continued subscriber.

Dennis E. Rees
Willard, OH 44890

Glad we are able to fill your need, Dennis. We look forward to bringing Atari users a fresh—and badly needed—approach to home-computer usage. **HCM**



TI Card Shuffler

by Randy Thompson
HCM Staff

Shuffle your file cards? Actually, computerized card shuffling can help you organize and expand your filing capacity.

[Note: TI Card Shuffler is the second part of TI Card Trix, which appeared in HCM Vol. 5 No. 5.]

In our last issue, we introduced you to *TI Card Trix*—a program designed to help you create and edit an inventory of file cards containing thoughts, book references, recipes, etc. Now we'd like you to meet *Card Trix*'s helpful companion—*TI Card Shuffler*. This companion completes important filing functions faster than even the best secretary. And it lets you shuffle your cards between many different cross-referenced folders. The program begins by offering you two options:

SEARCH CARDS
EXIT PROGRAM

Search Cards

This option brings you to the heart of the program. You can begin indexing your cards from 4 different *Card-Trix* folders. When you select Search Cards for the first time, the program asks you to enter the search parameters you want to use to pull cards from folders. First, enter the item for which you are searching—any sequence of up to 28 characters. Then, specify the field(s) by which you want to search. You can search using the Index, Subject, and Text fields or any combination of the three fields. You must select at least one field, or the computer becomes irritated and beeps at you. Search Cards consists of these 4 options:

GET A FOLDER
TOSS A FOLDER
SET SEARCH PARAMETERS
RETURN TO MAIN MENU

Get A Folder—When you select Get A Folder, the program asks for the name of the folder that you want to search through. After you enter the appropriate file name, the computer searches through the folder, noting all cards that meet the search parameter specifications. The computer then displays the number of cards, if any, that it finds.

You can perform this indexing search with a maximum of 4 folders in memory. If you want to search through a fifth folder, you must create space by discarding one of the folders with the Toss A Folder option.

Toss A Folder—When you select this option, the computer displays a list of all the folders that are currently in memory. To toss one of the displayed folders, enter the number that appears next to the unwanted folder. Don't worry: Tossing a folder does not erase it from the disk or cassette—only from memory.

Set Search Parameters—You can change the search parameters with this option. If you can't find what you're looking for, for example, try changing the fields by which you are searching.

Return To The Main Menu—When you return to the main menu with folders in memory, the menu expands to offer you two more options:

PRINT FOLDER INDEX
SAVE INDEXED FOLDER

Print Folder Index

The folder index is a cross-reference listing of all the folders in memory. The printout contains the name of each folder in memory, the item used in the search for these folders, and numbers corresponding to the card(s) where the item was found. Let's say you printed a search for the word "DOG." Now, any time you need to find a card containing "DOG," simply refer to the indexed printout. You can also display a cross reference on-screen by selecting the Screen option. You can press the space bar to pause the screen display as it scrolls by.

Save Indexed Folder

Want a whole new folder instead of just an indexed printout? With this Save option you can *shuffle* cards from the 4 folders in memory into a new folder.

When making a new folder, the program asks you for the disks or cassettes that hold the cards you want to shuffle. Now, insert the destination disk or cassette, and computer will save the indexed folder to the device with the file name you choose. (*Note: Each folder holds no more than 25 cards; therefore, the program places only the first 25 cards it finds in the new folder.*)

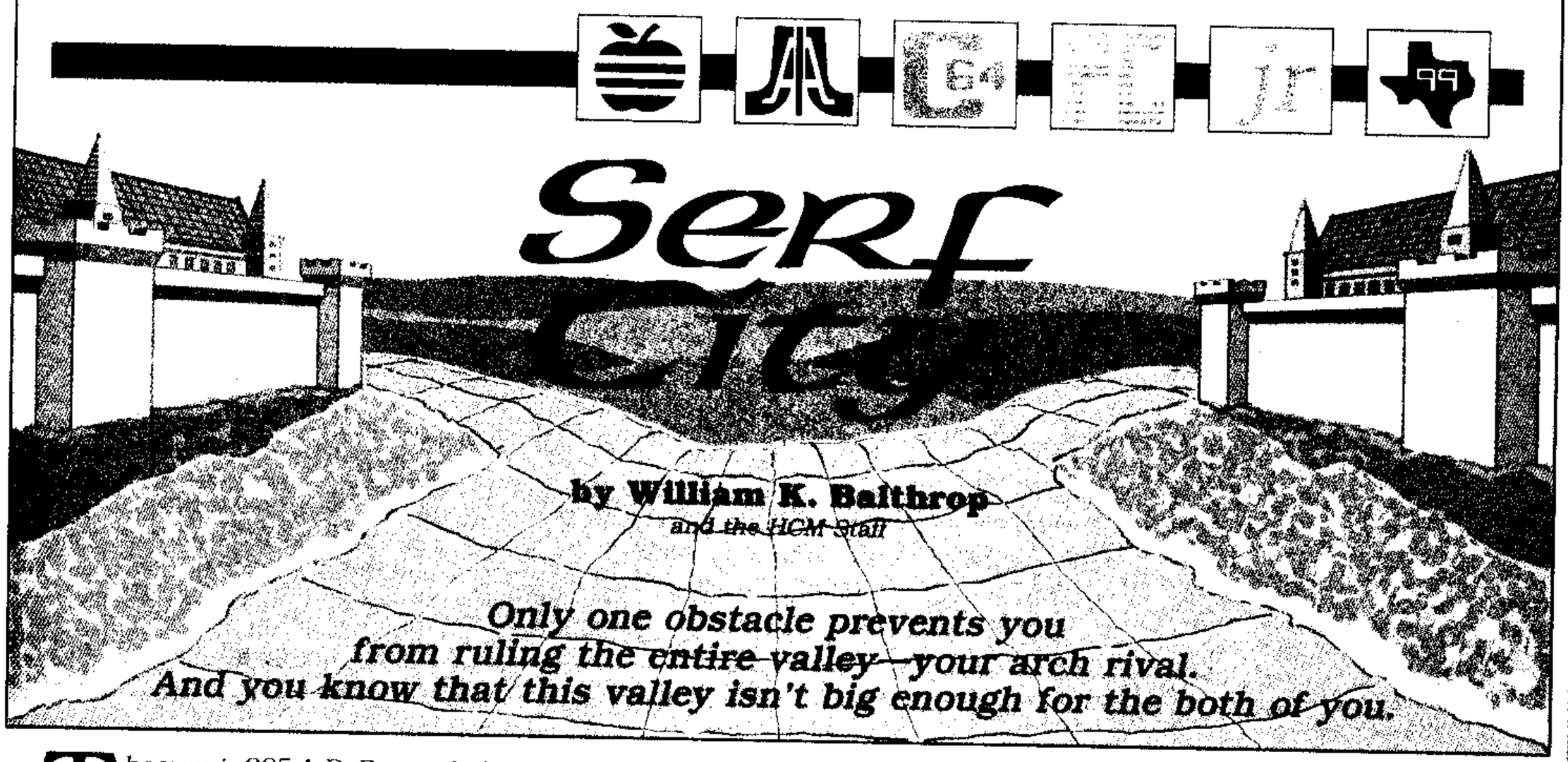
Exit Program

If you select this self-explanatory option and forget to make a printout or perform a search, don't worry. The program automatically asks you if you're sure you want to exit. At this point, you can either re-enter or exit *Card Shuffler*.

HCM Glossary terms: cross-reference, index, parameter.

For your type-in listings, see HCM PROGRAM LISTINGS CONTENTS.

HCM



SOFTWARE INSTRUCTIONS

The year is 985 A.D. From a balcony just outside your royal quarters, you survey your entire kingdom in the valley below. Across the valley sits the castle of your arch rival. For years, you and the other king have shared this valley, battling for land and food for your people. But now you tire of this status quo—you want the valley all to yourself. Your mind searches for a means to this end. An all out battle between armies? A fight-to-the-death between you and the other king? No. Experience has taught you that only through shrewd economics can you slowly weaken your opponent and rule the entire valley.

Serf City is a two-player economics simulation employing both text and graphics. Each player rules one of the valley's two kingdoms, both largely subsistent on wheat crops. Your objective is to increase the size, wealth, and population of your kingdom until you run your opponent out of the valley.

But this is no shoot-em-up arcade game. Only sly economic tactics can put you on the road to prosperity in this simulation. Several devices are at your disposal: You can purchase land, establish commerce, build wheat mills, plant crops, sell wheat, raise taxes, and hire or fire your own army. However, you don't have control over everything! The program contains random factors simulating unforeseen disasters that can severely diminish your kingdom's crops or population. (See Figure 1 for a schematic of how all these economic factors interrelate.)

Factors Of The Realm

When the program starts, you and your opponent must enter your names. Then, the main data display appears on the screen (see Photo 1). The data screen displays the current year—beginning in the year 985 A.D.—next to your name. A turn lasts one year, but the number of years that a game can last is unlimited. A game terminates when the players quit or when one player has no more population or land.

The main data display also lists your initial economic resources. Both players begin the game with 12 units of land, 90 people, 500 gold pieces, and 500 bushels of surplus wheat. The land is occupied by 2 units of commerce and 1 mill, leaving 9 units for farming.

After you've briefly studied the data display, you begin play by pressing the key corresponding to the economic device you want to manipulate. It's time to don your royal economic thinking cap and commence your most cunning ruling strategy! Prompts guide you through the rest of the simulation. When you complete entering data for the current year, press [RETURN] or [ENTER]. Now, it's your opponent's turn. When your opponent has finished entering data, press [RETURN] or [ENTER] again to display the year-end economic report, which is based on the decisions made during your last turn (see Photo 2). The program then shows your updated score on the main data display during your next turn.

Population

Population varies each year, due to births, deaths, immigration, and emigration. And these factors fluctuate depending on your kingdom's overall prosperity—a healthy food supply, reasonable tax rates, and a sufficient work force. To increase your population, for example, feed your people more (see "Food" section), making sure that you have enough field and mill workers to produce the extra food you need. Feeding your people less reduces your population. Population can also drop when the personal tax rate is too high (see "Personal Tax" section): people may move to other kingdoms with lower taxes (but never into your opponent's kingdom). If your population drops to zero, your kingdom falls, and your opponent rules the entire valley.

Employment

The program allocates portions of your population to work in your kingdom's fields, mills, and commerce. Field workers are your highest priority—without them you have no wheat to process for food. Therefore, the program automatically allots enough people to harvest all of the land you have planted. If you have sufficient population, the program then allocates any additional people to work in mills. Next, the program makes commerce workers out of any other people you have after fully staffing your fields and mills. Finally, the program considers any leftover population to be unemployed.

Commerce

The program expresses all merchant trade in terms of *units* of commerce, each unit costing 500 gold pieces and occupying one unit of land. Your compensation is the benefits of taxation (see "Commerce Tax" section).

You can place only one unit of commerce on each unit of land not occupied by mills. If your population is too small for the number of commerce units you own, some

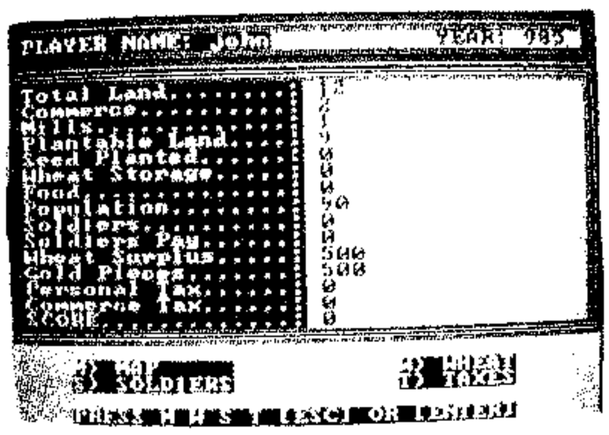


Photo 1
From the main data display, you can select and manipulate any of the economic devices you have at your disposal.

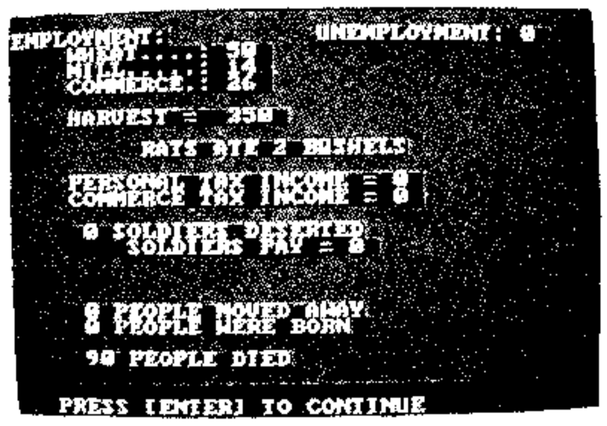


Photo 2
After each turn, the program displays the year-end economic report, listing the status of the economies of both kingdoms.

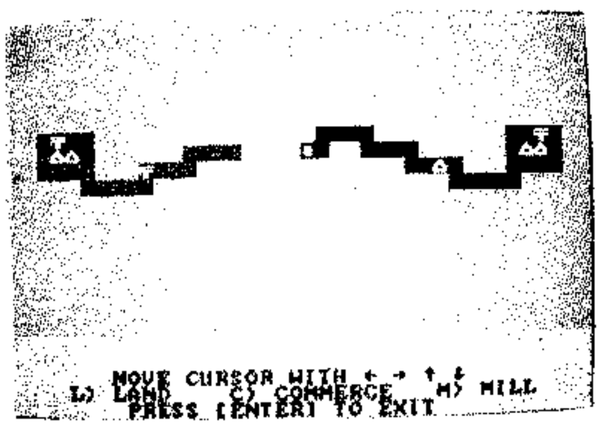


Photo 3
The map of the valley depicts land, mill, and commerce ownership by painting each kingdom a different color.

shops must close, thus reducing the flow of taxes.

To buy units of commerce, press [M] to select the Map option from the main data display (see Photo 1). A map of the valley appears, depicting your land and your opponent's land in different colors (see Photo 3). Position the cursor on top of the land you already own (see appropriate Control Capsule for cursor movement keys). Press [C] for Commerce, and a graphic symbol appears in place of the cursor to indicate the new addition. But, if you don't have the 500 gold pieces required to establish a unit of commerce, the symbol does not appear.

"It's time to don your royal economic thinking cap and commence your most cunning ruling strategy!"

Mills

Mills produce flour from the wheat you buy or harvest. A larger population requires more mills. Without at least one operating mill, all your people will starve. On the other hand, running too many mills pulls people out of commerce, reducing commerce-tax revenue.

A fully staffed mill (14 workers) produces enough food for 100 people. Understaffed mills produce less wheat—or with no workers, shut down entirely.

A new mill costs 1,000 gold pieces. To construct a mill, press [M] from the main data display to view the Map. Move the cursor on top of your own land, press [M] for Mill, and a graphic symbol representing the new mill appears on you land. Build as many mills as you want (only one per land unit), as long as you have enough open land and lots of gold!

Food

Your food supply relies on 3 factors: the number of mills operating; the number of people available to work in the mills; and the amount of wheat available for consumption.

Without enough food, your people will starve or move to another kingdom, reducing your personal-tax revenues (see "Personal Tax" section). Feeding your people slightly more than they need results in more births, fewer deaths, and an influx of people to your kingdom—which increases personal-tax revenue. However, large-scale over-feeding results in too many births, too few deaths, and too many emigrants—which eventually leads to over-population and an unstable economy. Overpopulation leads to high unemployment, with people depleting the food supply but not working to compensate for what they consume.

To decree the amount of wheat available for food, press [W] to select the Wheat option from the main data display. When the Wheat option menu appears, select Feed the People, and enter the number of bushels you want to feed your kingdom. Maintaining your current population requires a supply of at least 5 bushels of wheat per person.

Wheat Surplus

Surplus wheat includes all the wheat you harvest or purchase each year. As in real life, your yearly harvests vary. *Serf City* contains a random factor simulating yearly variances in harvests, such as bumper crops due to favorable conditions, or crop failures due to swarms of locusts.

You can allocate your surplus wheat however you wish—sell it for gold, plant it as seed, process it as food, or put it in storage. You may even want to use wheat as a commodity, buying bushels when the wheat price is low and selling when the price is high. But if you leave your wheat in surplus too long, rats might eat large portions (possibly all) of it.

Wheat Storage

Store your wheat that is not being used. To store wheat, press [W] to select the Wheat option from the main data display. When the Wheat option menu appears, select Wheat in Storage, and enter the *total* number of bushels you want to store. For example, if you want to add 20 bushels of wheat to the 50 bushels you already have in storage, simply enter "70." To take all of your wheat out of storage, enter "0" for the amount of bushels in storage, and the program automatically moves all your wheat into surplus. Now you can allocate the wheat any way you want.

Seed Planted

The amount of wheat you plant as seed determines the size of next year's harvest and the number of people required to work in the fields. You can plant up to 10 bushels on each unit of plantable land—land not occupied by mills or commerce. Planting more than 10 bushels per plot of land wastes the excess wheat seed.

Each bushel of seed planted requires one person working in the fields. Therefore, planting too much seed pulls workers from mills and commerce to work in the fields. And with no mill workers, your kingdom starves—even if you have a bumper crop! Let's say you have a population of 100 and plant enough wheat to put 95 workers in the fields, leaving only 5 people to work in the mills. These 5 workers cannot produce enough food to adequately feed your entire kingdom.

To plant seed, select the Plant Seed option from the Wheat option menu, and enter the desired amount.

Total Land

This total on the main data display represents all of the land you own. The only restrictions on the amount of land you can own are the dimensions of the screen and your own financial resources. The cost of land varies from 25 to 150 gold pieces—higher wheat prices raise the price of land.

Remember that every time you add a unit of commerce or a mill, you have one less unit of plantable land. Therefore, while expanding your kingdom, be sure to purchase enough land for crops to feed your people.

To acquire land, move the cursor to a plot of land that is *adjacent* to land you already own, and press [L] for Land. If you have enough gold to pay for this land, your color appears at the cursor's location.

Soldiers

You need soldiers to capture land from your opponent and to protect your own land from similar attacks. To seize a unit of your opponent's land, you must expand your kingdom's boundary until it touches your rival's boundary. Then move the cursor on top of your opponent's land (adjacent to your land), and press [L]. If you have enough gold to pay for the land, and enough well-paid soldiers to defeat the opposing army, the land is yours. However, as in any war, you pay a price—you lose a random percentage of soldiers.

To hire or fire soldiers, press [S] for Soldiers on the main data display, select either Hire or Fire, and enter the number you desire. Keep in mind that hiring soldiers pulls people out of your work force. Creating a large army to protect your lands may not leave enough people to work in the fields, mills, or commerce. On the other hand, too few soldiers may be an open invitation for your opponent to seize *your* lands.

Soldiers' Pay

You must pay every soldier an annual fee. The more you pay your soldiers, the harder they fight in battle. However, paying one soldier 100 gold pieces is not as wise as paying 5 soldiers 20 gold pieces each. And even in the 10th century, soldiers have their standards—pay your soldiers too little, and they will desert your kingdom, thereby reducing your population.

To set army wages, select the Pay option from the Soldiers option menu, and enter the amount you want to pay each soldier per year.

Gold Pieces

Gold is essential to the financial success of your kingdom. You can use gold to purchase land, wheat, mills, and commerce. You can acquire gold through taxation (see "Personal Tax" and "Commerce Tax" sections), or by selling wheat you have already harvested (see Wheat Surplus section).

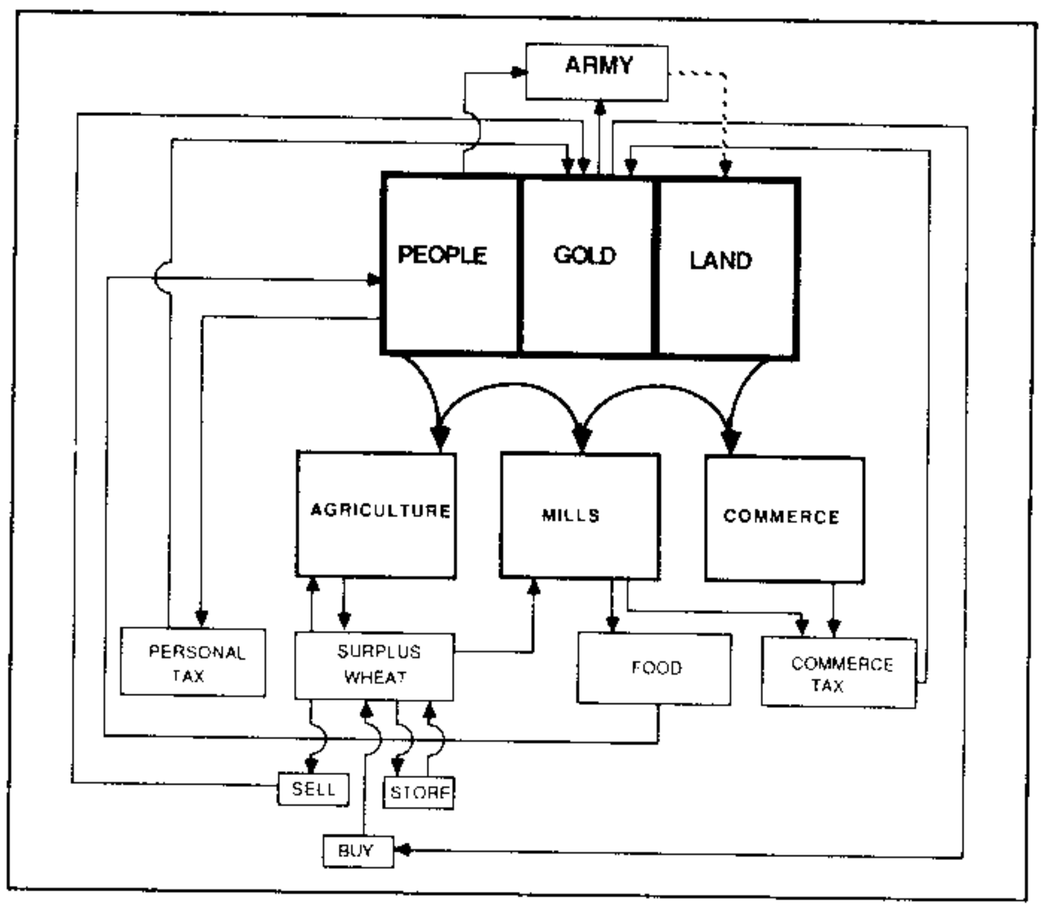


Figure 1
This diagram illustrates some of the intricate interrelationships that you must control to preserve and enrich your serf-based economy.

Commerce Tax

This tax is placed on your mills and the merchants engaged in commerce. Commerce-tax revenue varies, depending on the number of workers in mills and commerce and the prosperity of your people.

The tax also affects the number of people that can work in commerce and mills. For example, if you set the commerce tax at 10 percent, the merchants can employ 22 people per unit of commerce. At 30 percent, your merchants can afford only 18 employees per unit of commerce. Therefore, if you set the tax rate too high, the unemployment rate rises as some shops must close. Set the rate too low, and your revenues diminish. The maximum commerce-tax rate is 69 percent—commerce will not tolerate a higher tax rate.

Personal Tax

You can also establish, raise, and lower a personal tax to generate revenue based on population and the people's prosperity.

When over-taxed, your people spend less, reducing your commerce-tax revenue. Over-taxing may also drive people away and cause more deaths and fewer births.

To set the personal tax rate, select [T] for Tax from the main data display, press [I] to select the personal tax rate, and enter the percentage you desire. Like the commerce tax, the maximum personal tax rate is 69 percent—any higher and the people will revolt.

For your type-in listings, see HCM PROGRAM LISTINGS CONTENTS. **HCM**

CONTROL CAPSULE

Serf City

KEY	FUNCTION
E	Move map cursor up
S	Move map cursor left
D	Move map cursor right
X	Move map cursor down
ESC	Save/exit game
RETURN	Exit current option

CONTROL CAPSULE

Serf City

KEY	FUNCTION
↑	Move map cursor up
←	Move map cursor left
→	Move map cursor right
↓	Move map cursor down
ESC	Save/exit game
RETURN	Exit current option

CONTROL CAPSULE

Serf City

KEY	FUNCTION
E	Move map cursor up
S	Move map cursor left
D	Move map cursor right
X	Move map cursor down
F1	Save game
F7	Exit game
RETURN	Exit current option

CONTROL CAPSULE

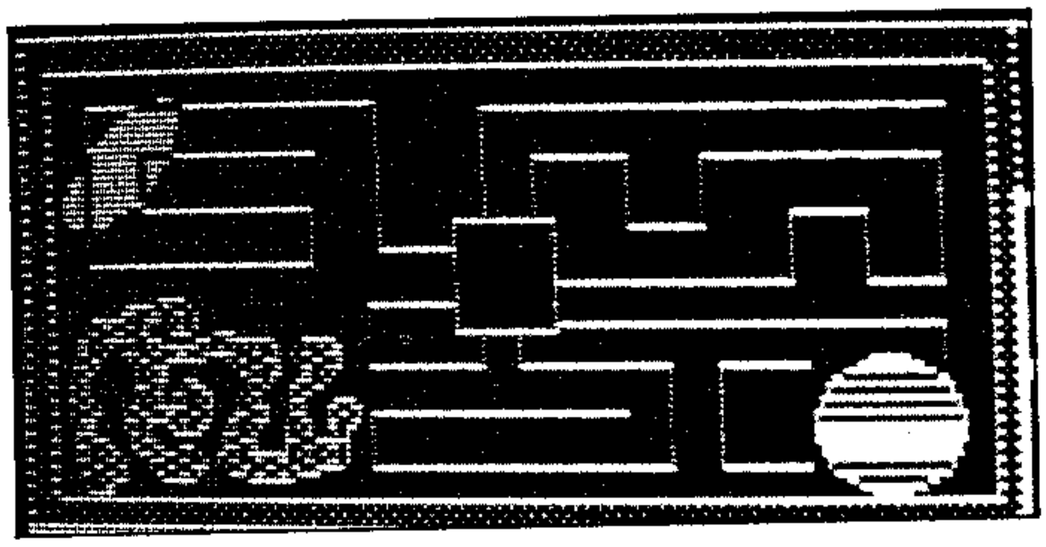
Serf City

KEY	FUNCTION
↑	Move map cursor up
←	Move map cursor left
→	Move map cursor right
↓	Move map cursor down
ESC	Save/exit game
ENTER	Exit current option

CONTROL CAPSULE

Serf City

KEY	FUNCTION
E	Move map cursor up
S	Move map cursor left
D	Move map cursor right
X	Move map cursor down
FCTN 9	Exit game
ENTER	Exit current option



CELL MATES

by William K. Balthrop
& Wayne Koberstein
HCM Staff

How complex is a living cell? Is it a static structure—or a dynamic system? How do symbiotic organisms make all cellular life possible? Find some answers with this fascinating bio-simulation.

There is more going on inside a living cell than meets the eye. Even the most powerful microscopes have not yet revealed the entire story—the secret life of the cell. Ask most people what they know about this basic life unit, and their reply, if any, may invoke a static picture of the cell and its interior: the cell wall, the nucleus, and "a bunch of little floaty things." But when biologists look closer at these little floaty things, they see participants in an incredibly complex, wonderfully cooperative, and dynamic *ecosystem*; for the metabolism of a living cell involves some strange and alien participants who apparently learned to live together millions of years ago. Directing this metabolic dance, communicating with each cell part through encoded molecules, is the nucleus. Some of the participants—called *organelles*—possess their own DNA (deoxyribonucleic acid) and reproduce themselves independently from the nucleus. Together, they are all *Cell Mates*.

What if you could place yourself at the center of this dance—consciously balancing the entire system moment by moment? There may be no better way to appreciate just how dynamic and basically cooperative this system is. We can't literally perform such a miracle, but we *can* give you *Cell Mates*, a simplified computer simulation that places you in the role of the nucleus, the cell's "choreographer."

The Major Performers And You

Plant and animal cells each have a somewhat different cast of dancers. But in both cases, each kind of organelle has a specific role to perform. In animal cells, some organelles, called *ribosomes*, manufacture various proteins. Another group, the *Golgi apparatus*, "packages" these proteins for internal use or secretion (export). *Lysosomes* are the garbage collectors. And tiny mobile chemical plants, called *mitochondria*, supply energy for all of these activities.

Although these organelles perform the cell's main functions, there are many other participants performing minor, but essential parts. Because of the difficulty in portraying all of the cell's diverse components, these minor players are beyond the scope of *Cell Mates*.

In *Cell Mates*, you direct the show, allocating energy to these 4 essential functions: protein production, waste disposal, ribosome production, and protein export. Your goal is to direct these functions in such a manner that the cell is able to divide—effectively producing two new cells in its place. When the cell divides, you receive a total score. The less time you take to reach cell division, the higher is your score.

You must meet certain requirements before the cell can divide: You must double the supply of protein as well as the number of several internal components (such as the ribosomes); and you must support the larger family of surrounding cells by exporting protein.

After displaying the title, the program presents the actual playing screen. This screen contains a graphics representation of the cell (in the upper-left corner), and 12 indicators that show the status of the various processes in the cell. You can gauge your progress by viewing these indicators. Some of the indicators contain areas highlighted against the background color. To maintain and eventually reproduce the cell, you must manipulate the cell's life processes so that the pointers remain in these "safe" areas. Indicators that contain no distinguished area show the actual level of energy going to specific activities.

Special thanks for technical advice goes to W.R. Sistrom, University of Oregon Department of Biology.

THE SECRET LIFE OF ORGANELLES

Millions of years ago, life wasn't what it is today. In the still waters of land-locked lakes and ponds, primitive single-celled creatures without true nuclei learned to obtain energy and reorganize materials through ingenious chemical processes. These were the ancestors of the organelles; for as life became more and more organized, such specialized organisms became permanent fixtures within higher lifeforms. Some—such as the *chloroplasts*—integrated themselves into the first true plants, capturing energy from the sun for use by the entire organism. Others—such as the *mitochondria*—took oxygen from the atmosphere (mostly produced by the activity of the chloroplasts) and used it to "burn" complex molecules as an energy source for animal life. Even today, these organelles possess their own DNA, and reproduce themselves independently of the organisms they support. (For example, mitochondria in a fertilized egg cell come partly from the mother cell and partly from the sperm—and as this cell divides, each new cell starts off with half the mitochondria of a mature cell.)

Why Life chose to organize itself in this way is a question that has preoccupied many people—not just the biologists who discovered these facts. For whatever reasons, the organizing process was primarily *cooperative*, rather than competitive. It is still an open question whether organelles live for the sake of the higher organisms—or whether we are living for them. What is apparent is that all the players in this game have a total stake in the system. Neither the organelles nor the organisms in which they reside could live without each other. The fact that each maintains a separate genetic indentity actually serves to dramatize the cooperative nature of their relationship. Of all life's mysteries, this may be one of the most curious and wonderful.

[For more on this subject, read *The Lives of A Cell* by Lewis Thomas.—Ed.]

We have incorporated a delay factor into the simulation, representing the real-world lag between the initiation of an action and its effect. This delay is not constant—it varies with internal and external conditions. Because of this built-in lag, any action that you take does not immediately show up on the screen indicators.

Dividing Energy

This simulation focuses on the 5 most important cell components: the mitochondria, the nucleus, the lysosome, the ribosome, and the Golgi apparatus.

". . . when biologists look closer at these little floaty things, they see participants in an incredibly complex, wonderfully cooperative, and dynamic ecosystem."

Mitochondria—In animal cells, mitochondria produce ATP (adenosine triphosphate)—life's most available source of chemical energy. Similar organelles in plant cells—the *chloroplasts*—actually perform photosynthesis (the conversion of light into energy). All of the cell components, including the mitochondria themselves, use ATP as their energy source. Without this substance, the cell would not be able to carry out any activity. Mitochondria manufacture ATP by combining products of available nutrients with oxygen (and emitting carbon dioxide as a waste product). Thus, when cellular biologists talk about *respiration*, they are talking about a process that occurs inside the mitochondria. The amount and quality of available nutrients directly affect the amount of ATP that the mitochondria can produce.

Mitochondria are mobile, and can move to the areas of the cell where energy is most needed. Most animal cells contain anywhere from one to hundreds of thousands of mitochondria. For the sake of simplicity, our cell representation contains only one. To increase the energy allocation to any of the cell's components, simply move the mitochondrion to the corresponding organelle symbol by pressing the appropriate key (see the Control Capsule). Once the mitochondrion reaches the designated organelle, it automatically starts producing ATP. The mitochondrion does not manufacture ATP while it is in transit between cell parts.

Nucleus—The nucleus is the cell's control center—the "central computer." It sends signals in the form of encoded molecules to each part of the cell, telling it to carry out various operations. (In addition, the nucleus carries genetic information in its DNA that determines the form of new cells resulting from cell division, or "mitosis.") Again for simplicity, we have symbolized the message-sending activity of the nucleus with one example: the message to increase production of ribosomes. Of course, the nucleus requires energy each time it sends a message.

To allocate more energy to the production of ribosomes, press **N**. The mitochondrion responds by migrating to the nucleus. Once the mitochondrion reaches the nucleus, it begins producing ATP. The increase in energy shows up on the screen indicator labeled **RIB.ENERGY** (energy for the ribosome-production message).

As you allocate this energy, the **RIBOSOMES** pointer shows an increase by moving to the right. If you direct the mitochondrion away from the nucleus, the **RIB.ENERGY** indicator shows a decline, and the **RIBOSOMES** indicator ceases to increase.

Ribosomes do not live forever. They die occasionally, and become part of the cell's waste. You need to replenish your supply of ribosomes occasionally to maintain their numbers.

Ribosomes—Ribosomes produce protein. These tiny organelles are attached to the *endoplasmic reticulum*—a convoluted membrane extending out from the nucleus. The ribosomes assemble various proteins according to patterns supplied by the nucleus. Although we symbolize the ribosomes in a discrete corner of our cell, in reality they are more evenly distributed.

Protein is the cell's prime material. Before a cell can reproduce, it must double its original supply of protein. The speed at which ribosomes produce protein depends on the total ribosome quantity. If you need to produce protein at a faster rate (to keep up with the protein exportation rate, for example), you must increase this quantity. (See the Nucleus section above for directions on increasing the ribosome count.)

Press **R** for Ribosomes to allocate energy to the production of protein. This signals the mitochondrion to migrate toward the ribosomes (upper-right corner of the cell area). Once the mitochondrion reaches the ribosomes, the **PRO.ENERGY** indicator shows an increase as the protein energy level ascends. This increase in

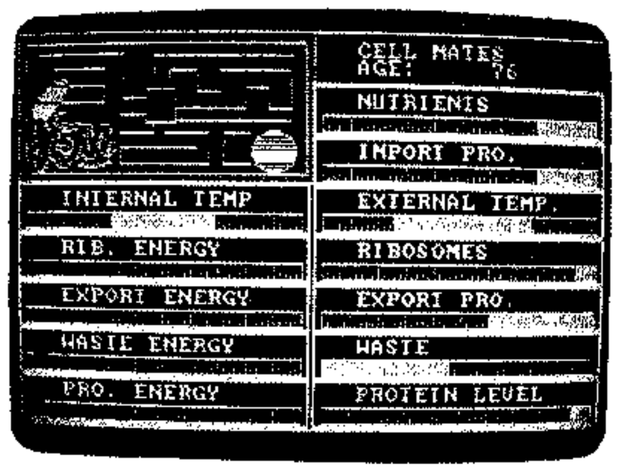


Photo 1
Cell display from IBM version at top left shows (clockwise from display's upper-right): protein-producing ribosomes; waste-eliminating lysosome; and the convoluted Golgi apparatus where the mitochondrion (green) is supplying ATP energy for protein export. Nucleus is at center with extending endoplasmic reticulum membrane.

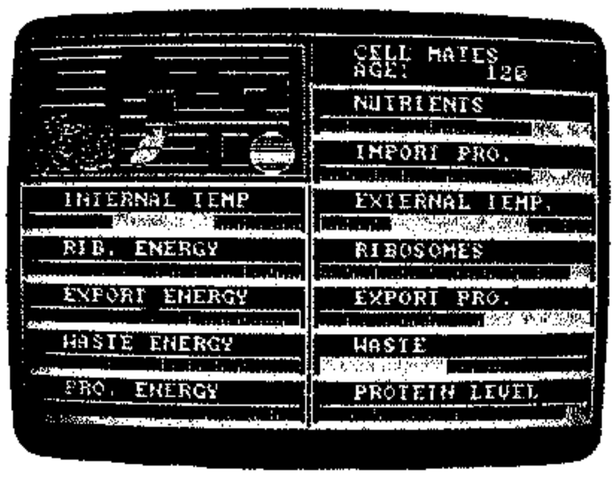


Photo 2
As the waste level continues to increase (as shown by waste indicator), the mitochondrion produces no ATP while traveling from the Golgi apparatus to the waste-eliminating lysosome at lower-right.

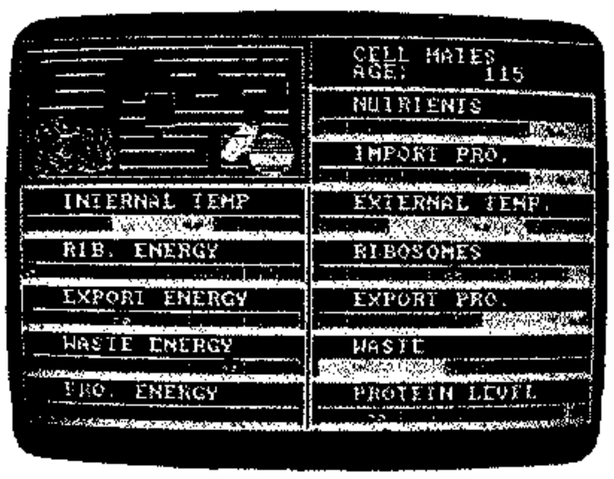


Photo 3
Now at the lysosome, the mitochondrion's ATP energy goes toward eliminating cell waste products—and in the nick of time, waste levels begin to decrease.

energy shows up in the PROTEIN indicator. But, as we stated, to achieve the desired increase in protein, you may need to increase the number of ribosomes first.

Lysosomes—The lysosomes are the cell's housekeepers. They carry degradative enzymes and travel around inside the cell collecting and digesting larger chunks of waste composed mostly of worn-out cell parts. This waste is always accumulating, so you must frequently budget energy for lysosome activity.

As the amount of waste in the cell increases, the WASTE pointer moves to the right. The only way to reduce the level of waste is to increase the amount of energy allocated to waste disposal. To do this, press L for Lysosome. In response, the mitochondrion moves to the lysosome symbol at lower-right corner of the cell. Once the mitochondrion arrives, the energy level shown by the WASTE ENERGY indicator increases. After the energy reaches a sufficient level, the amount of waste starts to decrease as the lysosomes do their job.

Golgi Apparatus—Your cell is also in the import/export business. Few cells can survive without the aid of other cells, so you must not only support your own cell, but those cells surrounding you. You must supply these cells with the protein they need for their existence, and in return you will receive protein from them—some of the nutrients vital to the production of ATP. The Golgi apparatus is responsible for exporting protein to other cells. As a secondary role, the Golgi apparatus spends a small percentage of its total energy packaging up waste and sending it out of the cell. If you fail to export sufficient protein, both the level of imported protein (IMPORT PRO.) and the nutrient level (NUTRIENTS) decline.

"Tiny mobile chemical plants, called mitochondria, supply energy for all of these activities."

To increase the amount of protein that you export, press G for Golgi apparatus. The mitochondrion responds by moving towards the lower-left corner of the cell area to the Golgi apparatus symbol. Once the mitochondrion reaches the Golgi apparatus, EXPORT ENERGY begins to show an increase. If you have a supply of protein, EXPORT PRO. also shows an increase.

The rate at which you can export depends entirely on how much protein is available. As you export protein, PROTEIN LEVEL declines. As the protein level decreases, EXPORT PRO. starts to decrease. All of the protein you export must come from the protein you have produced (shown on the PROTEIN LEVEL indicator).

Additional Factors—The AGE display indicates the passage of a relative amount of time. The AGE counter does not represent any real time frame—to make the simulation playable, we have accelerated certain aspects of the cell's metabolic system, while slowing down others.

The older the cell gets, the harder it must work to produce protein (or ribosomes). You may also notice that the average amount of nutrients and imported protein also drops off with age. Many factors affect an individual cell's lifespan—too many to explain here. But every living cell has a built-in "time limit": The cell must reproduce within a specific period of time, or it will die. Our model reflects this basic fact.

The NUTRIENTS indicator shows the cell's current food supply. The mitochondria use these nutrients to manufacture ATP. In order for the cell to reproduce, the pointer must be in the highlighted portion of the display.

The IMPORT PROTEIN indicator shows how much protein is available from the surrounding cells. This protein contributes to the cell's protein level. In order to reproduce the cell, you must move the pointer to the the highlighted area.

The INTERNAL TEMP indicator shows the cell's current temperature. The optimum temperature is directly at the center of the indicator. The highlighted area is the safe zone for the cell. If the pointer leaves this area, the cell dies. Even if the pointer remains in the safe zone, high and low temperatures (relative to center) affect the efficiency of every metabolic operation.

Most individual cells have little control over internal temperature when the external temperature goes outside a life-supporting range. Therefore, the "generic" animal cell that we depict in this simulation depends on the temperature of the tissue of which it is a part. If the cell doesn't adequately perform its part in this tissue structure—neither maintaining itself, nor supplying needed protein to the surrounding cells—its external environment tends to degrade. This can cause a drop in external temperature, as well as a shrinking nutrient supply. Normal cell activity usually maintains a safe internal temperature. If the energy levels for ribosomes, export, waste, or protein increase, the internal temperature also increases slightly. But it is more important to ensure a steady external temperature by exporting enough protein to maintain the health of the entire tissue. If the external temperature does become excessive due to high mitochondrion activity, you can press M to send the Mitochondrion to its neutral, inactive corner.

Of course, the external temperature sometimes goes outside a safe range, no matter what an individual cell does. If it remains outside this range, the cell will die—which, as they say, is Life.

Life Goes On

A BASIC computer program can only "scratch the surface" of a living cell's complex nature. While being purely fun to operate, *Cell Mates* should also serve to awaken your curiosity about this fascinating subject. Perhaps just knowing that your very life depends on the activity of microscopic creatures with their own separate genetic identities will contribute to your curiosity. As biologists continue to unravel the mysteries of cells in general, and organelles in particular, you may be in a better position to understand and appreciate this new knowledge. Here's to a lifetime of learning about life itself!

CONTROL CAPSULE
Cell Mates

Key	Function
M	Move mitochondrion to home position—upper-left corner of cell area
G	Move mitochondrion to Golgi apparatus—lower-left corner of cell area
L	Move mitochondrion to lysosome—lower-right corner of cell area
R	Move mitochondria to ribosomes—upper-right corner of cell area
N	Move mitochondria to nucleus—center of cell area
Q	Quit option—exit simulation

HCM Glossary terms: ATP, DNA, ecosystem, endoplasmic reticulum, enzyme, Golgi apparatus, lysosome, organelle, metabolism, mitochondrion, nucleus, protein, respiration, ribosome, symbiotic.

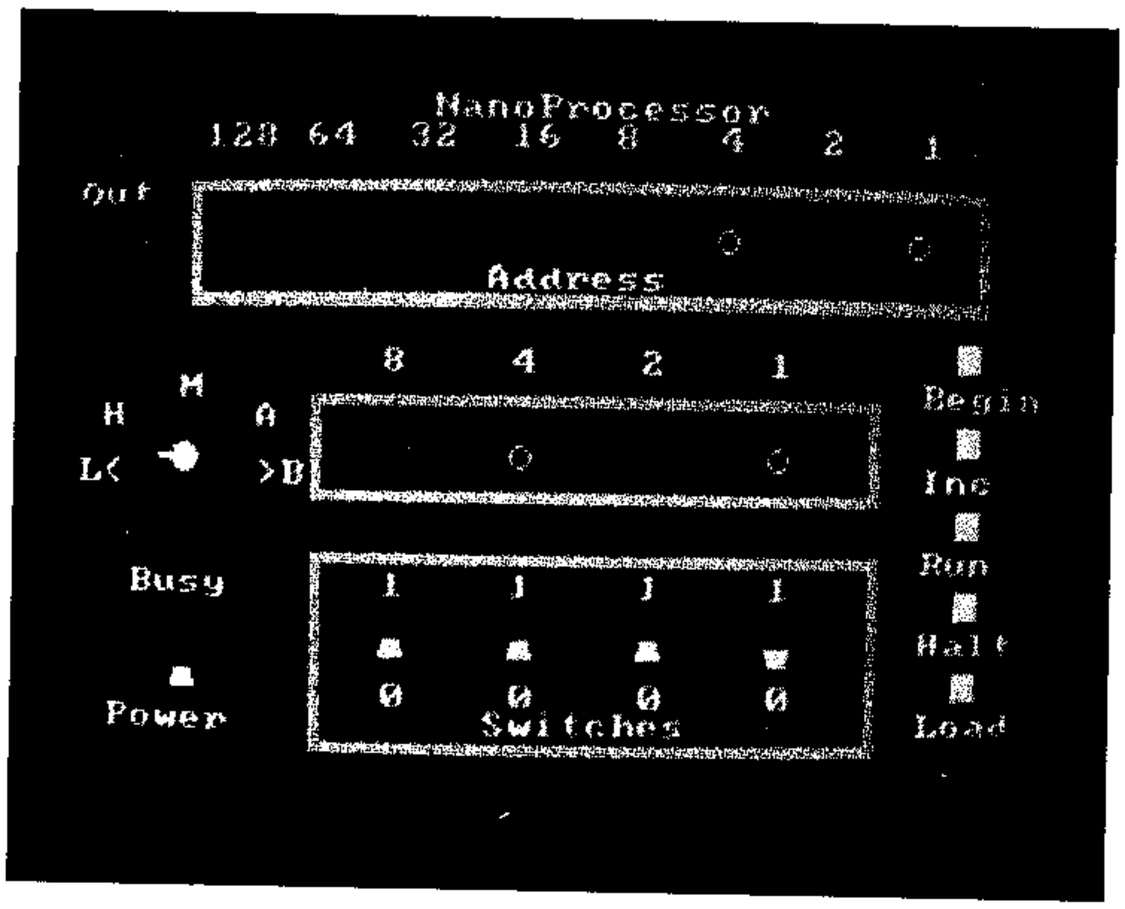
For your type-in listings, see HCM PROGRAM LISTINGS CONTENTS.

HCM

The NanoAssembler

by Roger Wood
HCM Staff

This companion to NanoProcessor shows you how an assembler can provide easy access to machine language—by translating simple instructions into the computer's native tongue.



In the last issue (*HCM* Vol. 5, No. 5), we presented *NanoProcessor*, a program that introduced the concepts of machine-language programming. This program demonstrated how a microprocessor works at its most fundamental level. Although entering and running simple programs on the *NanoProcessor* can be fun, longer and more complicated machine-language routines are another story. Even with short programs, you probably discovered what a time-consuming and error-prone process it can be to enter machine language one bit at a time.

To alleviate the difficulties involved in working with machine language, early computer users created programs called "assemblers." An assembler is a human-to-machine translator. It operates from a "dictionary" of mnemonics (a combination of letters that humans can understand), translating these mnemonics into the numbers of machine code. Using assemblers, you can write a program with the more easily remembered mnemonics, and let the computer create the actual machine language (the ones and zeros).

Thus, we present the *NanoAssembler*; a program that will teach you how to use assemblers. With the *NanoAssembler*, you will be able to write long, complicated programs for the *NanoProcessor* much more easily than you would using machine language.

Source Code To Object Code

You may find that many people refer to "assembly-language programs" and "machine-language programs" interchangeably, as though they were the same thing. Actually, an assembly-language program is a text file—known as a "source file"—that the computer cannot execute directly. It is simply a series of text lines comprising mnemonics, numbers, and labels. Before the computer can run such a program, the source file must be "assembled" or translated into a machine-language file—also known as an "object file."

Take a look at Sample Program 1, which you can load and run on the *NanoProcessor*. You may recognize this program, as it is identical to Sample Program 1 in the last issue. The two left-most columns, entitled Addr and Code, contain the machine language, (object code), which makes up the program. You can enter this object code bit by bit, or you can enter the more easily read and (with some training) understood assembly language (source code), contained in the Line, Label, Mnemonic, and Remark columns. The Remark column is like a **REM** statement in BASIC. It makes the program much easier to read and understand.

Our *NanoAssembler* package consists of two BASIC programs: the *NanoEditor* and the *NanoAssembler*. The *NanoEditor* is a simple text editor that lets you enter your program as source code and save it to disk (or alternatively tape on the Atari, Commodore, and TI computers). *NanoAssembler* can then read and translate that file into a corresponding file of object code, which you can save to disk or tape. You can then load the object code into the *NanoProcessor* and run it.

Creating A Program

We will use Sample Program 1 to demonstrate how the *NanoEditor* and the *NanoAssembler* work. To start, Load and **RUN** The *NanoEditor*. You begin with this menu:

```
1) EDIT
2) FILES
3) PRINT
4) EXIT
```

Choose the Edit option, which allows you to create and modify files. The *Editor* now displays the command prompt: **CMD**. You may enter one of 5 single-letter commands:

Command	Function
A	Add a line of text
E	Edit a line of text
D	Delete a line of text
I	Insert a line of text
L	List

To begin creating a new file—in this case Sample Program 1—press **A**. In response, the *Editor* displays line 001, with a flashing cursor waiting for your input. For each line of source code, the *Editor* provides a line number ranging from 001 to 200. When you enter the **A**dd-a-line mode, the program always displays the cursor on a new line of source code—one line past the last line in memory. You can automatically advance to the next line by pressing [**ENTER**] or [**RETURN**]. To exit the **A**dd-a-line mode, press the [**ESCAPE**] key (see your computer's Control Capsule if your machine does not have an Escape key).

Now enter the contents of the Label, Mnemonic, and Remark columns. Because our *Editor* is in BASIC, your text input will be slower than with a full-blown word processor. The Label column is empty in line 001 of our sample program, so press the proper key or key combination (see your Control Capsule) to tab into the Mnemonic field. (We will explain labels below.) Now type in the first instruction: LDA# 3. You must enter the text *exactly* as it appears in the listing, or the *NanoAssembler* program will not interpret the code properly. Make sure there is no space between the A and the #. You must, however, place a space between the # and the 3.

This spacing is critical because the Mnemonic field actually consists of two sub-fields; and the space acts as a separator for these sub-fields. The left sub-field is the "op-code," or instruction field, which defines the actual instruction. In line 001, the op-code is LDA#. The right field contains the "operand." The operand is either a two-nibble address or a single-nibble quantity to be loaded or stored in a register or memory location. It defines the number that the op-code is to operate on. In line 001, the number 3 (%0011) is the operand.

After you have entered the first instruction, you may tab into the Remark field. On a program as short as this one, however, you may choose to save time by omitting the remarks. Continue entering lines 002, 003, and 004 in a similar fashion.

Once you've entered part or all of the program into memory using the Add command, you can use the other editing commands. Each of these commands prompts you for a particular line number. E lets you Edit an already-existing line in memory. D allows you to Delete a line, and I lets you Insert a line. The L command lets you List up to 10 lines of a program to inspect what is in memory. If the program extends more than 10 lines beyond the beginning line number that you specify, you have the option to either continue listing more lines or quit and return to the command line.

Labels As Labor Savers

In line 005 (HERE JMP HERE), you encounter an important assembly-language tool—the "label." In the *NanoAssembler*, we define a label as a group of up to 6 alpha-numeric characters, beginning with a letter—in our example, the word HERE. Assembler programs use labels in place of numeric quantities. In this case, HERE represents the address to be JuMPed to. One major advantage of labels is that you do not have to know the actual numeric addresses used in a program. Instead, the assembler uses the labels to assign the correct address to a particular instruction for you.

Before continuing, let's clear up an area that sometimes confuses a beginner at assembly language: the difference between line numbers of a source file and addresses of an object file. Each line in a source file contains only one op-code. But when you assemble the source file into object code, the op-code may require as many as three addresses (see Figure 1 for the number of nibbles each instruction requires). Thus, a source file's line numbers and the actual addresses of the object code almost always differ. When the *Assembler* prints out its listing, the addresses and codes are located on the line just below the source code, representing the order of events during assembly.

By inspecting the two left-hand columns of Sample Program 1, you can see that the address to be JuMPed to is 6. You know this only because we have already assembled (or translated) the source code on the right into the object code on the left. If we hadn't provided the machine code, however, you would have to assemble all of the instructions to discover what address you wanted to JuMP to. The use of labels saves you from this tedious task and is one of the primary advantages of assemblers.

When you finish entering line 005 and press [RETURN] or [ENTER], a prompt tells you to enter line 6. This program has no line 006, so press the [ESCAPE] key for your machine (see your Control Capsule), and the program returns you to the command line. Now you can use the List command to see if you have entered everything correctly. If you find any errors, you can Edit the line or lines that they occur in. If you change a line, then decide that you don't want those changes, you can press the [ESCAPE] key instead of [RETURN] or [ENTER] to revert back to the original version of the line. This option is also available if you select Insert, but change your mind before finally entering the line.

From Editor To Assembler

After you are sure that you've correctly entered the program, save it to disk (or tape on Atari, C-64, or TI). To save your file, select option (2) Files. Then select the appropriate menu options, and enter the file name. If your operating system does not normally support extensions to file names (all but Atari and IBM), the name must be at least two characters shorter than a normal legal file name. The program will automatically append a .S (_S on the TI), for Source, so that you can use the same name for both source and object files without any confusion. If you have a printer, you may also wish to get a hardcopy of your program. This is helpful when you are tracking down errors during assembly. To use

CONTROL CAPSULE

NanoEditor

KEY	FUNCTION
ESC	Escape
Edit Mode:	
BACKSPACE	Backspace
CONTROL D	Erase line
TAB	Tab.
←	Cursor left
→	Cursor right
RETURN	Enter Line

CONTROL CAPSULE

NanoEditor

KEY	FUNCTION
ESC	Escape
Edit Mode:	
DELETE	Backspace
SHIFT DELETE	Erase line
TAB	Tab
CONTROL ←	Cursor left
CONTROL →	Cursor right
RETURN	Enter Line

CONTROL CAPSULE

NanoEditor

KEY	FUNCTION
F1	Escape
Edit Mode:	
DEL	Backspace
F3	Erase line
F5	Tab
CRSR ←	Cursor left
CRSR →	Cursor right
RETURN	Enter line

NanoEditor

KEY	FUNCTION
ESCAPE	Escape
Edit Mode:	
BACKSPACE	Backspace
DELETE	Delete character
TAB	Tab
←	Cursor left.
→	Cursor right
ENTER	Enter line

CONTROL CAPSULE

NanoEdtior

KEY	FUNCTION
FTCN 9	Escape
Edit Mode:	
FCTN 1	Delete
FCTN 3	Erase line
FCTN 7	Tab.
FCTN S	Cursor left
FCTN D	Cursor right
ENTER	Enter line

the Print option, just select it from the main menu (3). After you save (and print) the source file, select the Exit option from the main menu. The program gives you a chance to change your mind before ending, so you don't need to worry about losing the program in memory due to an erroneous keypress.

Now it is time to load and RUN the *NanoAssembler*. The program prompts you to load your source file for assembly. As the program translates your source code into machine code, it lists the source file, the addresses, and object code to either the screen or a printer (if you have one).

Passing Through

The actual assembly of the program occurs in two steps, or "passes." Thus, the *NanoAssembler* is a "two-pass" assembler. The first pass does most of the work, determining the correct machine-language instructions and the instruction addresses. However, sorting out labels requires a second pass because, until it identifies all address labels, the program may not know the exact address of each instruction.

Try assembling Sample Program 1. If you have entered it correctly, the *NanoAssembler* should output the assembled version, as shown in Figure 1, to the screen or printer. If you have made an error in entering the program into the *NanoEditor*, the *NanoAssembler* informs you of the line number in the source code that contains the error, and states the type of error. For example, if in line 1 you enter LDA #3 instead of LDA# 3, when you try to assemble the program the computer displays the error: ILLEGAL USE OF LABEL IN LINE 1. Here, the computer interprets the code as a LoaD A *addr* instruction (object code = 2), instead of a LoaD A immediate instruction (object code = 1). Then, when the computer evaluates the "label" #3, it finds that the label is illegal because it does not begin with a letter.

Figure 1: Instruction Set

Dec.	Binary	Nibbles per instr.	Mnemonic	Flags* affected C Z	Function
0	%0000	1	ADD	Y Y	Add the contents of B register to the contents of A register—result in A
1	%0001	2	LDA#	N Y	Load A with number following instruction.
2	%0010	3	LDA *addr*	N Y	Load A with number at location specified by *addr*.
3	%0011	3	STA *addr*	N N	Store the contents of A at location specified by *addr*.
4	%0100	1	TAB	N N	Transfer contents of A to B
5	%0101	1	TBA	N Y	Transfer contents of B to A
6	%0110	1	RRC	Y Y	Rotate A right one bit through carry
7	%0111	1	RLC	Y Y	Rotate A left one bit through carry
8	%1000	1	AND	Y Y	Logically AND A and B—Result in A
9	%1001	1	OR	Y Y	Logically OR A and B—Result in A
10	%1010	1	XOR	Y Y	Logically XOR A and B—Result in A.
11	%1011	3	BZ *addr*	N N	Branch to *addr* if Zero flag is set
12	%1100	3	BNZ *addr*	N N	Branch to *addr* if Zero flag is not set.
13	%1101	3	BCS *addr*	N N	Branch to *addr* if Carry flag is set
14	%1110	3	BCC *addr*	N N	Branch to *addr* if Carry flag is not set.
15	%1111	3	JMP *addr*	N N	Branch to *addr* unconditionally
Assembler Directives.					
n/a	n/a	0	ORG	n/a	Use to specify a particular address (e g . specify starting address of program)
n/a	n/a	0	EQU	n/a	Equate label with value—assigns the value to the right of the EQU statement to the label to the left.
n/a	n/a	1	DN	n/a	Define Nibble—assigns the value to the right of the DN statement to the label at the left

**Flags affected refers to whether or not the instruction has any effect on the flags in the status register. The C column stands for the Carry flag (did the operation result in a carry being generated?), and the Z stands for the Zero flag (did the operation result in a zero?). A Y appears in the column if the flag is affected by the instruction. An N indicates the flag is not changed by the instruction.*

After displaying the program, *NanoAssembler* prompts you to save the object file. The saved file is identical in format to the ones you loaded and saved with the *NanoProcessor* last issue; that is, the file contains the contents of all addresses from 0 through 255. To see that your program works properly, load and RUN the *NanoProcessor*. You can then load and run the program you've just created according to the instructions detailed in Vol. 5, No. 5.

For a short program such as Sample Program 1, this process may seem a bit time consuming. For longer and more complex programs, however, the ease of writing and debugging provided by an assembler more than makes up for the added steps.

Assembler Directives

Figure 1 displays the 16 instructions that we detailed in the *NanoProcessor*. You may specify any of these instructions when writing an assembly-language program with the *NanoEditor*. The *NanoAssembler*, in turn, converts these instructions into their machine codes. There are three additional commands, known as assembler directives, that the *Assembler* understands:

Directive	Purpose
ORG	Start object code here
DN	Define a nibble
EQU	Define a label

The ORG command directs the *NanoAssembler* to assemble the program at a specified address between 0 and 255. For an example of this instruction, see line 1 of Sample Program 2. This program is a slightly modified version of Sample Program 2 that we presented in last issue's *NanoProcessor*. It performs a two nibble addition of numbers located at addresses 240 and 241, placing the answer in addresses 248 and 249. The ORG statement makes the starting address %1010.

The DN instruction allows you to include a particular value at any address. Just specify the address using the ORG directive, and then define the value to be placed at that address with the DN directive. Lines 22 through 24 of Sample Program 2 define the two nibbles that the program adds.

Figure 2

Decimal	Binary	Hexadecimal
0	%0000	$0
1	%0001	$1
2	%0010	$2
3	%0011	$3
4	%0100	$4
5	%0101	$5
6	%0110	$6
7	%0111	$7
8	%1000	$8
9	%1001	$9
10	%1010	$A
11	%1011	$B
12	%1100	$C
13	%1101	$D
14	%1110	$E
15	%1111	$F

Sample Program 1

Addr	Code	Line	Label	Mnemonic	Remark
		001		LDA# 3	;Get first number
0	%0001				
1	%0011				
		002		TAB	;Move to B
2	%0100				
		003		LDA# 7	;Get second number
3	%0001				
4	%0111				
		004		ADD	;Figure sum
5	%0000				
		005	HERE	JMP HERE	;Jump self to stop
6	%1111				
7	%0110				
8	%0000				

The EQU command lets you identify any address with a particular label. Lines 2 through 6 of Sample Program 2 use this directive. These statements make Sample Program 2 more readable by assigning descriptive labels to the 5 data addresses: NIB1 and NIB2 for the two numbers to be added; LONIB and HINIB for the low and high nibbles of the answer; and OUT for the OUT light. (See last issue's *NanoProcessor* for a complete explanation of how Program 2 uses these 4 locations.)

The other change to Program 2 in this issue is in the use of the OUT light located at the upper-left of the *NanoProcessor* screen. When you assemble Sample Program 2 and run it, you will find that the OUT light is off when the program begins, but it turns on when the program is complete. Thus, you do not need to know what address the program will end on. Instead, the OUT light signals that the program is finished.

Sample Program 3 accesses the *NanoProcessor*'s "sound chip." Any time you store a number at either location 254 or 255, the *NanoProcessor* responds with a tone. With 16 different values possible at each of these locations, you can make a total of 32 different tones. Sample Program 3 plays a C scale.

We hope that you have found these *Nano* programs instructive and enjoyable. With what you have learned, you should be able to create your own "machine-language" routines. Feel free to let us know in "Letters to the Editor" of any programs you create, so we may share them with our readers.

HCM Glossary Terms: assembler, label, object code, op-code, operand, pass, source code.

For your type-in listings, see HCM PROGRAM LISTING CONTENTS.

HCM

Three Number Systems Supported

Machine language on the *NanoProcessor* can be entered only in binary. The *NanoAssembler*, however, understands decimal and hexadecimal in addition to binary. Last issue we explained how to convert between decimal and binary—this issue we introduce you to hexadecimal.

As we explained in the previous issue, decimal is a base 10 system. It uses ten digits (0 through 9) to represent numbers. Similarly, binary is a base 2 system and uses two digits (0 and 1). Hexadecimal is a base 16 number system and uses 16 different digits—0 through 9 plus the letters A through F. (See Figure 2 for a conversion chart.) As the conversion chart shows, we can express the number 11 decimal as either the binary number %1010 or the hexadecimal number $B. (Note that the % symbol denotes a binary number, and the $ symbol a hexadecimal number.)

To convert a two-digit hexadecimal number (say $C8) to decimal you simply find the decimal equivalent of the left-most digit (i.e., $C = 12), and multiply it by 16. Then simply add the decimal equivalent of the right-most digit (12*16 + 8 = 200). Hexadecimal is a particularly useful system in assembly language because it can express any nibble as a single character or any byte as two characters.

Sample Program 2

```
Addr  Code     Line  Label  Mnemonic   Remark
                001          ORG 10
                002   NIB1   EQU $F0
                003   NIB2   EQU $F1
                004   LONIB  EQU $F8
                005   HINIB  EQU $F9
                006   OUT    EQU $FD
                007          LDA# 0     ;Turn OUT light off
 10   %0001
 11   %0000
                008          STA OUT
 12   %0011
 13   %1101
 14   %1111
                009          LDA NIB1   ;Get first number
 15   %0010
 16   %0000
 17   %1111
                010          TAB        ;Move to B
 18   %0100
                011          LDA NIB2   ;Get second number
 19   %0010
 20   %0001
 21   %1111
                012          ADD        ;Figure sum
 22   %0000
                013          STA LONIB  ;Low to memory
 23   %0011
 24   %1000
 25   %1111
                014          BCC NIB    ;One nibble answer
 26   %1110
 27   %0010
 28   %0010
                015          LDA# 1
 29   %0001
 30   %0001
                016          JMP STHI   ;All done
 31   %1111
 32   %0100
 33   %0010
                017   NIB    LDA# 0     ;Zero A
 34   %0001
 35   %0000
                018   STHI   STA HINIB  ;HIgh to memory
 36   %0011
 37   %1001
 38   %1111
                019          LDA# ON    ;Set OUT light
 39   %0001
 40   %0001
                020          STA OUT
 41   %0011
 42   %1101
 43   %1111
                021   HERE   JMP HERE   ;Jump self to end
 44   %1111
 45   %1100
 46   %0010
                022          ORG $F0
                023          DN $A
                024          DN $C
```

Sample Program 3

```
Addr  Code     Line  Label  Mnemonic   Remark
                001   SOUND  EQU 254
                002          LDA# 2
  0   %0001
  1   %0010
                003          TAB
  2   %0100
                004          AND
  3   %1000
                005          RRC
  4   %0110
                006          STA SOUND
  5   %0011
  6   %1110
  7   %1111
                007          ADD
  8   %0000
                008          STA SOUND
  9   %0011
 10   %1110
 11   %1111
                009          ADD
 12   %0000
                010          STA SOUND
 13   %0011
 14   %1110
 15   %1111
                011          LDA# 6
 16   %0001
 17   %0110
                012          STA SOUND
 18   %0011
 19   %1110
 20   %1111
                013          ADD
 21   %0000
                014          STA SOUND
 22   %0011
 23   %1110
 24   %1111
                015          ADD
 25   %0000
                016          STA SOUND
 26   %0011
 27   %1110
 28   %1111
                017          ADD
 29   %0000
                018          STA SOUND
 30   %0011
 31   %1110
 32   %1111
                019          LDA# $0
 33   %0001
 34   %1101
                020          STA SOUND
 35   %1101
 36   %1110
 37   %1111
                021   HERE   JMP HERE
 38   %1111
 39   %0110
 40   %0010
```

HCM Review Criteria

Each month, *Home Computer Magazine* (*HCM*) reviews products designed for the Apple II family, Atari 800 family and compatibles, Commodore 64, IBM PC and PCjr, and Texas Instruments 99/4A computers. *HCM* reviews take a detailed look at the quality, utility, and value of commercially available packages for these machines. Because our publishing charter forbids accepting outside advertising, we strive to make the scope and content of our review pages shine with a unique blend of humanistic frankness and objectivity.

Not only will you find all relevant information for making a wise purchase decision, but in some special cases we also provide nuggets of compu-prestidigitation.* For example, we frequently include essential documentation not furnished by the manufacturer. Additionally, each issue of *HCM* tries to review at least one outstanding product—a "Diamond in the Rough"—which, because of company size, marketing clout, or for some other reason, has not received the attention it deserves.

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

HCM Review

Name: Old Art
Program Type: Recycled Graphics
Machine: Apple II family, Atari 800XL, C-64 IBM PC & PCjr, TI-99/4A
Distributor: Hit 'n' RUN Software, Inc.
Price: $99.99 (or trade for '72 Pinto)

System Requirements:
Disk Drive, Joystick, Trash Can optional

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

Products may also be evaluated in the following areas:

*** Flexibility—**
Can the product be adapted to the specific needs of the users?

*** Cost/Benefit—**
Is the product worth the user's investment in time and money?

*** Necessity—**
Is the product a solution for which a problem already exists?

*** Originality—**
Is it unique in concept, or simply a "me too" product?

*** Longevity—**
The "Boredom Factor." Does the program sustain interest?

*** Rewards—**
Are the audio-visual rewards motivating and appropriate?

*** Concept Presentation—**
Are the concepts presented clearly, logically, and in depth?

*** Special Effects—**
How does quality of sound and visual effects rate? Do they enhance or detract from the product or learning process?

*** Performance—**
How well the product performs as intended; how well it takes advantage of a specific machine's capabilities; how well it responds to the user's commands; how effectively the graphics, sound effects, music, or speech are integrated with the software.

*** Engrossment—**
Whether the game or activity has that intangible quality that holds players on the edge of their seats while the hours tick by unnoticed.

OR

*** Ease of Use—**
The degree to which a user can interact with the product 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.

OR

*** Ease of Set-up—**
How well the product design facilitates easy installation.

*** Documentation—**
The quality of the printed matter that comes with the product; whether the instructions are clear and comprehensive; whether the machine configuration requirements are spelled out. Information such as how to load a program, use the keyboard, and restart an activity contributes to the documentation rating, as do tips on performance peculiarities.

Attention Software Authors & Peripheral Inventors:

* WANT TO BE DISCOVERED? *

Home Computer Magazine Wants To Give You A Chance!

We are looking for home computer products that have not received the attention they deserve. Each month, we will be singling out one such package for special review. If you have a unique commercial product of exceptional quality—but your advertising and promotion budget has not allowed you to capture major media attention—we want to see it. We will consider reviewing any product that meets our high standards.

We are an Equal Opportunity Reviewer!

In order to qualify for possible review, your product must:

1. Currently be available for purchase to readers of this magazine.
2. Make a unique and important contribution to the home computer industry.
3. Be of outstanding merit, quality, and value.
4. Be consistent with the type of machines and products we normally cover.

If you feel that your product qualifies, mail it to:

Home Computer Magazine
Attn: Editorial Submissions
1500 Valley River Drive, Suite 250
Eugene, OR. 97401

We reserve the right *not* to reply to each inquiry, so please do *not* contact us except to request return of your product. If you want your product to be returned, please include sufficient return postage.

***Compu-prestidigitation** (kóm´• pū • prĕs´• teh • dĭ • jeh • tā´• shŭn) —*n* 1. The magical quality of unexpected comprehension that results from presenting technical information about computers in a lively, entertaining, visually attractive and easy-to-understand format. 2. The magical tricks that make a computer sing, dance, and do all sorts of wonderfully useful things.

HCM Review

Name:	File Directory
Program Type:	Utility
Machines:	TI-99/4A
Distributor:	John Bulakowski 162 Lorann Drive Naugatuck, CT 06770
Price:	$29.95
System Requirements:	32K memory expansion, disk drive

	Poor	Fair	Good	Excellent
Performance:			■	
Ease of Use:				■
Documentation:			■	
Cost/Benefit:			■	

FILE DIRECTORY

A Review by Steve Nelson
HCM Staff

This program won't turn your house into a post office, but it could make your monthly mailing chores a little bit easier.

John Bulakowski's *File Directory* is a single-purpose program, designed simply to store, access, and print mailing labels. Small, narrowly-focused programs can, by their very simplicity, make specific tasks easier—eliminating unnecessary distractions. But, in its simplicity, can *File Directory* enable you to perform your mailroom chores with ease?

File Directory requires a disk drive, Extended Basic, and 32K memory expansion. A single disk is capable of maintaining 6 files, each with up to 100 records. Each record consists of 4 fields: name; street address; city, state, and zip code; and telephone number.

Entering Records

File Directory boots automatically when you select the Extended BASIC option. Once you select a file to work on, you can create a new record or change existing records. To create a new record, enter all of the data (name, address, etc.) into the 4 fields. The program automatically saves the information for you when you exit the program or begin working on another record. The program can also list all of a file's records—or just the names—either to the screen or to a printer.

File Directory lets you alter selected portions of your records. Using the [FCTN D] or [FCTN S] options, you can change a single word or letter without altering any other part of the record. Unfortunately, the first time you try to delete an entire record, you may experience some difficulty because there is a discrepancy between the instructions in the documentation and the prompt on screen. The on-screen prompt directs you to DELETE RECORD WITH ERASE. The documentation reads "If you want to delete this record, enter FCTN 1 (DEL) and input ENTER." This is an obvious error. You *can* delete the record using [FCTN 1], but to do so, you must hold the key down until it deletes all of the letters in the top line of the first field, then press [ENTER]. On the other hand, the erase function on the TI-99/4A—[FCTN 3]—deletes files immediately and with ease.

This error detracts from the overall quality of the documentation, which, though extensive, is sometimes poorly-worded and often confusing.

Printing Possibilities

File Directory prints labels and envelopes to specific preset tab locations, although you can manipulate them if you need to. *File Directory* is written for use with a parallel printer. But if you are using a serial printer, the documentation provides instructions on editing the program to allow you to use it. The program works with any printer that is compatible with the TI-99/4A computer and that can recognize the program's imbedded commands.

The program contains two printing options: Exception and Selection. Printing by Exception allows you to receive a printout in three possible typestyles: correspondence quality, standard, or expanded. Correspondence quality is crisper than standard quality, and expanded is a blown-up version of the typestyle used with correspondence quality. In the Exception mode, *File Directory* prints all files that you do not "except." After you select the option, the program prompts you for deletions. Use the [FCTN 3] option to delete the records that you don't want printed, and select a typestyle. The program then prints out all but the records you have excepted, without deleting any records from disk. You can print an entire file simply by making no "exceptions."

The Selection option offers you only two typestyles: correspondence quality and standard. The program prompts you for two record numbers, then prints the records you specify. Each time it completes printing, the program prompts you for two more numbers and repeats the procedure until you finish. Unfortunately, this printing method has the unnecessary drawback of tying you to the keyboard.

You can find phone numbers with ease, using the phone-number option. Enter the name (or partial name) of the person whose phone number you want, and *File Directory* searches through the records and prints the number or displays it on the screen.

Finally, *File Directory* has the handy feature of sorting your records alphabetically by last name. The program sorts by looking for the space before the last name in each record. This means that if you enter a name like De Silva and leave a space before Silva, the program sorts Silva, not De Silva. The sorting process is also somewhat slow, to say the least—taking about 30 minutes for 100 records!

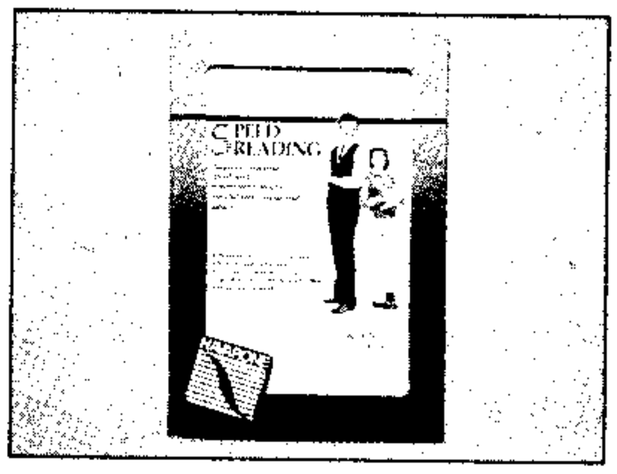
Considering *File Directory*'s poor documentation and extrememly slow response time, many TI-99/4A owners might prefer their old address-filing method—scratching and erasing in an old, worn-out address book. But for those of you with more than a little time and patience to spare, *File Directory* is a safe, tidy way to organize and print all of your important addresses.

HCM

Speed Reading

A Review by Steve Nelson
HCM Staff

HCM Review



Name:	Speed Reading
Program Type:	Utility
Machines:	TI-99/4A
Distributor:	Navarone Industries, Inc. 510 Lawrence Expressway #180 Sunnyvale, CA 94086
Price:	$29.95
System Requirements:	32K memory expansion (Disk drive recommended)

	Poor	Fair	Good	Excellent
Performance:			■	
Ease of Use:				■
Documentation:	■			
Cost/Benefit:			■	

Speed reading—some people swear it works, others swear because it doesn't work for them. Either way, Navarone's reading course still needs more work.

The last time I checked, the jury was still out, trying to decide if speed reading courses really work. Some experts say they do, some say they don't. From my own experience with such courses, I think they can help, but I'm not sure how long the effects will last. It seems that once you attain a certain reading speed, you must continue to read at that speed on a regular basis. If you don't, your reading speed will slowly diminish to the level you started from. Be that as it may, it's up to you to decide if you want to spend $29.95 to find out if *Speed Reading* works for you.

Speed Reading by Navarone, is a cartridge-based reading course for the TI-99/4A. The program comes in two versions: one for adults and one for children. Except for differences in the skill level of the reading selections, both versions are identical.

Speed Reading is easy to use. The program simply highlights the portions of the reading material that you are to read. The highlighting moves along the text at the wpm (words per minute) pace you select. The screen defaults to a light blue, with the entire text faintly visible in the background, and the emphasized portion in white. You can adjust the screen colors to the combination that appears sharpest to you.

According to the manual, there are three basic elements that determine your reading speed: eye movement—the rate at which you move your eyes from one group of letters to another; eye span—the number of letters you take in with each fixation, or pause; and "perception"—the duration of each fixation. *Speed Reading* offers three exercises designed to enhance your abilites in these areas.

Exercise For The Eyes

The first exercise focuses on "eye fixation." This tricky little routine is supposed to help develop peripheral vision by flashing two sets of letters on each side of a dot in the center of the screen; first two letters, then four, and finally six—at three different speeds. You must stare at the dot, read the letters using your peripheral vision, and retain them in your memory long enough to duplicate them with the keyboard.

The second lesson exercises eye movement. After you enter your reading rate (in wpm), the computer displays the text faintly on the screen and highlights groups of words for you to read in a zigzag pattern, across and down the page. Each time the program highlights a new set of words, it directs the computer to sound a beep.

The third exercise is a column-reading drill. It is similar to the eye movement lesson in format; but instead of zigzagging, it displays the text down the center of the screen. You set the wpm rate and the width of the line of highlighted text (from 16 to 26 characters).

Speed Reading provides practice text, however you can load your own text file, as long as it is an ASCII file saved in display/variable 80 format, and the text does not exceed 100 lines.

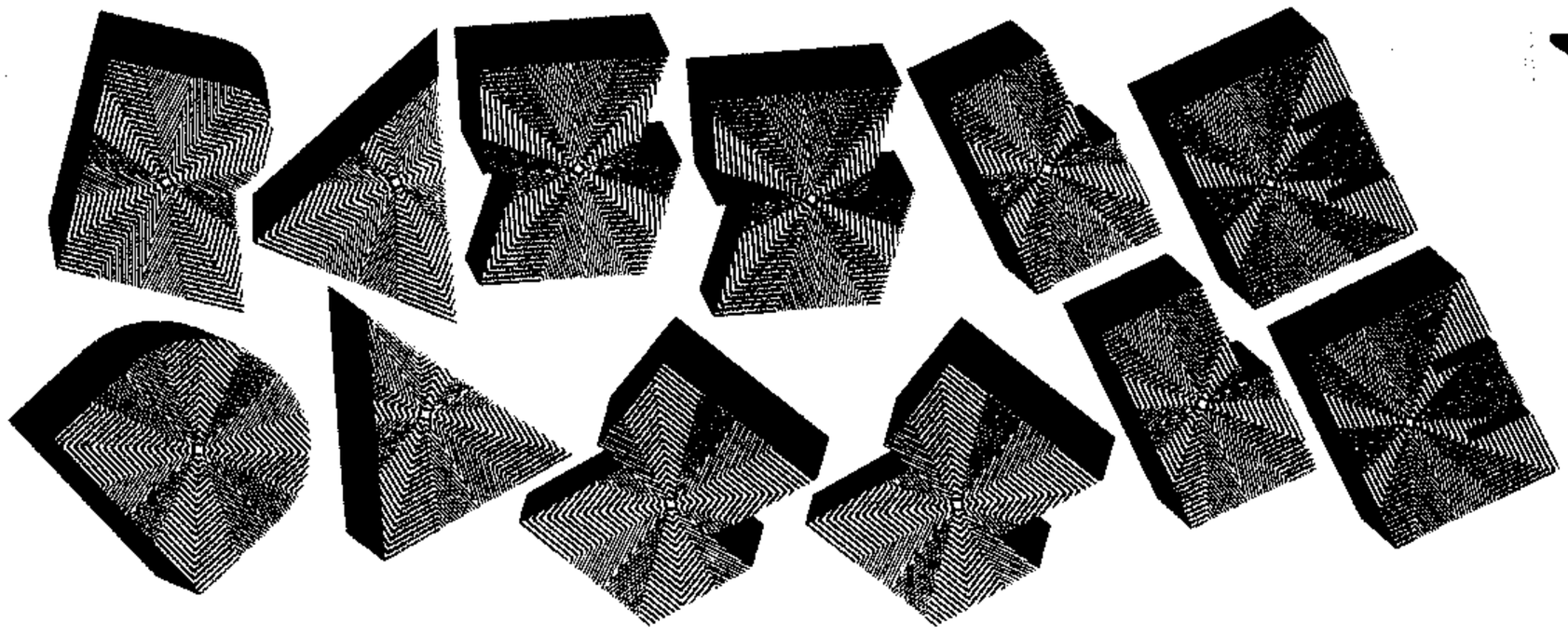
To help you pace yourself as you read from a source other than the computer, the program supplies a reading pacer. You enter the speed you want to practice with, and the program beeps out a steady rhythm to help you maintain it.

The manual provides practice sessions that incorporate the eye fixation and eye movement exercises. You work with these for 15 minutes each, then take a comprehension test. The final practice session has you using the reading pacer option as you read from a book or magazine.

Is It Worth It?

Although *Speed Reading* is very easy to use, it does have a few problems that I want to point out. First, the manual seems hastily written, with lots of typos and vague sentences. Its explanation of how people read and descriptions of exercises are ambiguous. For example, the eye fixation exercise limits you to 6 letters, but the manual implies that you can practice with 8. Navarone should have maximized its program to 8 or even 10 letters—I found it just too easy to read and retain 6 letters. The eye movement exercise limits you to only 100 lines of text, so you end up re-reading the same text over and over, or stopping your exercise every few minutes to load a new file. And finally, I was a bit put off by the general "it's all up to you" tone of the manual. People who really need to improve their reading speed require more guidance than the program's documentation provides.

What really bothers me about *Speed Reading*, however, is the apparent lack of thoroughness on the part of the developers. It left me feeling that the entire package was prepared in a somewhat perfunctory fashion. It could really stand some improvement, starting with a new manual. With a little more work, Navarone could possibly turn *Speed Reading* into a more useful and much friendlier program. But until this happens, you might want to spend your money on more conventional and perhaps even more thorough speed-reading courses. **HCM**



TI VENTRILOQUIST

by Scott Williams
HCM Staff

Tune-in to intriguing sound effects with your TI computer and a few dazzling routines . . .

What pops into your mind when someone mentions computer simulations or arcade games? If you're like most people, you probably associate these words with striking graphics animation or the game scenario itself.

But there is one very important feature built into most games and simulations that we tend to overlook, at least until we have to do without it—*sound effects*. Try turning down the volume on your television or monitor, then play all your favorite games. You won't find them nearly as interesting.

The TI-99/4A is a great machine for sound effects because of its complex 4-voice sound chip, the TMS 9919. This chip is capable of producing up to three tones and one noise at the same time. You can choose from 8 different types of noises and a wide range of tones as the example programs on these two pages demonstrate. These programs are short enough for you to enter each one in just a few minutes, then sit back and let your computer take you into the world of special effects.

Buzzy Bees

Program 1 incorporates one tone and the noise channel to approximate the sound of a swarm of bees. The first parameter in the **CALL SOUND** statement in line 200 sets the duration of the sound in milliseconds. We use a negative number to set the duration. This causes the sound to continue until the next **CALL SOUND** statement. When it encounters the next **CALL SOUND** statement, the computer simply leaves the sound on and changes the tone. This results in a smooth transition between tones and a very realistic honey bee.

Gun Fight

Program 2 simulates a gun shot repeated at random intervals. The noise channel plays a big part in this sound effect.

The TI computer responds to 8 noise parameters, each designating a different noise. To specify a noise, you need to use a negative value between −1 and −8 as a frequency parameter. Value −5 is the closest approximation to white noise—random frequencies dispersed evenly throughout the audio spectrum. Values −6 through −8 also produce noise—however, the lower frequencies contain more energy, which results in a lower pitched noise. Values −1 through −4 create what the TI BASIC Manual refers to as "periodic noise." Technically speaking, this periodic noise is not a noise at all. It more nearly resembles the sound of a narrow pulse wave. If you specify a value of −4 or −8, the frequency setting of the third tone in the **CALL SOUND** statement determines the pitch of the noise. This reference chart displays the relative pitches of the various types of noise available on the TI sound chip:

−1 or −5 = high pitch
−2 or −6 = medium pitch
−3 or −7 = low pitch
−4 or −8 = set by frequency of the third tone

Motor Rev

By mixing several sound effects together we can create a more complex simulation. Program 3 produces a very familiar sound: a car having difficulty starting up. The car motor turns over 8 times, then starts and slowly revs up. The engine holds a steady RPM for a few seconds then slows until it stops.

A **FOR-NEXT** loop in lines 110 through 140 handles the initial turning over of the engine. We use a noise setting of −8 with the third tone controlling the pitch of the noise. The loop counter then alters the frequency of the third tone, making the noise's pitch change. In the first **CALL SOUND** statement, the volume is at its maximum level of 0. In the next sound statement we decrease the volume to 5 and alter the frequency of the third tone to drop the pitch slightly. This results in the whirring sound of the crank case turning over.

The second **FOR-NEXT** loop starts the pitch of the noise channel at a very low level so that you can hear only clicks. These clicks increase in frequency with an increase in the frequency of the third tone, providing the thrumming noise of an idling engine.

The next loop holds the idle tone for a few seconds. The final **FOR-NEXT** loop takes the pitch of the noise from its high "idle" back to a slow clicking.

Hiccup

Program 4 incorporates two noise types and all three tones to produce the type of sound you might expect to hear if your computer had the hiccups. The first sound you hear is a high-pitched "hic." Line 110 generates this sound by setting the noise channel to −8 and setting the third tone at a frequency of 20000 Hertz (Hz). Voice two is used at a low volume to add a little tone color to the hic.

The second sound that you hear is a lower frequency. To achieve this effect, we specify the fixed-pitch noise −7 and mix it with three tones of 330, 337, and 380 Hz.

We have specified a positive duration to create a distinct break between notes. The sound lasts for the entire length of the duration, then ceases completely before the next note is played.

Siren

A police siren is a surprisingly complex sound to simulate. The sound, as first heard from a distance, is quiet. As the police car approaches, the siren becomes louder. Then the car passes, and the siren becomes quieter, fading into the distance. This variation in amplitude is fairly easy to reproduce on the computer. The complexity lies in a simultaneous oscillation in frequency.

To produce this sound effect in Program 5, we need to control two parameters: the frequency of the siren as the sound fluctuates up and down, and the volume of the siren as it approaches and recedes.

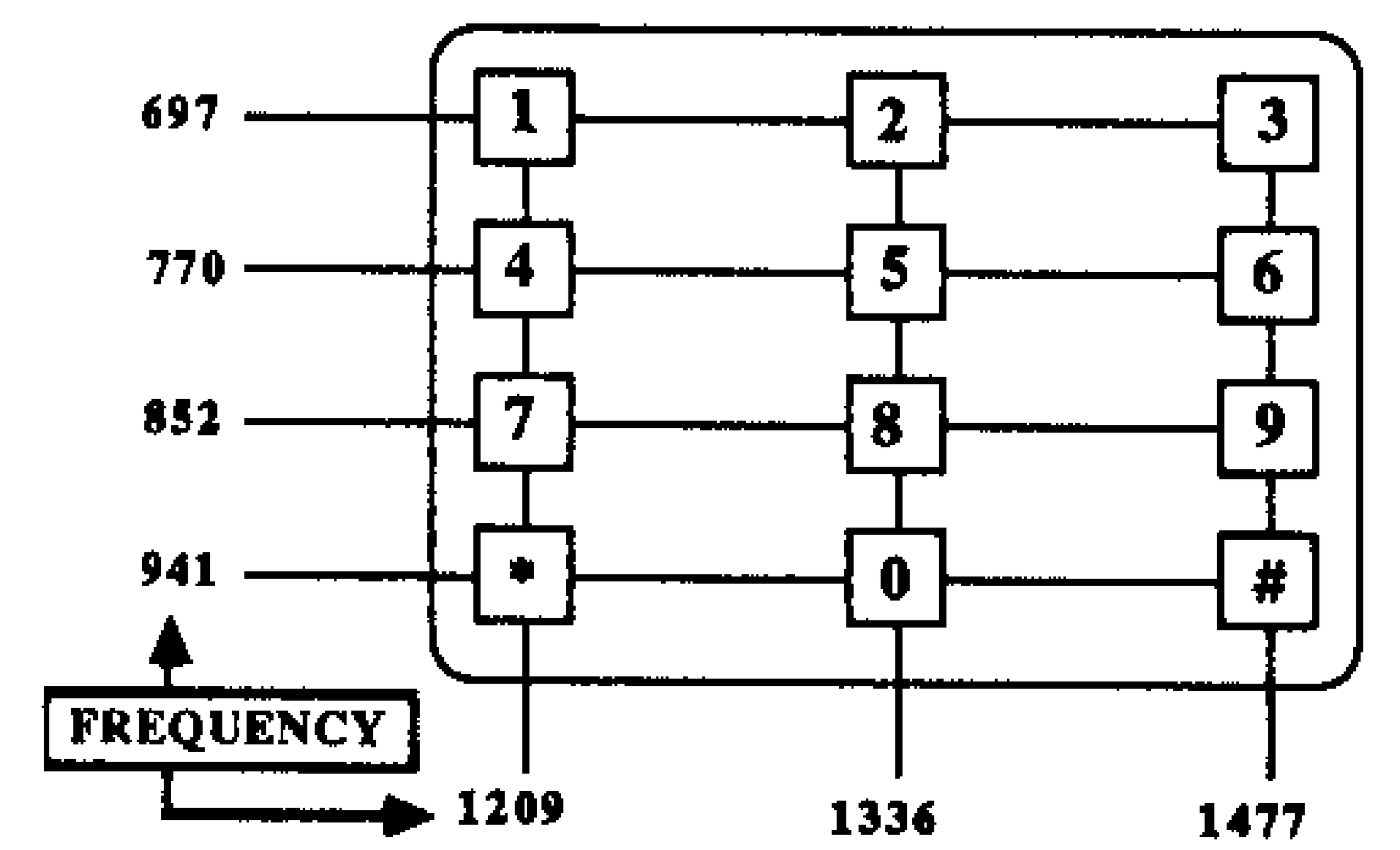
The simplest way to achieve this sound effect is to place the frequency oscillation aspect in a subroutine located in lines 250 through 280. The main program then cycles through two FOR-NEXT loops to control the increase and decrease in volume, calling the subroutine to play the siren.

In nature, few things are as perfectly linear as FOR-NEXT loops. So, for the sake of realism, we added a further dimension to the routine. We placed both the volume and pitch under the control of the SIN function. Unfortunately the SIN function is not very fast. We have therefore eliminated it from the most repetitive areas of the program, placing a list of pre-calculated SIN values in an array to save processing time later. We also have calculated the SIN of Z before calling the subroutine, rather than calculating it through every iteration of the Y loop.

Touch Tones

Have you ever wondered what would happen if you sent simulated touch tones to the mouthpiece of your phone? The answer: You would dial the number corresponding to the tones. The touch tones that dial your phone do not have to originate inside the phone; you can produce them externally with the same results.

Figure 1 TOUCH TONE PHONE BUTTON ARRANGEMENT



Touch tones are easy for your computer to generate. Each touch tone actually consists of a combination of two separate tones. One tone is determined by the row that the key lies in; the other is determined by the column (see Figure 1).

Program 6 generates random phone numbers, playing the accompanying tones. Do not hold your phone up to the speaker of your TV set while this program is running—you may end up with an expanded phone bill!

HCM Glossary terms: iteration, linear, noise channel, periodic noise, white noise.

HCM

Program Listing 1: *BUZZY BEES*

```
P 100 REM BUZZY BEES
L 110 I=INT(RND*10+1)
R 120 J=INT(RND*10+1)
L 130 FOR V=I TO J STEP -1*(J>I)+1*(I>=J)
A 140 GOSUB 180
L 150 NEXT V
T 160 I=J
A 170 GOTO 120
Q 180 F=INT(RND*2)+115
L 190 SN=SGN(RND-.5)
M 200 CALL SOUND(-500,-3,10,F+(SN),V)
L 210 RETURN
```

Program Listing 2: *GUN FIGHT*

```
P 100 REM GUN FIGHT
J 110 CALL SOUND(100,110,0,130,5,34000,30,-8,0)
V 120 FOR Z=0 TO 30 STEP 15
F 130 CALL SOUND(-100,110,30,110,30,3400,30,-8,Z)
O 140 NEXT Z
Q 150 FOR Z=1 TO INT(RND*5)*30
A 160 NEXT Z
R 170 GOTO 110
```

Program Listing 3: *MOTOR REV*

```
P 100 REM MOTOR REV
G 110 FOR Z=1 TO 8
B 120 CALL SOUND(-100,110,30,110,30,2800+Z*50,30,-8,0)
Y 130 CALL SOUND(-100,110,30,110,30,2700+Z*50,30,-8,5)
O 140 NEXT Z
N 150 FOR Z=300 TO 3000 STEP 60
A 160 CALL SOUND(-100,110,10,120,10,Z,30,-8,0)
J 170 NEXT Z
A 180 FOR TD=1 TO 70
P 190 CALL SOUND(-100,110,10,120,10,Z,30,-8,0)
B 200 NEXT TD
H 210 FOR Z=3000 TO 200 STEP -100
N 220 CALL SOUND(-100,110,10,120,10,Z,30,-8,0)
S 230 NEXT Z
```

Program Listing 4: *HICCUP*

```
P 100 REM HICCUP
S 110 CALL SOUND(40,110,30,440,10,20000,30,-8,0)
M 120 FOR TD=1 TO 25
P 130 NEXT TD
P 140 CALL SOUND(30,330,0,337,0,380,10,-7,0)
I 150 CALL SCREEN(1)
N 160 CALL SCREEN(16)
V 170 FOR TD=1 TO RND*500+300
O 180 NEXT TD
L 190 GOTO 110
```

Program Listing 5: *SIREN*

```
P 100 REM SIREN
Q 110 DIM S(40)
E 120 FOR X=0 TO 3.14 STEP .079
L 130 C=C+1
G 140 S(C)=SIN(X)
L 150 NEXT X
M 160 FOR Z=1.57 TO 0 STEP -.314
F 170 Z1=SIN(Z)
A 180 GOSUB 250
H 190 NEXT Z
Z 200 FOR Z=.2 TO 1.57 STEP .5
K 210 Z1=SIN(Z)
D 220 GOSUB 250
S 230 NEXT Z
R 240 END
U 250 FOR Y=1 TO 40
G 260 CALL SOUND(-100,S(Y)*200+440,Z1*30)
H 270 NEXT Y
A 280 RETURN
```

Program Listing 6: *TOUCH TONES*

```
P 100 REM TOUCH TONES
T 110 RANDOMIZE
U 120 READ T1(1),T1(2),T1(3),T1(4),T2(1),T2(2),T2(3)
L 130 DATA 697,770,852,941,1209,1336,1477
S 140 FOR Z=1 TO 3
Q 150 GOSUB 330
I 160 PRINT C$;
Y 170 CALL SOUND(200,T1(S1),0,T2(S2),0)
H 180 CALL SOUND(70,110,30)
H 190 NEXT Z
Z 200 PRINT "-";
P 210 FOR TD=1 TO 60
D 220 NEXT TD
N 230 FOR Z=1 TO 4
E 240 GOSUB 330
W 250 PRINT C$;
M 260 CALL SOUND(200,T1(S1),0,T2(S2),0)
V 270 CALL SOUND(70,110,30)
B 280 NEXT Z
E 290 PRINT :::::
W 300 FOR TD=1 TO 500
R 310 NEXT TD
Z 320 GOTO 140
N 330 S1=INT(RND*4)+1
K 340 S2=INT(RND*3)+1
A 350 IF (S1=4)*((S2=1)+(S2=3))THEN 330
Q 360 C$=SEG$("123456789*0#",(S1-1)*3+S2,1)
G 370 RETURN
```

Several Flags In One Variable

Flags are an essential programming element. For most programs, you can keep each flag in a separate numeric variable. But as memory gets tight (a common occurrence if you are using a TI-99/4A without memory expansion, and your programming task requires many different flags), you may wish for an alternative to this method.

Well, here is a trick in TI Extended BASIC that lets you keep several flag values in just one numeric variable! This method has two key elements: First, because a flag is binary in nature (yes or no, on or off, etc.), you must interpret the numeric variable as a binary quantity. The other important part of this trick is using the TI Extended BASIC logical operators **OR** and **AND** to set, clear, and test (read) the various flag "bits" in a variable.

Before you can use one of the logical operators in this manner, you must first know the value of each bit within a binary number. The first chart below shows these values.

Bit number:	7	6	5	4	3	2	1	0
Bit value:	128	64	32	16	8	4	2	1
Bit:	1	1	1	1	1	1	1	1

For example, the decimal number 32 is a binary 0010 0000, and 34 is a binary 0010 0010. Therefore in an 8-bit quantity you have 8 different flags—one for each bit.

Let's see what effect logical operators can have on binary numbers. Both the **AND** and **OR** operators compare two binary numbers and produce a numeric result.

To set a flag, use the **OR** operator. You can set a particular bit within a numeric variable by **OR**ing that variable with the value of the bit to be set. The **OR** operator checks to see if a bit in the first number (the flag byte) is set or the corresponding bit in the second number (or mask) is set. If so, the **OR** operator sets that bit in the resulting number. The second figure shows the result of the operation **40 OR 12**.

Let's use the variable **FL** as an example. To set bit number 2 in **FL** do the following: **FL = FL OR 4**. To set bit number 3 do this: **FL = FL OR 8**.

Decimal	Binary								
40	0	0	1	0	1	0	0	0	Flag byte
12	0	0	0	0	1	1	0	0	Mask
44	0	0	1	0	1	1	0	0	Result

To clear a flag, use the **AND** operator. You can clear a particular bit within a numeric variable by **AND**ing that variable with everything *but* the value of the bit to be set. The **AND** operator compares two numbers; if a bit in the flag byte is set *and* the corresponding bit in the mask is also set, then the **AND** operator sets that bit in the resulting number. An 8-bit number holds a maximum value of 255—all bits set. To clear just the zero bit, execute the following: **FL = FL AND 254** (255 minus the value of the bit to be cleared). To clear bit number 3, do this: **FL = FL AND 247**. The third figure shows the result of the operation **40 AND 247**.

Decimal	Binary								
40	0	0	1	0	1	0	0	0	Flag byte
247	1	1	1	1	0	1	1	1	Mask
32	0	0	1	0	0	0	0	0	Result

Now that you can clear and set bits, how do you read them? You test a bit by **AND**ing it with the value of the bit that you want to read. To test if the zero bit of a flag is set or clear, do the following: **RESULT = FL AND 1**. To test bit number 3, execute this: **RESULT = FL AND 8**. If **RESULT** ends up equal to zero, the flag is clear. If **RESULT** is a nonzero number, the flag is set.

With the **AND** and **OR** operators, you can manipulate each bit separately. With this capability, you can use each bit within a numeric variable as an independent flag. For example, you can use the zero bit in a variable to flag whether output should go to the screen (bit is clear) or the printer (bit is set). Bit number 3 can determine whether the program will use sound effects or not.

Decimal	Binary								
40	0	0	1	0	1	0	0	0	Flag byte
8	0	0	0	0	1	0	0	0	Mask
8	0	0	0	0	1	0	0	0	Result

— Scott Williams

HCM Glossary terms: bit, byte, flag, logical operator, mask.

Problems In

With this new column and HCM productivity software, you can build solid decision-making tools.

A CHALLENGE TO OUR READERS

In past issues, HCM has published a number of productivity programs as tools for making calculated decisions. *The Organizer* for example, helps you prioritize and systematize your thoughts. And *Snap-Calc* lets you adapt a simple spreadsheet for your own applications—for example, tracking investments, comparing purchase options, calculating payroll, etc.

"Problems in Productivity" is a column designed to help you find new ways of using these productivity tools. Each issue, this column presents one "problem in productivity"—a real-life situation that cries out for a computer solution. You can solve all the problems by using one of our previously-published programs as a tool.

Next issue, we will publish our own solution to the problem presented here, together with a *new* problem. You can then compare our solution to your own.

This issue's productivity problem calls for *Snap-Calc*, published in the Vol. 4, No. 3 (August, 1984) issue. Your challenge is to use this simplified computer spreadsheet to answer a question many families face every year: Where can we find the college education to best fit our financial situation? In solving the problem, you will learn more about spreadsheets in general, and *Snap-Calc* in particular. [At present, there is no version of *Snap-Calc* for the Atari 800, 800XL, or 130 XE. We will, however, make this version available soon—Ed.]

Asking Questions

How *do* you plan a college education? What framework can you use to compare all the financial alternatives offered by all the different schools you're likely to consider? In Figures 3, 4, and 5, we give you a typical (but imaginary) choice between three different institutions. Although this data varies between colleges, all items reflect common expenses that any college student encounters. Your "assignment," should you choose to accept it, is to develop a spreadsheet template for comparing the three sets of data and computing a "bottom line" that indicates the best financial alternative. Next issue, we will publish our solution: a completed template that manipulates the data in Figures 3, 4, and 5.

Snap-Calc In Short

What on Earth is a spreadsheet template? Put simply, it's a framework of formulas used to study relationships between numeric "elements"—both constant and variable. In the old style of bookkeeping (handwritten), these mathematical comparisons were often confusing, time-consuming, and tedious. Every time one number changed, the bookkeeper had to refigure all the calculations and make corrections.

But with *Snap-Calc*, you can design a pre-set template of formulas and let the computer do most of the work. By entering raw data into the template, you can do a day's worth of bookkeeping in a few seconds. If one of your figures changes, you merely input the new number, and the computer recalculates all the formulas and complex relationships for you. This makes *Snap-Calc* a powerful "what if" tool, allowing for rapid comparison of many different alternatives. You can use it time and again to test different options or hypothetical situations.

Here's an example: Let's say you want to calculate the cost of your dream vacation this summer. Before you leave, create a template with *Snap-Calc* that enables you to list all your daily expenditures. Include columns for meals, hotels, plane tickets, rental cars, fuel, tips, and tours. You can then enter a formula that sums up these daily costs (either actual or estimated). Then by incorporating a formula that subtracts your expenses from your available funds, you can balance each day's expenditures.

You may use this template either to plan a vacation that fits your budget—by comparing all the alternatives—or to keep track of your daily expenses while traveling. Figure 1 shows the information needed for such a spreadsheet, and Photo 1 shows what the spreadsheet would look like on-screen.

For actual *Snap-Calc* operating instructions, see Vol. 4, No. 3 (August, 1984) of *HCM*. If you don't have *Snap-Calc* with all its updates, see "DeBugs on Display" Vol.

Figure 1

```
LOGIC NAME IS TRIPCOST
TOTAL COLUMN IS 10
LAST COLUMN IS 9
1 IS DATE
2 IS CAR RENTAL
3 IS GAS COST
4 IS GALLONS
5 IS ODOMETER START
6 IS ODOMETER STOP
7 IS MILES
7 = 6 - 5
8 IS MILEAGE
8 = 7 / 4
9 IS COST/MILE
9 = 3 / 7
10 IS BREAKFAST
11 IS LUNCH
12 IS DINNER
13 IS SNACKS
14 IS ENTERTAINMENT
15 IS TRIPS
16 IS HOTEL ROOM
17 IS AIR FARE
18 IS TAXI FARE
19 IS DAILY TOTAL
19 = 2 + 3 + 10 + 11 + 12 + 13 + 14
   + 15 + 16 + 17 + 18
20 IS RUN TOTAL
20 IS 19 + LAG 20
```

Photo 1

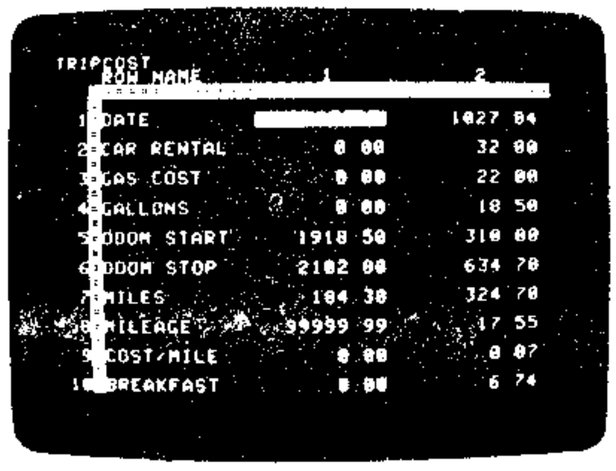


Figure 1 shows each "Tripcost" formula numbered by row. Photo 1 shows a typical spreadsheet with entries

Productivity

4, No. 5; Vol. 5, No. 1; Vol. 5, No. 2; and Vol. 5, No. 3; or order the Vol. 4, No. 3 ON DISK/ON TAPE. (Note: *Snap-Calc* on the TI-99/4A requires memory expansion.)

Now For The Challenge

As mentioned, this issue's productivity challenge is to create a spreadsheet template that allows a comparison between different college education alternatives. Here are some suggestions that will help you formulate your template: You will need gross income rows for the student, both parents, and income from other sources. (Gross income figures are used to analyze the student's qualifications and need for some types of student financial aid.) You'll need 4 rows for expendable income. Make expense categories for tuition, housing and meals, books and supplies, transportation, entertainment, clothing, and vacations.

You will need formulas that will sum the various quantities and subtract the expenses from the total income. (As you develop the spreadsheet, you will encounter the need for other formulas to create subtotals, etc. If you need extra help, turn the page upside down and read "Hints.") If the *variance*—the difference between income and expenses—is a positive number, the student can afford the particular college used in the parameters. But, if total financial resources fall short of expenses (a negative variance), the student must qualify for financial aid to make up the difference. Try entering different levels of financial aid. If you project that scholarships or grants would cover less than the variance, add in available savings. And if this still doesn't cover the costs, examine other types of loans that allow low payments while the student is still in school.

Keep in mind all the options available. You may even discover some options that we have not suggested. The example in Figure 5 assumes that the student starts at a community college, attends part-time, lives at home, and is working for funds to transfer to another institution. But in this case, transfer procedures may require 5 years—instead of 4—to complete the student's education. Also, as shown in Figure 4, if the student wants to attend an out-of-state school, additional costs for transportation and out-of-state tuition affect the total financial picture.

Work first with the sample parameters and numbers provided in Figures 2, 3, 4, and 5. (These will appear in our published "solution" next issue.) Then, you might try using other items and statistics you've accumulated from researching real institutions that interest you. Changing figures and parameters—and then letting the program recalculate the results—will show you how flexible and useful a spreadsheet like *Snap-Calc* really is.

HCM

Figure 2
Sample Annual Income Base

1st Year Income:	
Father's Gross	20,000
Mother's Gross	16,000
Student's Gross	1,500
Other	1,000
Father's Expendable	1,500
Mother's Expendable	1,500
Student's Expendable	500
Other	500
Savings Available	500
Loan costs	7.5%

Note: Income increases 10% per year.

Figure 3
College A
(4 Years In State)

1st Year Expenses:	
Tuition	1432.50
Meals & Housing	1098.00
Books & Supplies	300.00
Misc. Personal	990.00

Note: Expenses increase 5% per year

Figure 4
College B
(4 Years Out of State)

1st Year Expenses:	
Tuition	4057.50
Meals & Housing	2340.00
Books & Supplies	300.00
Misc. Personal	990.00

Note: Expenses increase 5% per year

Figure 5
College C
(2 Years Community College Then Transfer)

1st Year Expenses:	
Tuition	594.00
Meals & Housing	400.00
Books & Supplies	300.00
Misc. Personal	800.00

Note: Expenses increase 5% per year. For expenses after the transfer to either an in-state or out-of-state university (for 3 remaining years), see Figures 3 and 4.

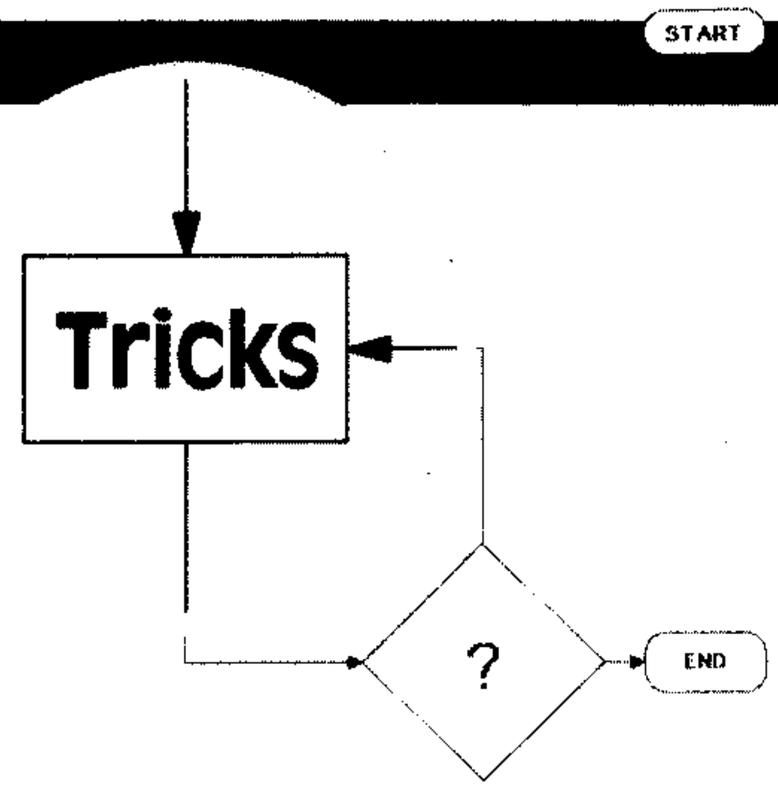
Hints

After familiarizing yourself with *Snap-Calc*, keep in mind these helpful hints: Use a large piece of paper to sketch in the positions of each formula and how they relate to each other. When you've finished the sketch, plug the formulas into *Snap-Calc* to make the spreadsheet template. Save it onto a disk or tape (Commodore and TI only) before loading it with data.

You will need to include formulas in the template to perform the following operations:

1. Total gross income
2. Add (10%) rise in income (Lag function)
3. Total expendable income
4. Add (10%) rise in income (Lag function)
5. Total expenses
6. Add (5%) inflation rate to expenses
7. Subtract total expenses from expendable income (VARIANCE 1)
8. Total financial aid
9. Add variance 1 to total aid (VARIANCE 2)
10. Add variance 1 to savings and conventional loan, then subtract your loan payment (VARIANCE 3)

NOTE: Your formula template may vary very from this list depending on your machines capabilities.



ASSEMBLING AN ALGORITHM

by the HCM Staff

WHAT IS AN ALGORITHM?

An algorithm is simply a procedure—one that a program uses to complete a task or solve a problem. Flow diagrams and flow charts are handy tools for representing the steps in this procedure. Any program can be viewed as a collection of separate procedures. In this column, we focus on and explain one unusual or interesting algorithm that is found in one of the programs we publish each issue.

From mnemonics to numbers: How does an assembler do it?

This issue's "Algorithm-A-Tricks" focuses on the *NanoAssembler*. Assemblers, by nature, require several complex algorithms, and the *NanoAssembler* is no exception. The main job of any assembler, however, is to identify mnemonics and translate them into their machine-language equivalent. Here we discuss how the *NanoAssembler* accomplishes this formidable task.

Using a series of **DATA** statements, the program **READ**s all the possible mnemonics into a string array— **OP$( )** for example. Each element of the array stores one mnemonic. (Note that because Atari BASIC does not support string arrays, we must simulate them. We do this by creating a single long string that contains all of the different mnemonics, and then use the Atari computer's string manipulation operator to extract the mnemonic we wish to check for. For the purposes of this column, we will continue to use arrays as our example, but Atari users should refer to line 1130 to see how the *NanoAssembler* simulates these arrays.)

The secret behind this technique is the order in which the program stores the mnemonics: according to their numeric machine-language equivalent (their op-code value). For example, the **ADD** mnemonic has an op-code value of 0, so **OP$(0)** = "**ADD**". Similarly, because the **STA** mnemonic has an op-code value of 3, **OP$(3)** = "**STA**". This conversion method applies to all 16 mnemonics.

Now, how do we use the **OP$( )** array to convert a *NanoEditor* source code listing into machine language? Fairly easily. Using a loop, the program compares the mnemonic field of a line with each element of the **OP$( )** array. The loop executes a maximum of 19 times—i.e., 16 times for the 16 mnemonics and 3 times for the three assembler directives (**ORG**, **EQU**, **DN**). If the program goes through the entire loop without finding a match, an ILLEGAL OP-CODE error results. If the program does find a match, the loop counter provides the op-code value of the mnemonic. When the program finds a **STA**, for example, the loop counter equals 3, the op-code value of a **STA**. Figure 1 shows a flow chart of this loop. Note that the variable names in this chart do not necessarily reflect the actual variable names in each version of *NanoAssembler*. We have chosen these names purely for explanatory purposes.

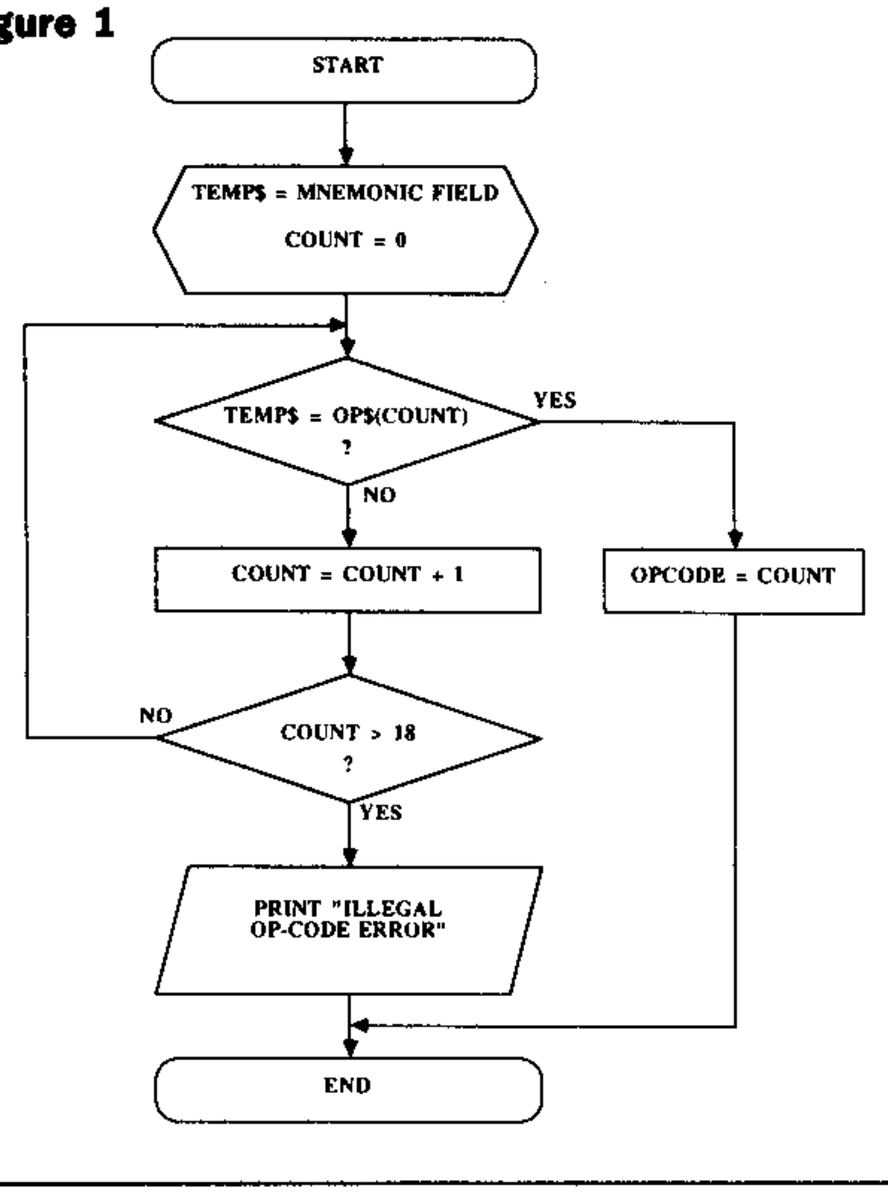
The Number Of Nibbles

Once the program has decoded a mnemonic into its op-code value, it must alot space for the op-code in memory. Not every op-code instruction demands the same number of nibbles. We therefore use another array—the **NN( )** array (Number of Nibbles) to keep track of this amount. Set up in the same order as the **OP$( )** array, this array stores the number of nibbles that each op-code instruction requires. For example, to find the number of nibbles that the **STA** instruction requires, you can print **NN(3)**, with 3 representing the op-code value of a **STA**.

HCM Glossary terms: assembler, mnemonic, nibble, op-code.

HCM

Figure 1



Here they are . . . the best of the one-line programs that we have received since printing the fifth "HCM One-Liners" column in *Home Computer Magazine* Vol. 5, No. 5. Although many interesting programs were submitted, we have selected what we felt were the best 6 of those that arrived prior to this issue's press date (one for each brand of computer covered in our magazine, including a TI BASIC "10 liner"). If you have not yet submitted your masterpiece, it is not too late! As long as we keep getting great one-liners written in any computer language, we'll keep filling this page for you. Our prize winners this issue will each receive a check for $50 for sharing their ideas with our readers.

Pixel Palette
[TI Extended BASIC on the TI-99/4A]

Dear Sir:

My TI-99/4A one-liner lets you use a joystick to draw on the screen. You can design geometric configurations, letters, and numbers. To **RUN** this one-liner enter **CALL CLEAR** :: **RUN**. Make sure that **ALPHA LOCK** is up. The program displays your row and column position at the bottom of the screen. Type in the program until you hear the input beep, press **[ENTER]**, then **[FCTN 8]**, and finish typing the program in.

Robert Amenta
New City, NY

```
1 CALL CHAR(42,RPT$("F"
,16))::DISPLAY AT(24,9
)SIZE(7):R+1;C+1::CAL
L JOYST(1,X,Y)::R=R+Y/
-4::C=C+X/4::IF R>2
3 OR C>31 OR R<0 OR C<0
 THEN 1 ELSE CALL HCHAR
(R+1,C+1,42)::GOTO 1
```

Two By Two
[Commodore 64 BASIC on the C-64]

Dear Sir:

Here is a one-liner that allows limited word-processing. You can print two screen lines at a time, pressing **[RETURN]** after every two lines. You cannot use the comma, colon, or quotation marks. It works with most printers.

Chris Caldwell
Paris, TN

```
1 PRINTCHR$(14):OPEN3,4
:CMD3:INPUTA$:PRINT#3,"
■CRSRDOWN■";A$:CLOSE3:G
OTO1
```

Mandala
[Applesoft BASIC on the Apple II family computers]

Dear Sir:

This program creates oval patterns oriented around the center of the screen. With each plotted oval, the color alternates between black and white, forming an intricate design that covers the entire screen.

Andrea Sigurdson
Sidney, BC Canada

```
1 HGR2:FORD=1TO999:HCOL
OR=3:HPLOT0,0TO279,0TO2
79,190TO0,190TO0,0:FORG
=1TO2:S=(RND(1)*9+6)*.0
1:X=RND(1)*139:Y=RND(1)
*95:FORA=1.55TO3.15STEP
S:B=SIN(A)*X+139:C=COS(
A)*Y+95:HPLOTB,CTOB,190
-CTO279-B,190-CTO279-B,
CTOB,C:NEXTA:HCOLOR=0:N
EXTG:NEXT
```

Sum It UP
[TI BASIC on the TI-99/4A]

Dear Sir:

My great 10-liner is called the *Add/Subtract Calculator*. When you first **RUN** the program, the "Totals" value is 0. You can add to this value by selecting 1 then entering the number you wish to add, or subtract from it by selecting 2 then entering a number. Press **[ENTER]** and the screen clears, showing the new "Totals" value.

Richard Lin
Moberly, MO

```
1 CALL CLEAR
2 PRINT C
3 CALL KEY(0,S,D)
4 IF (S<49)+(S>50)THEN
3
5 INPUT B
6 ON S-48 GOTO 7,9
7 C=C+B
8 GOTO 1
9 C=C-B
10 GOTO 1
```

[NOTE: Because of built-in line length limitations in TI BASIC, we are now accepting "Ten-Liners" as entries for this column in a newly established TI BASIC category—Ed.]

Flying Fingers
[BASICA on the IBM PC, Cartridge BASIC on the IBM PCjr]

Dear Sir:

Here's a quick, little one-liner that will get your fingers flying in no time. *Qwiktype* lets you set the time for each practice session and keeps a running total of right and wrong answers as you type. It displays wrong answers as negative numbers. If you find that the program is displaying all negative numbers—no matter what you enter—hit **[Caps Lock]**. *Qwiktype* requires DOS.

Bob Langill
Hillsboro, OR

```
1 IF A THEN INPUT"TRY A
GAIN (Y/N)";B$:IF B$="Y
"THEN RUN ELSE END ELSE
 A=1:RANDOMIZE TIMER:CL
S:INPUT"MINUTES,SECONDS
";M,S:ON TIMER(M*60+S)
GOSUB 1:BEEP:TIMER ON:W
HILE 1:LOCATE 5,10:C$=C
HR$((RND*26)+64.5):PRIN
T C$,:C-D:A$=INPUT$(1)
:D=D+1:C=C-(A$=C$):WEND
```

Topsy Turvy
[Atari BASIC on the Atari 800,800 XL, and 130 XE]

Dear Sir:

This unusual one-liner turns the characters on your screen display upside-down. It takes a while to **RUN**, but it's worth the wait.

Thomas Zaninovich
Pittsburgh, PA

```
1 POKE106,155:GRAPHICS0
:FORI=0TO1023:D=D+2*(D<
18)-16*(D=16):POKE39944
+I-D,PEEK(57344+I):NEXT
I:POKE756,156
```

All One-Liner submissions are subject to the same publishing criteria as Letters to the Editor (explained in the magazine's masthead, page 6). If you have written a great One-Liner in any language on any computer covered by *HCM*, send it addressed to: Letters to the Editor, 1500 Valley River Drive, Suite. 250, Eugene, OR 97401. You too may win a cash prize and be immortalized in print!

HOME COMPUTER

VOL. V NO. 6 ★★★★ INTERNATIONAL EDITION

Apple Attack

Corporate Image Tarnished As Firm Sues Ex-Chairman

Will Sales Of New Add-Ons, Drop In Holiday Prices Help Give Stock More A-Peel?

A tarnished corporate image, an unsteady financial picture, cutbacks, and legal saber-rattling now characterize the new Apple Corp. New product campaigns and improved fourth-quarter financials fail to wash away the spectacle of Apple suing co-founder Steven Jobs. Apple filed suit against Jobs—who split from the company he co-founded, taking 5 key employees—for breach of fiduciary trust. Meanwhile, Apple itself may be facing some litigation: Three publishers of recently released books about MacBasic may file suit against Big Red for cancelling the new language—and licensing the code to Microsoft to incorporate in its own version of BASIC for the Mac—prior to MacBASIC's long-awaited release. But the beleagured Apple is not deserting its new corporate image. Dropping its "Macintosh Office" marketing approach, the firm says it will stick with its already developed office peripherals. The new 20 megabyte hard disk drive, a forthcoming file server, and various third-party items linking Mac to the IBM PC could help carve a large enough specialized office niche. But Apple will also push its graphics-intensive Mac into vertical markets—such as desktop publishing. A new Mac keyboard, color monitor, "open chassis" Mac, Apple II/Mac cross-compliler, and reduced-price Laser printer are expected for 1986 debut. Among recently released products for the Apple II are a 3 1/2" disk drive for $299, compact 1200-baud modem for $399, and a combination color/monochrome monitor for $399. A memory expansion card that will increase the Apple IIe all the way up to 1 megabyte is forthcoming.

Ironically, the scuffle with Jobs and others may be drawing attention away from these new toys—just when the holiday season is hitting. Yet, Apple has cut the Mac 512K price by $300, bundled Apple II family products to create below-$1000 systems at retail, and revived its holiday ad campaign in a bid likely to lift Christmas sales.

"QUOTABLES"

"We believe the leading edge is often the bleeding edge in this business."

—James J. Edgette, of Entre Computer Centers, Inc., a retail chain that is reportedly not stocking Commodore's Amiga.

What's News—

* * *

Tandy Corporation's success with its $1000 PC-compatible family is spurring it on to smaller and better things—a 16-bit laptop machine with 3 1/2" disk for $1595. Watch for holiday bundling of color monitor with model 1000 at same $1000 price.

* * *

"Shrink Wrap" laws, like the one recently passed by Illinois, may put copy-protection "under wraps." Under such legislation, now being considered by several states, the act of opening a software package implies agreement to copyright conditions. Detractors fear this could weaken consumer warranties.

* * *

Digital Research buckled under Apple legal pressure to make its GEM format more "distinguishable" from Mac programs. IBM also ruffled DR's feathers by announcing it wouldn't sell GEM as earlier planned. Apple may threaten suit against other Mac lookalikes for "visual copyright" violations.

* * *

A complete encyclopedia on one compact laser disk by Grolier, Inc. will soon sell at $199 (for the IBM PC and Atari ST). Requiring a CD-ROM drive and interface ($500-1000), the medium provides super fast data search and full cross-indexing of its 9 million words—a real home computer advantage.

* * *

Old soldiers never die . . . TexComp has premiered a 128K 99/4A-compatible, while Franklin has announced a new Apple II-compatible with an operating system that reportedly meets Big Red's legal scrutiny.

* * *

Big Blue strikes again. IBM has pulled off a major coup—recently signing a deal giving it access to several hundred high-tech Japanese patents for advanced optical and chip technology. The deal stems from patent-licensing access IBM gave Japan after World War II, and is predicted to give IBM an unprecedented technological edge.

IBM Hanging Tough

PC Sales Hold Constant PCjr Gets Push

Pact With Microsoft Promises Stability

Sales of both the IBM PC and the XT are holding steady during the market shakeout, while Big Blue has reaffirmed its commitment to the PC line's open architecture—recently agreeing to continue the reign of Microsoft's PC-DOS, rather than revert to a once-rumored proprietary operating system. This is good news for both third-party developers and those who use their products. Some skeptics think this is merely a ruse to buy time until IBM can issue a new closed-DOS 80386-based machine. The facts suggest, however, that IBM will probably have Microsoft revamp their TopView software, and fold MS-Windows into it.

Inclusion of a 30 Mbyte hard disk on the AT for a mere $200 increase in cost over the 20 Mbyte "garden variety" version may foreshadow yet another round of price cuts for all IBM personal computers. Meanwhile, expect another PCjr Christmas. With the street price of Junior with color monitor around $599 and special promotional offers to schools, the estimated 300,000-unit inventory should be cleaned out by early spring—just in time for the possible U.S. introduction of the Japanese-introduced (and Australian test-marketed) JX model with 3 1/2" disk drive.

Commodore Struggles

Firm Still Clinging To Amiga's Potential Success

But Will New Model Be A True Friend?

Commodore is suffering. Doomful company predictions foretell an $80 million loss in the firm's fourth quarter, a layoff of 700 employees, and a $50 million inventory write-down. Commodore executives are counting on Amiga to bring the company out of its slump. The firm hopes that the new computer, billed in a scheduled $40 million dollar ad campaign as the "Creative Edge," will return the company to solid financial ground.

But Amiga may not be enough. Analysts speculate that Commodore's home market image, its unreliable reputation, and lack of distribution will be impossible to overcome—even though a national ad has listed 662 dealers ready to handle the new computer. And Amiga has problems of its own: Its late arrival and lack of software development—in part due to an incompletely defined operating system—weaken its holiday appeal to all but a dwindling supply of "early innovators."

Commodore's unexpected success in dealer sign-up may be attributed to its single-level pricing structure and lack of purchasing volume requirements—a lure that allows stores of all sizes an equal chance to compete, while discouraging gray marketeering.

Soundbytes

Of Music and Microchips

by Andy Widders-Ellis
HCM Staff

Hark! Does the sound of electronically-produced music herald the next stage in the evolution of the personal computer?

It's a fact: Personal computers have permanently changed the way people produce music. The impact of this technology can be heard all around us. Curiously, the other side of the coin is often overlooked—how music has stirred up the world of home computers . . .

Heresy!

Music has been around a *long* time, and—let's face it—humans would have continued to express themselves through sound with or without the electronic revolution of the 80s (albeit in much more traditional ways). Music *benefits* tremendously from the home computer, but doesn't *require* it to stay vital.

On the other hand, what about the field of home computing—hasn't evolution been necessary for some time now? How many personal computers are gathering dust because their owners lack some deeper, satisfying long-term involvement with them?

Home computers have the reputation of drawing the user into isolated, solitary activities. We all know the typical stereotype: A computer user spends a lot of time alone—apart from family and friends—playing games, keeping books, studying, word processing, etc. Of course, most of us need some time alone, simply for peace of mind and mental health. An activity that allows us to work in reflective seclusion can be valuable.

But how much *more* valuable is a hobby that promotes both private time and *together* time? Computer-assisted synthesizer programming, composing, arranging, sequencing, and editing *does* necessitate working by yourself. Yet, once a creative project is complete, the musical fruits of this labor practically demand to be *shared*. Ultimately, music deserves an audience—the obvious thing to do once you've created your music is to let someone hear it.

Out Of The Closet

We are witnessing a renaissance: Once used as fancy calculators, game players, and effective but utilitarian business machines, personal computers are now becoming creative tools for personal self-expression. Thanks to this creative potential, *inspiration* for working with home computers is re-kindling. Home computers are emerging from the closet to appear in the living room again. Why? Because folks want hobbies and pastimes that they can develop over a lifetime, and that yield useful skills in exchange for the energy invested. To date, few computer activities have offered this much long-term, open-ended growth potential.

Also contributing to the rejuvenation of the home computer is the new owner who was never interested in mere "computing." These first-time buyers—at least initially—have one application in mind, namely music. It will be interesting to see whether, in time, these people find other uses for their computers.

What's It All About, PC?

If you subscribe to the theory that many of mankind's noblest ambitions and achievements are in the realm of *communication*, then something that supports this effort becomes very precious. Since the dawn of civilization, music has manifested this power to bring people together. We know man-made sounds can heal or wound, create or destroy, excite or soothe, elevate or depress. Cruise the radio dial to confirm this!

The point is this: Home computers place this powerful means of self-expression—once monopolized by a trained elite—into more hands than ever before. If you own a home computer, then *you* can wield previously unheard of control over this potent force. Doesn't this realization make the thought of playing "just another video game" seem a tad tame?

Knock On New Doors

The old notion that musical expression is for a chosen few no longer applies. With today's technology, you can compose and sequence wonderfully complex and detailed music with only *one finger*! If you have musical ideas, they can be realized using a home computer and computer-based electronic musical instruments to carry out the *physical* task of real-time sound production. Computers force us to redefine the term "musician."

Musical self-expression is within your grasp, even if you are a novice. Sure, you have to work at it—creative expression takes persistent effort. But the rewards justify this effort. As you gain knowledge and experience in music, you will come to see both yourself and your computer in a new light. Self-expression is satisfying and liberating, as therapists will attest. And the instrument that facilitates this creativity becomes especially important to the user.

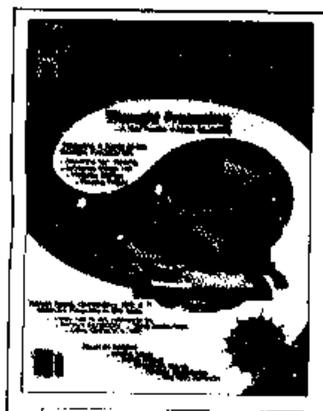
A new generation of computers is emerging, combining greater sophistication, super hi-res graphics, and more power—with lower cost. These features make them ideally suited for use as creative tools. The new Atari 520 ST even comes with a built-in MIDI (*Musical Instrument Digital Interface*) port! Obviously, Atari is counting on musical applications to enhance the value of its new product.

Turn On, Tune In, Take Off

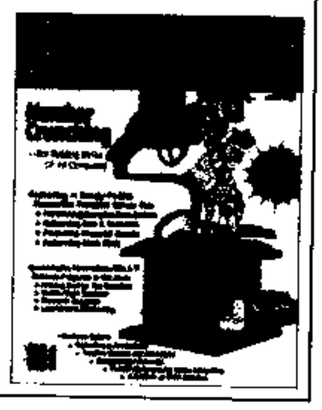
As our acknowledged universal language, music has accompanied (and often shaped) the unfolding drama of man. Why not come "on line" with your own contributions? A home computer can be your ticket to this creative journey. Who knows what you'll discover?

HCM Glossary terms: MIDI. **HCM**

HOME COMPUTER magazine SHOPPING LIST

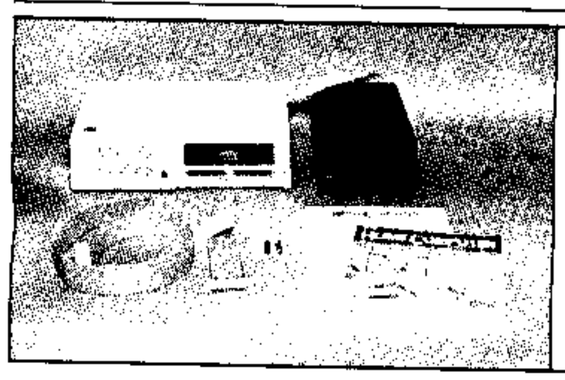


BACK ISSUES OF HOME COMPUTER MAGAZINE & MEDIA



	U.S. Surface	CANADA	FOREIGN Surface	FOREIGN Air
Magazine	3.95	5.95	5.50	7.50
Media	6.95	7.95	N/A	10.95
Magazine/Media Set	9.90	12.95	13.95	17.50

SEE PAGES 2 & 3 FOR ALL INDIVIDUAL CONTENTS



PCjr Disk Expansion Kit

Add A Second Disk Drive To Your PCjr for $49.95 PLUS 3.50 SHIP. & HANDL.

The Kit Includes:

- Special Cable To Connect the Second Disk Drive.
- 2 Integrated Circuits.
- A Copy of HCM Vol. 4 No. 4 Containing Instructions.
- Plus An Additional Update Sheet.

—PCjr, Disk Drive, and Power Supply NOT Included—

COMPUTER ACCESSORIES

ITEM NO.	ITEM DESCIPTION	PRICE	SHIPPING 1st one	SHIPPING each add'l
DSK20	20 Blank 5¼" Disks	22.00	3.50	3.50
DSK40	40 Blank 5¼" Disks	42.00	3.50	3.50
DSK60	60 Blank 5¼" Disks	59.40	3.50	3.50
PCJ101	PCjr Disk Expansion Kit	49.95	3.50	—
ODI	ON DISK Revue Vol. 1	14.95	2.00	2.00
	DUST COVERS			
DUS997	10" Monitor	10.95	2.00	.50
DUS72	13" Monior	12.95	2.00	.50
DUS74	MX80 Printer	9.95	2.00	.50
DUS75	Cassette Recorder	4.95	2.00	.50

TEXAS INSTRUMENTS 99/4A CLOSE–OUTS

ITEM NO.	ITEM DESCIPTION	PRICE	SHIPPING 1st one	SHIPPING each add'l
TBK	Best of 99'er	19.95	3.00	3.00
BTPOOL	Best of 99'er—Book & Tape Combo	35.00	ppd./U.S.	—
BPG999	Programmer's Guide	5.00	ppd./U.S.	—
TI0001	99'er Directory	5.00	ppd./U.S.	—
FSN-01	Simon's Saucer	10.95	2.00	2.00
	DUST COVERS			
DUS73	Expansion Box	12.95	2.00	.50
DUS76	TI Console	8.95	2.00	.50
DUS77	Speech Synthesizer	3.95	2.00	.50
DUS78	Disk Box Memory	5.95	2.00	.50
DUS79	32K Memory Exp.	5.95	2.00	.50
DUS80	RS232 Interface	5.95	2.00	.50
DUS81	Disk Controller	5.95	2.00	.50
DUS82	Thermal Printer	8.95	2.00	.50
CBL11	Single Cassette Cable	4.95	1.50	1.50
CBL12	Dual Cassette Cable	4.95	1.50	1.50

Complete the Subscription & Product Order Form and enclose payment or credit card information. Mail To:

Emerald Valley Publishing Co. ● P.O. Box 70288 ● Eugene, OR 97401

Or use our TOLL-FREE Order Line for VISA/MasterCard orders only:

1-800-828-2212 (Minimum $10 Order) In Oregon, Alaska, Hawaii Tel. (503) 485-8796

HOURS: MONDAY THROUGH FRIDAY 7:00AM TO 4:00PM; SATURDAY 9:00AM TO 1:00PM P.S.T.

Defective media gladly exchanged. NO REFUNDS on media. Offer & Prices Subject To Change Without Notice. Satisfaction Guaranteed—or the unfilled portion of your subscription will be refunded, less the cost of any premiums you have received.

BLANK DISK MEDIA OFFER

AS LOW AS 99¢ EACH!

As a service to our readers who prefer to type-in HCM programs, we are able to offer high-quality blank diskettes at very low prices.

	20 DISKS DSK20	40 DISKS DSK40	60 DISKS DSK60
5¼" certified single-sided, double density with reinforced hub ring. Bulk packed with white sleeves and indentification labels.	$22.00	$42.00	$59.40

PLUS $3.50 SHIPPING & HANDLING

BACK ISSUES OF 99'er HOME COMPUTER MAGAZINE & MEDIA

(The Forerunner of HCM)

QTY	MAGAZINE PRICES* U.S.	Canada	Foreign‡	MAG/MEDIA SET PRICES U.S.	Canada	Foreign‡	MEDIA PRICES U.S.	Canada	Foreign‡
3	$7.95	$10.95	$10.95	$15.95	$19.95	$21.95	$10.95	$12.95	$14.95
6	$12.95	$18.95	$18.95	$29.50	$37.95	$39.95	$20.95	$22.95	$25.95
12	$21.95	$29.95	$29.95	$49.95	$59.95	$63.95	$39.95	$44.95	$47.95

‡ Magazines Shipped to Foreign Countries via Surface (Minimum Order of 3)

— Partial Contents —

Issues No. 1-5 OUT OF PRINT

Contents available in book form as "Best of 99'er"—Vol. 1.

See Home Computer Digest or Inquire

THESE 2 MAGAZINES ARE OUT OF PRINT —TAPES AND DISKS ARE STILL AVAILABLE

ISSUE #5 (Partial Contents)
• How To Produce Sound Effects • Debugging a Game Program • How to Start a User's Group • Verbose A Speech Vocabulary Expansion Aid • Color Mapping • Dynamic Manipulation of Screen Character Graphics • The Beginner's Guide to Cassette Operation With the Home Computer • Pre-School Block Letters and Data Comparison • Picking the Ponies in TI BASIC • Battle Star Space Game • 3-D Animation on the Home Computer • Programming Tips • Who is LOGO for? • Tower of Hanoi in TI LOGO • A Review of the TI Lesson Development Software • An Interview with a Game Designer • Learning Assembly Language with a Magic Crayon • and much, much more

NOVEMBER 1982 (Partial Contents)
• Chatting with Your Micro: Languages for the Home Computer • A Review of the Smith Corona TP-1 Daisy Wheel Printer • The Micro Jaws Arcade Game • A Knight's Tour in TI BASIC • LOGO Has Style • ASPIC: A Language for Children • A p-System Beginners Tutorial • An interview with a p-System Pioneer • A Mini Memory Screen Dump for the Home Computer Printer • Up Scope! An exciting Undersea Combat Game • Strategy for Munch Man • A Brief Encounter with a TI Hand-Held Computer • 99'er Shopping Bus • A Pocket Battleship • Sub Programs in Extended BASIC • Arcade & Adventure Game Reviews • and much, much more

DECEMBER 1982 (Partial Contents)
• Tex Scribe: A Text Editor for the Home Computer • A Christmas Computer Carol • Managing a Mailing List the Futura Way • Parsec: The Arcade Game • Plotting With the Home Computer – Pixel by Pixel • Preventing the Situation—Oh No: Memory Full • A Colorful Tour of "I Feel" The Home Computer Show • Santa's Workshop: The Making of a Home Computer • The Turtle Arcade: Movies & Video Games in LOGO • Controlling a BASIC Termite • The 99'er Gold Rush: An Arcade/Adventure in the Home • 99'er Digest of News & Happenings in the TI World • Plus Games, Reviews and much, much more

JANUARY 1983 (Partial Contents)
• Computer Assisted Instruction for the Handicapped • p-System Basics • Debugging in LOGO • The Dow-a Gazelle Flight Simulator • Note Whiz and Pitch Master Musical Game Reviews • Learning With the PLATO Computer Library • Strategies for Adventure Gaming • Death Drones • Using the Line-By-Line Assembler • Class Encounters of the Simon Kind • Electrical Engineering Education Program • Interview With an Arcade Game Designer • TI Invaders • Programming With Pascal • Cyber Dice • News and Happenings in the Home Computer World • Arcade Game Reviews • The Thief Adventure Game • Programming Tips • and much, much more

FEBRUARY 1983 (Partial Contents)
• Texas Instruments at the Winter Consumer Electronics Show • Home Computer Printers on Review • How to Create Math Daisies in LOGO • Vectors in LOGO • ASPIC: A Language for Teachers • The Joys of Adventuring—Part 2 • Programming Printers with Chuck A Luck—Part 4 • Interview With the Voice of Parsec • Why You Need a Printer for Your Home Computer • Undine: In Titan Space Game • Night Brockade Battleship Game • Tower of Hanoi Pocket Program • Computer Gaming Software Reviews • News of Late Developments in the World of Home Computers • and much, much, more

MARCH 1983 (Partial Contents)
• An Introduction to the TI 99/2 Basic Computer • The Hex-bus and the 4/A Connection • Making Your Own Say and Spell Game • Disabled Children Learn and Grow • Super Cataloger—A Review of a Disk Library Utility Program • TI's New CC-40 Compact Computer • Robots and their Social Impact • Twenty Questions With Robot Redford • The Gravity of LOGO • Joystick Jockey: An Overview of Remote Controllers • Parsec Strategy • Converting Extended BASIC to Assembly Language • Matrix Muncher • Mini Memory Disassembler Utility • Pulling the Shade on Sprites • Letters on LOGO • Tiny Tutorials • Games programs reviews and much, much more

APRIL 1983 (Partial Contents)
• Computer Assisted Savings Planning to Build Your Nest Egg • Tex Cipher Writes and Decodes Secret Messages • Crossbytes—Computer Vocabulary Crossword Puzzle • Cutting Corners On Your Food Budget Using Coupons • Introducing Financial Planning with Multiplan • The Design Philosophy of the Compact Computer • LOGO Takes On the Popular Fifteen Puzzle • Super Language—Programming Sprites in Mini Memory • Colorful Word Wrap—Reading Readiness for Pre-schoolers • Gamewave Buffet's A Maze-ing Bna Alley Game • Giant and Dwarfs Entrapment Game • Game Reviews • Programming Tips • Money Saving Hints • and much, much more

MAY 1983 (Partial Contents)
• A Consumer's Guide to Word Processing • Word Processing Market Basket • A Generalized Filing Program for VIPs • The Multiplan Medium: Balances Your Checkbook and Budget • Activity Accountant Helps School Secretaries with Extracurricular Activities • Maximizing Your Mini Memory: 1K of RAM • Exploring Enhanced BASIC on the Compact Computer • The LOGO Tortoise Debates the BASIC Hare • A Pocket Program to Organize Data with Linked Lists • Mentally Handicapped Learners Team Up with the TI-99/4A • The Wonders of Diskette Storage • Beeline: a Multi-Screen Strategy Game • Lost Ruins—an Archeological Dig Game • 3-D Illusions with Sprites in Depth • Game reviews, Group Grapevine and much, much more

JUNE 1983 (Partial Contents)
• Children and Computers Make the 99/4A Connection • Tune Your Guitar with Our TI Tuning Fork • Talk to Your Computer—Voice Technology Is Here • Gamewave Buffet's Eat or Be Eaten Aardvark Game • Protect Your Station in the Space Zapper Game • What Multiplan Can and Can't Do • Understanding Inputs and Outputs in Drive For Diskettes Part 2 • Calculate Loan Schedules on the CC-40 • Go on a LOGO Vacation • Letters on LOGO • A Review of Upper Room Software's Programs for Special Learners • Construct an RS232/Joystick Interface • Group Grapevine • Shopping Bus • A Natural Language Interface for the Professional • Game Reviews • and much, much more

JULY 1983 (Partial Contents)
• The Evolution of Home Computer Graphics Comes Alive in Graphics Grows Up • Five Data Organizers in Never Out of Sorts • TI & 99'er at the Consumer Electronics Show • WarGames: The Movie and the Book • Editing with Multiplan • The LOGO Logician Presents To Model Is to Learn • LOGO Mosaic Designs Fill the Screen • Your Speech Synthesizer as a Spelling and Foreign Language Teacher • Software for Your Low-cost Printer Port • Gamewave Buffet's Treasure Island and the Colorful Switch-A-Row • A Book Review of Learn BASIC for CC-40 Users • 3-D Animation with the TMS 9918A Video Chip • Games Reviews • Group Grapevine • and much, much more

AUGUST, 1983 (Partial Contents)
• The Home Computer Goes To Work • Bit One: Part Two at the Fashion Factory • Better Business Bar Graphs in Graphic Persuasion • An Ensemble of Assemblers • Casilina Helps Money Management • Keystrokes for Thrifty Folks—a Review of Typewriter • Game Reviews of Cavern Quest and Starprobe 99 • Counting Fun for Preschoolers • Peripheral Vision 99 • Mean Machines and Smart Potatoes • Multiplan Medium Groups Cells into Rectangles • Turtle Text & LOGO Word Processor • Group Grapevine • Hello Little Brother—CC-40 Speaks to 99/4A • Pluto's Progress • Tunes at Student Assessment • Pocket Sunrise • Public Investigator • Gamewave Buffet's Jungle Jim Success Formula, and much, much more!

SEPTEMBER, 1983 (Partial Contents)
• Adding on to Your Home Computer • 99'er Directory of Commercially Available Software, Accessories and Peripherals • Peripheral Vision 99 Hardware Reviews • Pocket Sunrise: Part Two in Extended BASIC • Byte Lightning—Mini Memory Plays a Game • TI WRITER Tutorial • LOGO's Functions, Sets and Turtles • The CC-40 and 99/4A Take the Data and RUN • PLATO's Progress: Looks at Geometry Courseware and the Shape of Things to Come • Gamewave Buffet's Challenge of Camelot and The Fly • Game Reviews of No Wit and Crime and Punishment • Group Grapevine • 99'er Hall of Fame • 99'er Digest Update on New Products and much, much more!

OCTOBER, 1983 (Partial Contents)
• Adventures and Fantasy with Your Home Computer • Do-It-Yourself Adventure • Preprocessor Emulator: a 4-Bit Microprocessor • Have No Fear: Assembly Language Won't Byte: Part 1 • Make Your Mark • Letterbore and Debug Computing Cartoon • Once Upon a Tortoise Shell: A LOGO Adventure • Turtle Strut • Number Nibbler for Children • Lots of Plots on Your CC-40 and HX1000 Four-Color Printer/Plotter • Multiplan Medium: Bartender • A Grisly Adventure Bear Hunt • Escape From Wizard's Keep in Extended BASIC • Game Reviews • Shopping Bus • and much, much more

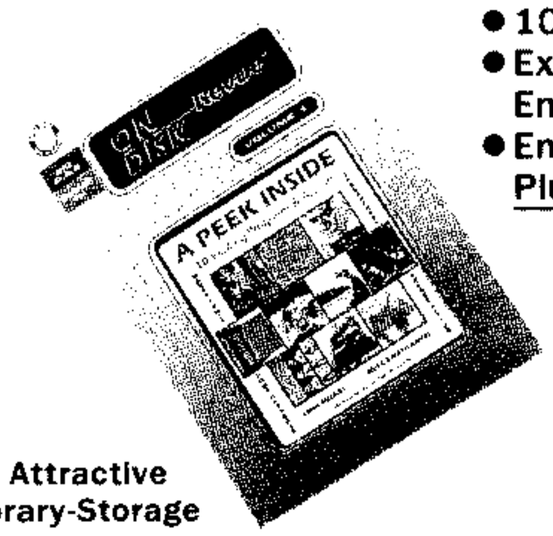
NOVEMBER, 1983 (Partial Contents)
• Education with your Home Computer • Five Creative Learning Activities for Children • Let's Build America • Have No Fear: Assembly Language Won't Byte, Part 2 • Squeezing the Most Out of T. BASIC • T. WRITER Tutorial • LOGO Exercises • Interview with Dale Osborn • TI WRITER: At Home in the Office • PLATO's Progress • Letterbore and Debug • The Multiplan Medium • Computer Arrested Instruction—99'er Interviews the Kids • Gamewave Buffet: Taco Man and Robo Chase • Game Reviews of Jet Break and Arithmetex • Hall of Fame • 99'er Digest • and much, much more

For Apple II Family, C-64, IBM PC & PCjr Users

ON DISK Revue™ Volume 1

- 10 Complete Programs On A Ready-To-Run Disk
- Exciting Activities For The Entire Family—At An Affordable Price
- Entertainment, Productivity & Education Plus Valuable Programming Secrets

Valuable Proof-of-Purchase Coupon For Gift Redemption



An Attractive Library-Storage Slipcase

CONTENTS

1. Boolean Brain
2. Electronic Home Secretary
3. Savings Planner
4. Snap-Calc
5. Tablut
6. Wild Kingdom
7. Musical Mystery Words
8. Market Madness
9. Tower of Hanoi
10. Missile Math

64-page Full-Color Book

Disk Stored in Re-usable Protective Pouch

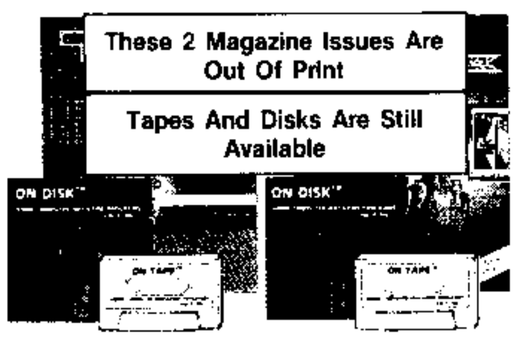
ONLY $14.95 plus $2.00 SHIPPING & HANDLING

A Back-Issue/Software Bonanza of 99'er HOME COMPUTER magazine™ At Unbeatable Prices

AS LOW AS $1.83 EACH MAGAZINE!
See Order Card on Other side of this page
ABOUT $4 PER SET!

The original 99'er Magazine and 99'er Home Computer Magazine were the forerunners of the present-day Home Computer Magazine. Each of these magazine back issues—exclusively covering the Texas Instruments TI-99/4A—is now available with your choice of either a floppy disk or a cassette tape that contains all the programs in that issue.

For Magazine Contents See "Shopping List" On Opposite Page



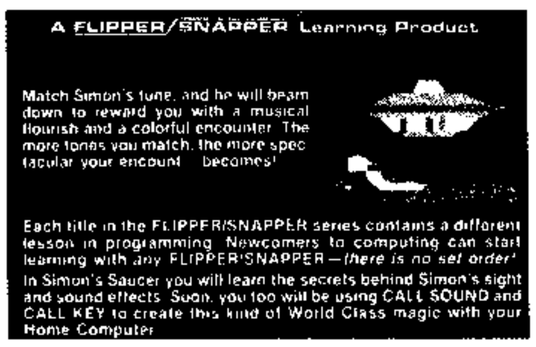
12 MAGAZINE & MEDIA SETS SAVE UP TO $44

✶ SPECIAL MIX OR MATCH BONUS ✶

—WHILE SUPPLIES LAST—

You will receive a FREE Simon's Saucer™ package —A $12.95 Premium Value— when your order includes at least 12 items from this Back Issue offer (either 12 media, 12 magazines, or 6 of each).

- A quality, ready-to-run game on cassette tape.
- A durable and attractive ring-binder collector's case for your software library.
- A complete, easy-to-use programming lesson on a deck of colorful flip cards.



MAGAZINES OR MEDIA MAY BE ORDERED SEPARATELY

To Order, Use Order Card On Other Side Of This Page.

Offer & Prices Subject To Change Without Notice.

Subscription & Product Order Form

Enclose payment or credit card information & mail with completed form to:
Emerald Valley Publishing Co. • P.O. Box 70288 • Eugene, OR 97401
Or use our TOLL-FREE Order Line for VISA/MasterCard orders only:
1-800-828-2212 (Minimum $10. Order) In Oregon, Alaska, Hawaii Tel. (503) 485-8796
HOURS: MONDAY THROUGH FRIDAY 7:00AM TO 4:00PM; SATURDAY 9:00AM TO 1:00PM P.S.T.

For more information see Shopping List on inside page.
Defective media gladly exchanged. NO REFUNDS on media.
Offer & Prices Subject To Change Without Notice.

☐ Check or Money Order Enclosed **(MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK)**

Charge my ☐ VISA ☐ MasterCard Date Expires _____ Tel. No. _____

Account No. _____ Signature _____

PURCHASER — SHIPPING ADDRESS [A] *Please Print*

Name _____
Address _____
TO INSURE FASTER DELIVERY, STREET ADDRESS IS ESSENTIAL
City _____ State _____ Zip _____

SHIPPING ADDRESS [B]

Name _____
Address _____
TO INSURE FASTER DELIVERY, STREET ADDRESS IS ESSENTIAL
City _____ State _____ Zip _____

MAGAZINE & PROGRAM SUBSCRIPTIONS: (FOR OFFICE USE ONLY)
New Mag.: _____ CODE Old Mag.: _____ CODE New Program: _____ CODE Old Program: _____ CODE

If any portion of your order is to be shipped to address "B", please indicate your preference by checking box A or B for each item.

ON DISK or ON TAPE
(CURRENT ISSUE)
$4.95 + $1.00 ship. & handl.
$6.95 Canada — $10.95 Foreign Airmail
(Non-purchaser, non-subscriber price: $12.95)

(Please indicate your type of computer media)

DISK VERSION						CASSETTE VERSION		
APPLE	ATARI	C-64	IBM PC	IBM PCjr	TI	ATARI	C-64	TI

Volume	Issue	
5	6	Price good Through Dec. 31, 1985

AMOUNT
$
SHIP TO: [A] ADDRESS [B]

HOME COMPUTER MAGAZINE SUBSCRIPTION: NEW ☐ RENEWAL ☐ GIFT ☐

LABEL NO. 7_____

	USA Surface	CANANDA Surface	FOREIGN Surface
1-YEAR (10 issue)	$25	$32	$46
2-YEAR (20 issue)	$45	$59	N/A
3-YEAR (30 Issue)	$63	$84	N/A

(Please indicate type of computer media for your 2 premium items)

DISK VERSION						CASSETTE VERSION		
APPLE	ATARI	C-64	IBM PC	IBM PCjr	TI	ATARI	C-64	TI

Your Subscription will begin with the up-coming new issue. ☐ *Please send a raincheck for the free software.*

Office Use Only
Sub.:
Media:
$
SHIP TO: [A] ADDRESS [B]

HCM'S ON DISK or ON TAPE PROGRAM SUBSCRIPTION NEW ☐ RENEWAL ☐ GIFT ☐

(Please mark your choice)

	USA	CANADA
8 ISSUES	$44.95	$49.95
10 ISSUES	$49.95	$54.95

(Please indicate your type of computer media) LABEL NO. 8_____

DISK VERSION						CASSETTE VERSION		
APPLE	ATARI	C-64	IBM PC	IBM PCjr	TI	ATARI	C-64	TI

$
SHIP TO: [A] ADDRESS [B]

ADDITIONAL SPACE FOR ORDERING MERCHANDISE (See Shopping List On Inside)

ITEM NO.	QTY	Description	UNIT PRICE (Each)	SHIP. & Handl. (Each)	SHIPPING INSTRUCTIONS	
ME010106	1	**Music & Electronics Magazine** (6-Issue Charter Subscription) Includes 6 Magazines and 6 M & E ON TAPE™ Instructional Cassette Tapes (U.S. Newsstand Value $56.70)	$32.97 (U.S.) $39.97 (CANADA)	INCLUDED	[A] SHIP TO: ADDRESS [B]	$
					[A] SHIP TO: ADDRESS [B]	$
					[A] SHIP TO: ADDRESS [B]	$
					[A] SHIP TO: ADDRESS [B]	$
					[A] SHIP TO: ADDRESS [B]	$
					[A] SHIP TO: ADDRESS [B]	$
					[A] SHIP TO: ADDRESS [B]	$

BACK ISSUES: HOME COMPUTER MAGAZINE and MEDIA*

(Please mark your choice)

ISSUE	4-1	4-2	AUG 4-3	4-4	4-5	5-1	5-2	5-3	5-4
Magazine									
Media									
Magazine/Media Set									

(Please indicate your type of computer media)

DISK VERSION						CASSETTE VERSION		
APPLE	ATARI	C-64	IBM PC	IBM PCjr	TI	ATARI	C-64	TI

U.S. Surface	CANADA	FOREIGN Surface	FOREIGN Air	QTY.	
3.95	5.95	5.50	7.50	×	=
6.95	7.95	N/A	10.95	×	=
9.90	12.95	13.95	17.50	×	=

$
SHIP TO: [A] ADDRESS [B]

BACK ISSUES: 99'er HOME COMPUTER MAGAZINE and MEDIA (COVERS ONLY TI COMPUTERS—THE FORERUNNER OF HCM)

(Please mark your choice) ☐ TAPE ☐ DISK **(Minimum Order of 3)**

ISSUE	1-6	NOV 2-1	DEC 2-2	JAN 2-3	FEB 2-4	MAR 2-5	APR 2-6	MAY 2-7	JUN 2-8	JUL 2-9	AUG 2-10	SEP 2-11	OCT 2-12	NOV 2-13
Magazine														
Media														
Magazine/Media Set														

QTY	MAGAZINE PRICES* U.S.	Canada	Foreign‡	MAG/MEDIA SET PRICES U.S.	Canada	Foreign‡	MEDIA PRICES U.S.	Canada	Foreign‡
3	$7.95	$10.95	$10.95	$15.95	$19.95	$21.95	$10.95	$12.95	$14.95
6	$12.95	$18.95	$18.95	$29.50	$37.95	$39.95	$20.95	$22.95	$25.95
12	$21.95	$29.95	$29.95	$49.95	$59.95	$63.95	$39.95	$44.95	$47.95

$
SHIP TO: [A] ADDRESS [B]

FOR OFFICE USE ONLY

A	B		A	B	
☐	☐	Air Mail	☐	☐	2nd Class
☐	☐	First Class	☐	☐	3rd Class
☐	☐	UPS (Standard)	☐	☐	Air Canada
☐	☐	UPS (Blue Label)	☐	☐	4th Class
☐	☐	Other _____	☐	☐	Foreign Surface

NOTES:
‡ Magazines Shipped to Foreign Countries via Surface
* Atari coverage commenced with Volume 5, No. 5.
- Merchandise shipped via UPS—requires street address.
- Satisfaction Guaranteed—or the unfilled portion of your subscription will be refunded, less the cost of any premiums you have received.
- Machines covered: Apple II family, Atari 800/800-XL/130-XE, Commodore 64/128, IBM PC/PCjr, TI-99/4A.
- For subscriptions, please allow 6-8 weeks for your first issue —magazines and media shipped separately.

APO/FPO Include $3.00 Extra Per Total Merchandise Order $

SUBTOTAL $

Priority Merchandise Handling ADD $2.00 (SUBSCRIPTIONS NOT APPLICABLE) $

TOTAL $

ORDER "A" DATE SHIPPED: _____ BY: _____
ORDER "B" DATE SHIPPED: _____ BY: _____
BACK ORDER DATE SHIPPED: _____ BY: _____

COMMENTS: _____
ACCT: _____
DATE: _____

SPO2561185

HISTORICAL NOTE
99'er Magazine (founded in December, 1980) was the forerunner of Home Computer Magazine.

THE BEST OF 99'er™

SUPER CLOSE-OUT SPECIAL FOR TI-99/4A USERS

A Giant Home Computer Compendium™ for the Texas Instruments 99/4A

The largest, most comprehensive collection of programs and articles ever assembled for the TI Home Computer

VOLUME 1

- Over 200 thoroughly tested key-in-and-RUN programs and sub-programs typeset in a grid format for maximum clarity.
- Programming instruction in 4 languages learn to use BASIC, Extended BASIC, LOGO and Assembly Language for everything from record keeping and money management to arcade-quality action games.
- A selection of sensational game software featuring full-color graphics, animation, and sound effects.
- Beyond the owner's manual—tips and techniques for getting the most out of your computer system.
- Computer-Assisted Instruction The home computer becomes your private tutor.
- Page after page of innovative applications transforming your computer into a home productivity center.

Regular (Pre-Close-out) Prices:

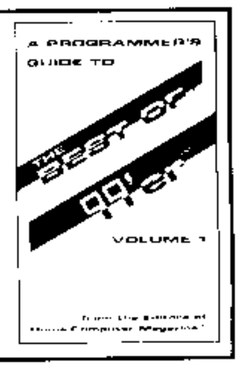
Best Of 99'er (Book alone)	$19.95	+ $3.00 shipping
Best Of 99'er (Tape Set alone)	$35.00	+ $2.50 shipping

SPECIAL OFFER ▶ Buy the Tape Set for ONLY $35. And Get the Book FREE + FREE SHIPPING (Canada add $3.00)

PLUS

FREE BONUS WHILE SUPPLIES LAST

ORDER THE BOOK & TAPE PACKAGE AND RECEIVE A Simon's Saucer AND A 99'er Programmer's Guide **ABSOLUTELY FREE!** This Additional $18 gift IS YOURS IF YOU ACT TODAY!



FREE BONUS WHILE SUPPLIES LAST

People Who Know the Home Computer Best™

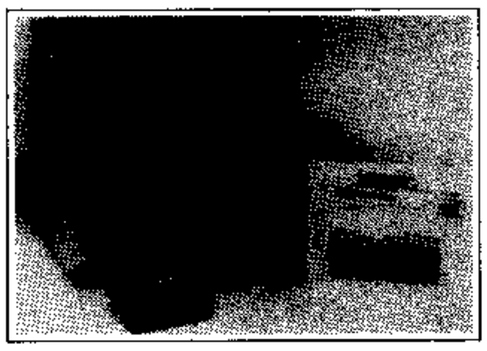
HOME COMPUTER magazine™

AN EMERALD VALLEY BOOK

Color Plotter "Solution"

Keychart Gets to Core of Apple Graphics and More

Softkey Software Products has rebuilt Keychart—a business graphics program—to operate on the Apple II, IIc, and IIe. The Apple-compatible program features on-screen layout preview capabilities and supports more than 20 different plotters and printers, including those manufactured by Hewlett Packard, Houston Instruments, and Apple. Keychart produces combination bar, line, symbol, pie, X-Y, horizontal bar, and scatter charts from data compiled on Lotus



1-2-3, Symphony, SuperCalc, Visicalc, MultiPlan, and Framework. Versions of Keychart are also available for the IBM PC, AT, XT, PCjr; IBM compatibles; and Texas Instruments Professional. It sells for $375.

Softkey Software Products, Inc.
18480 Decatur Dr.
Monte Sereno, CA 95030
(408) 395-1974

❖ ❖ ❖ ❖

Four Years in the Wings

Super Pascal Developed for Commodore

Abacus Software has released a complete Pascal Development System—Super Pascal—for the Commodore 64 and 128. Super Pascal utilizes a file editor, an assembler for coding-in machine language routines, a utility package for file and disk management, high-precision 11-digit arithmetic, and high-speed DOS. The package, which includes a 200-page handbook, sells for $59.95.

Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

❖ ❖ ❖ ❖

Carrying a Heavy Load

Big-Byte Disk Drive for Atari

Astra System's newest disk drive, the Big D, is a double-sided, single- or double-density dual disk drive for Atari computers. Big D is supplied with the latest version of TOPDOS and can store 720K. It's compatible with Atari's new XE Series, as well as Atari's intermediate computers—the 400, 800, 600XL, 800XL, and 1200XL. Astra also makes two other Atari-compatible disk drives—the 2001 and the 1620. Both are supplied with SMARTDOS or MYDOS and can store 360K. The Big D has a suggested retail price of $595. The 2001 is $495, and the 1620 will be closed out at $300.

Astra Systems, Inc.
2500 Fairview, Unit L
Santa Ana, CA 92704
(714) 549-2141

❖ ❖ ❖ ❖

Hang On To Your TI

Software Combines PC and 99/4A

Intelpro has introduced Upwards!, a new product designed to quickly transfer data between the TI 99/4A and the IBM PC. The package includes two programs that run concurrently on both computers, so that the TI serves as a "remote file server," transferring data at a rate of 8,000 characters per minute. The product requires a fully configured TI 99/4A, an Extended BASIC cartridge, an IBM PC with at least 128K of memory (256K is recommended), DOS 2.0 or 2.1 for the IBM, RS-232 capabilities for both computers, and a cable to connect the two computers. Upwards! has a suggested retail price of $79.95 for U.S. buyers and $99.95 for Canadian buyers.

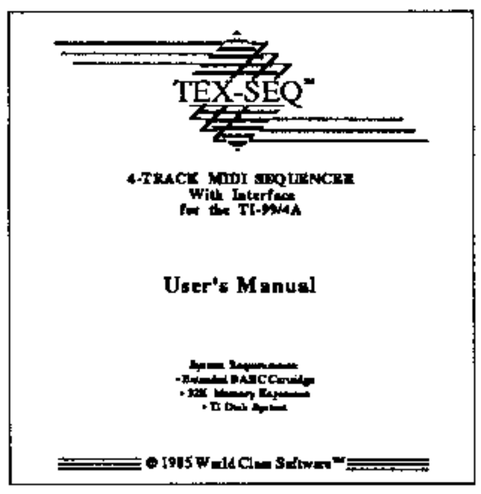
Intelpro
5825 Baillargeon St.
Brossard, Quebec, Canada J4Z 1T1
(514) 656-8798

❖ ❖ ❖ ❖

MIDI Meets TI

New World Class MIDI Software For 99/4A

World Class Software unveils TEX-SEQ, a MIDI (Musical Instrument Digital Interface) sequencer package for the TI-99/4A computer. TEX-SEQ comes complete with MIDI cable interface and software. The package allows note-by-note entry of musical compositions, which will play on any MIDI synthesizer. Software provides 4 play tracks and supports several different time signatures. TEX-SEQ requires 32K memory expansion, Extended BASIC, and a disk drive. Retail price is $49.95.



World Class Software
1500 Valley River Dr., Suite 250
Eugene, OR 97401
(503) 485-8796

❖ ❖ ❖ ❖

Homework Helpers

Bring the Teacher Home

Spinnaker Software unveiled Homework Helper Math World and Homework Helper Writing to teach students in grades 7-12 step-by-step methods for math and writing homework. Math World contains tutorial and problem-solving sections, and Writing focuses on book reports, essays, and research papers. Both programs are available for the Commodore 64 and 128 at $32.95 and for the IBM PC and Apple family computers at $39.95.

Spinnaker Software
One Kendell Square
Cambridge, MA 02139
(617) 494-1200

❖ ❖ ❖ ❖

Commodore Cost Calculator

Crunch Bills With Energy Manager

A new program for the Commodore 64, designed to track energy use in homes and buildings, is now available from Powerline Software. Energy Manager uses utility bills as a data base to analyze heating and cooling energy usage. It enables month-by-month comparison of energy expenditures for the current year with those of previous years and measures the effect of energy conservation tactics. Energy Manager comes on diskette with interactive data-entry and editing programs, analysis programs, sample data, and user's manual. It's available for $59.95.

Powerline Software
P.O. Box 635
New Hartford, NY 13413
(315) 735-0836

❖ ❖ ❖ ❖

Hands-On Screen System

Apples Get the Personal Touch

Personal Touch is now shipping Touch Window, a four-in-one touch screen input device for the Apple II family of computers. Mounted directly on a monitor, Touch Window provides any Apple II family computer with a see-through touch screen system. The package comprises a word processor, a picture/graphic creator, a spreadsheet application, a puzzle, and a program to check and recalibrate the touch screen. The product can be removed from the monitor for use as a graphics tablet, input pad, or interactive book pad. The system sells for $199.95 and comes with the Interactive Book I, a book containing touch screen applications and activities for ages three to adult.

Personal Touch Corp.
4320 Stevens Creek Blvd. Suite 290
San Jose, CA 95129
(408) 246-8822

❖ ❖ ❖ ❖

A First for Forth

Apples Speak Same Language

MicroMotion recently released Masterforth for the Apple family of computers. This version of the Forth programming language provides a complete programming environment for the Apple family, comprising a 65C02 macro-assembler and full interface to Apple DOS 3.3. As standard features, MasterForth employs a string package, screen editor, and resident debugger. Version 1.0 retails for $125 and is also available for the Macintosh, the IBM PC and CP/M-operating system machines. Users can write software on one system and run it on all the others.

MicroMotion
12077 Wilshire Blvd. #506
Los Angeles, CA 90025
(213) 821-4340

❖ ❖ ❖ ❖

Energized Word Processing

PaperClip is Revitalized for Atari, Commodore Computers

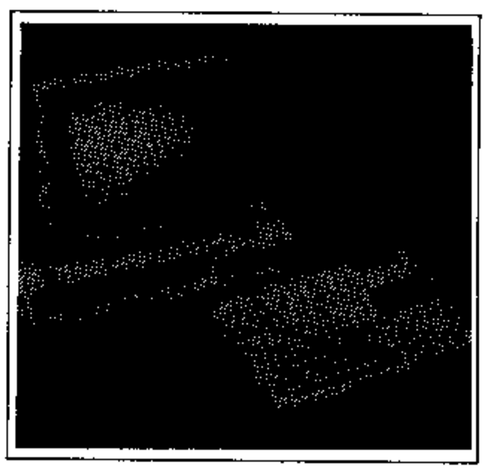
Originally issued for the C-64, the PaperClip Professional Word Processor by Batteries Included is now available in an updated version for Atari 800 and compatible computers. The program offers Mail-Merge, dual-editing windows, graphic capability, and many printing functions. It sells for $80. PaperClip has also upgraded the Commodore version with a SpellPack. Its dictionary includes over 20,000 entries and can be *user-expanded by 5,000* words. PaperClip with SpellPack for the Commodore retails for $149.

Batteries Included
30 Mural St.
Richmond Hill, Ontario
L4B 1B5 Canada
(416) 881-9816

❖ ❖ ❖ ❖

TI Clone With A Plus

Tex-Comp Unveils 128K 99/4A-Compatible, Peripherals



Tex-Comp has announced its latest development, the TC-99/9. It's a 128K TI-compatible computer with a TI processor board modified for a 40/80 column display. Tex-Comp does not plan to produce the TC-99/9 except on a special order basis. The firm expects to sell the computer for under $1500. Tex-Comp also introduced its latest in enhanced peripherals for the TI-99/4A: the TC-1, TC-2, and TC-3. The TC-1 is a new expansion enclosure equipped with a fan-cooled regulated power supply, two double-sided, double-density drives, and a rear panel of 6 spike and surge protected accessory outlets. It sells individually for $450 and, packaged with the Corcomp 9900 Micro Expansion, for $700. The TC-2 includes the TC-1, the TI speech synthesizer, the Corcomp 9900, and the the TI-99/4A console. Available by special order only, it sells in the $800 range. Also supplied by special order is the TC-3—containing the same electronics as the TC-2, but with an added 128K of memory. It sells for under $1000.

Tex-Comp
P.O. Box 33084
Granada Hills, CA 91344
(818) 366-6631

❖ ❖ ❖ ❖

Black Belts Beware

Games Converted for Atari

Broderbund's Championship Lode Runner and Karateka are now available for the Atari 400, 800, and XL/XE computers. In Lode Runner, players attempt to elude guards and move a galactic commando across floors to retrieve gold. Karateka is a karate simulation, wherein users play the role of a young karate master who must battle with an evil warlord and his warriors to rescue a princess. Karateka for Atari computers requires a joystick and 48K. Lode Runner may be played with a joystick or keyboard and also requires 48K. Both games are $34.95.

Broderbund Software
17 Paul Dr.
San Rafael, CA 94903
(415) 479-1170

❖ ❖ ❖ ❖

Software Cloning

MIDI Music Shop Developed

Passport Designs has entered into a licensing agreement with Broderbund Software and developed a MIDI version of Broderbund's The Music Shop. Passport reviewed many music-composing programs and chose The Music Shop for its editing features. The software allows users to create, store, and edit compositions and print sheet music in piano or single-staff (treble, bass, or both) formats. It requires a one-drive Commodore 64, a MIDI keyboard, a joystick, the Passport MIDI Interface, and a dot-matrix printer. The MIDI version of The Music Shop retails for $99.95 and is available at music stores.

Passport Designs, Inc.
625 Miramontes St.
Half Moon Bay, CA 94109
(415) 726-0280

❖ ❖ ❖ ❖

Quantum Leap for Commodore

Telecommunications Network Introduced

Quantum Computer Services has announced the start up of QuantumLink, a nationwide interactive telecommunications service for Commodore computers. The network offers software previews, electronic messaging and bulletin boards, teleshopping, news, a computer information center, interactive telegaming with full-color graphics, on-line chat, and an electronic encyclopedia. QuantumLink's monthly subscription fee of $9.95 includes connection fees and communications charges for most of the offered services. In early 1986, QuantumLink will be available to other computers.

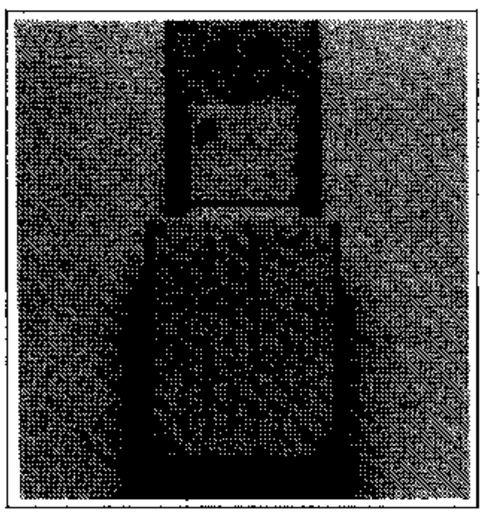
Quantum Computer Services, Inc.
8620 Westwood Center Dr.
Vienna, VA 22180
(703) 448-8700

❖ ❖ ❖ ❖

Modem, Modem On The Wall . . .

Telecommunications With The Apple II

The Apple Personal Modem, now available from Apple, is a compact 1200/300-baud modem that plugs into a wall socket or power strip and enables any Apple personal computer system to communicate with mainframes, minicomputers, and other microcomputers. The Apple Personal Modem works with either tone-generating or pulsed telephone systems. It has dial, answer, and status-display capabilities. The Apple Personal Modem is available for $399.



Apple Computer, Inc.
20525 Mariani Ave.
Cupertino, CA 95014
(408) 996-1010

❖ ❖ ❖ ❖

The Next Generation of Printers

Epson Introduces FX New-Borns

Epson America recently rolled out two new printers designed for business applications—the FX-85 and FX-185. Both models produce draft copy at 169 cps and near-letter-quality copy at 32 cps. They also offer single-stroke-selectable printing modes from built-in IBM character sets and an 8K print buffer that frees computer time while printing. The FX-85 is listed at $499, the FX-185 at $699. Optional cut-sheet feeders for automatic single-sheet use are available for both models ($269 for the FX-85 feeder and $319 for the FX-185 feeder). Epson also offers $85 upgrade kits for older FX-series printers.

Epson America
2780 Lomita Blvd.
Torrance, CA. 90505
(213) 421-5426

❖ ❖ ❖ ❖

It's Only Logical

Software Designed to 'Booleanize' Students

Sunburst Communications recently made High Wire Logic available for the Apple II family of computers. Originally part of Sunburst's Discrimination, Attributes and Rules package, the software introduces students to Boolean logic and strengthens and extends students' capabilities in logical thought. The program is designed for students in grades 5-12 and retails for $59. It requires a 48K Apple II-series computer.

Sunburst Communications, Inc.
Pleasantville, NY 10570
(914) 769-5030

❖ ❖ ❖ ❖

A Special Note on C-64 Listings

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.

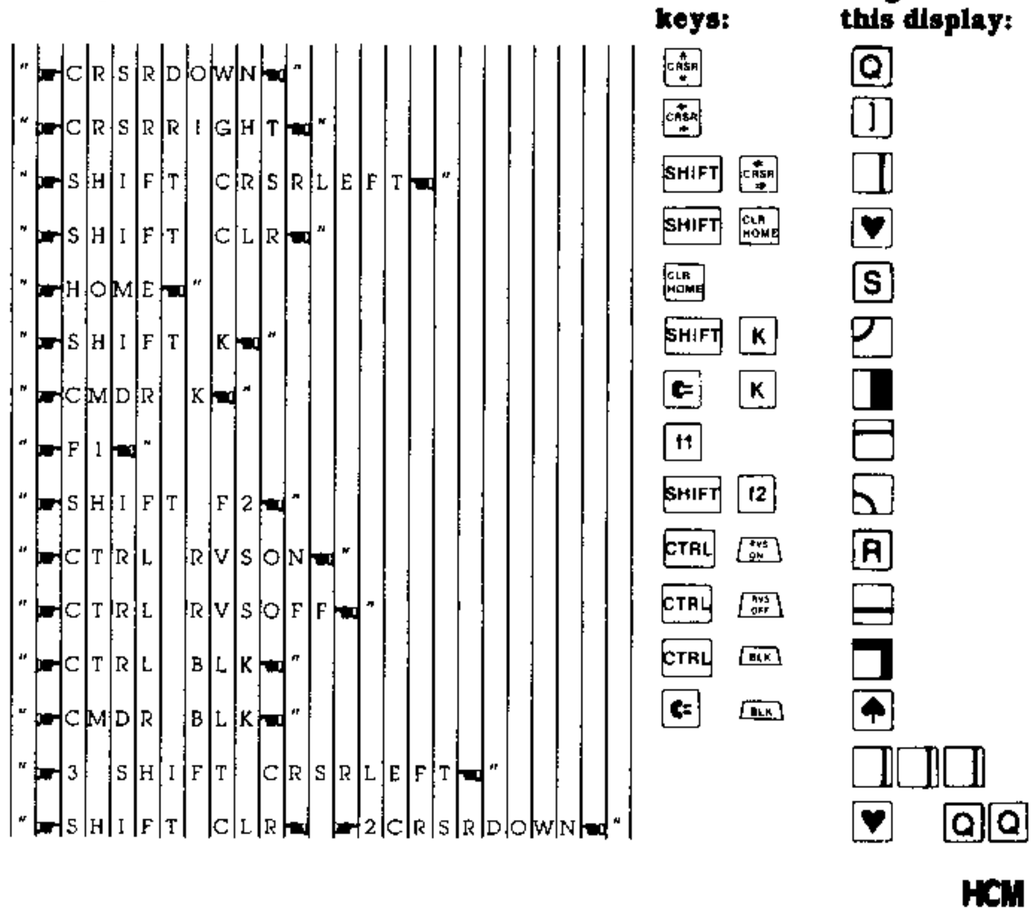
A number is included if you need to repeat the operation: ☞8 SHIFT CRSRLEFT☜ tells you to hold the [SHIFT] key down and press the cursor left key (on the bottom right of the keyboard) 8 times.

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, the Commodore key, or the shift key *first* and hold it down before pressing the second key.**
- **Everything between a pair of hand symbols is set in a different typeface.**

In Figure 3, we have included a chart showing you a representative sample of the symbols that appear when you use keystrokes enclosed by the hand symbols. (Notice that the hand symbols always appear within quotation marks—as in a print statement.)

Figure 3: C-64 Special Symbols



HCM BUG-OUT

Bug-Out is an error detection program for catching type-in mistakes. It is available for every computer brand *HCM* covers—Apple, Atari, Commodore, IBM, and Texas Instruments. When you use this utility, typos are easily found and corrected.

Before you type-in another *HCM* program, type-in the *Bug-Out* program specified for your computer. Because a properly typed-in *Bug-Out* routine is essential for it to accurately detect typing errors in other programs, be extra careful to ensure accuracy. Once you have it entered, save *Bug-Out* to tape (an option for Atari and Commodore only) or disk.

Comparing Two Sets of BOC's

When you look at our listings, you will notice an upper-case letter placed in the left-most column at the beginning of each program line. Separated from the line numbers by a bold vertical bar, this letter is the correct BOC. Do **not** type these letters in. The BOC is a quality-control character. Each program line is carefully dissected, and mathematically compacted into a single-character representation.

These letters will help you detect key-in errors *after* you have a listing fully entered. When you **RUN** the *Bug-Out* program as directed below, it will generate another series of BOCs either on the screen, or to a printer. Compare the codes generated by the *Bug-Out* program with the codes published in the left-most column of our listings. If a published BOC for a line is different from a BOC for your typed version of the same line, you will know that line contains a typing error.

How To Do It

1. The first step is to type-in the desired program.
2. After typing the program in, **SAVE** it as usual.
3. Then also **SAVE** the program as an ASCII text file—this is the format needed for *Bug-Out* to do its job. Always use a different file name to distinguish between the program file **SAVE**d in step 2 and this text file. We suggest you add a suffix like .T (or _T on the 99/4A) to the end of the text file name for added clarity. The process of saving programs as ASCII text files on each machine is detailed on page 78 (see "Turning Programs Into Text Files").
4. After you've **SAVE**d your program as an ASCII text file, make sure that the disk or tape containing the text file is inserted, then **RUN** the *Bug-Out* program. Once **RUN**, *Bug-Out* will ask for the name of the ASCII file and whether you want the program's output to go to the screen or the printer. After all this has been entered, the computer will print out its list of BOCs and the corresponding line number. For example:

```
N 100
S 110
Q 120 . . .
```

5. Carefully go through the program listing in the magazine to find, and take note of, all the published BOCs that are different from the BOCs generated by the *Bug-Out* program. Every line that you find with a different BOC code has been typed incorrectly, and should be carefully examined and corrected.

To correct your mistakes (if any), **LOAD** (**OLD** on the 99/4A) the program version (not the text file) that you keyed in previously. Now, make the necessary changes to the incorrect program lines and repeat the previous 5 steps until all the BOC codes match. Once they all match, your program should be error free.

REM statements that are not typed correctly will result in erroneous BOCs. If the only differences between a typed-in program and the magazine listing are in **REM** statements, the program will still **RUN** as intended. So you needn't waste time concerning every REM statement before running your *HCM* programs. ***Continued***

TURNING PROGRAMS INTO TEXT FILES

Apple II Family: The Apple method of making a program into a text file requires merging your typed-in program with a short *Capture* program (see below). [Note: in *HCM* Vol. 5, No. 5, the *Capture* program was regretably omitted.—Ed.] Our version is based on the *Capture* program found on page 140 of the *BASIC Programming with ProDOS* manual published by Apple. If you do not have either *Apple Programmer's Assistant (APA)* or *Renumber* program in DOS 3.3, you must type-in *Capture* at the same time as the program you are **SAVE**ing as a text file.

In either case, with *Capture* (at lines 1 through 10), and the program you wish to capture as a text file in memory, just type **RUN**. The *Capture* program then **LIST**s the program in memory to disk as a text file under whatever name is **INPUT** when *Capture* first starts running. Because *HCM* programs always begin with line 100, the *Capture* program always **LIST**s starting at line 100, and does ***not*** **LIST** lines 1-10, which do the actual capturing.

Atari disk users enter: LIST "D:PRGNAME**.T"**
Atari tape users enter: LIST "C:PRGNAME**.T"**

C-64 disk users enter: OPEN 8,8,8,"PRGNAME**.T,S,W" : CMD 8 : LIST**
When the cursor returns, enter: **PRINT#8 : CLOSE 8**
C-64 tape users enter: OPEN 1,1,1,"PRGNAME**.T" : CMD 1 : LIST**
When the cursor returns, enter: **PRINT#1 : CLOSE 1**

IBM PC and PCjr users: SAVE the program with the ASCII option like this: **SAVE"**PRGNAME**.T",A**

TI-99/4A users: You can **LIST** your program to disk from Extended BASIC by typing the following, with your newly typed-in program residing in memory: **LIST "DSK1.**PRGNAME_**T"**

Special Note to Apple Users:

Applesoft BASIC has an idiosyncracy concerning REM and DATA statements that our Bug-Out program takes into consideration. To ensure identical BOCs, always make sure your REM and DATA statements are typed *exactly* as they appear in the listing. Although REM statements do not affect a program's performance, DATA statements are often a source of typing bugs. Therefore, after either typing in or editing a DATA statement, be sure that it has the same number of spaces as in the magazine.

CAPTURE

APPLE II Family

```
Q    1 D$ = CHR$ (4)
O    2 HOME : PRINT "PLACE DISK IN DRIVE 1": PRINT "LIST TO WHAT FILE NAME (NOT SAME NAME AS PROGRAM FILE NAME!)": INPUT FL$
Y    3 PRINT D$;"OPEN ";FL$;",D1"
T    4 PRINT D$;"CLOSE ";FL$
T    5 PRINT D$;"DELETE ";FL$;",D1"
V    6 PRINT D$;"OPEN ";FL$;",D1"
H    7 PRINT D$;"WRITE ";FL$
A    8 LIST 100 -
1    9 PRINT D$;"CLOSE ";FL$
V   10 PRINT "*** DONE ***": END
```

BUG-OUT

APPLE II Family

```
B  100 REM ***********
X  110 REM * BUG-OUT *
D  120 REM ***********
Y  130 REM COPYRIGHT 1985
N  140 REM EMERALD VALLEY PUBLISHING CO.
Y  150 REM BY THE HCM STAFF
B  160 REM HOME COMPUTER MAGAZINE
B  170 REM VERSION 5.5.1
P  180 REM APPLE II FAMILY APPLESOFT
Q  190 D$ = CHR$ (4):CR$ = CHR$ (13):SP$ = CHR$ (32)
Q  200 TEXT : HOME : PRINT TAB( 10);"HCM BUG-OUT PROGRAM": PRINT : PRINT "BEFORE USING, LIST PROGRAM TO DISK WITH": INVERSE : PRINT "CAPTURE";: NORMAL : PRINT " PROGRAM -- ALSO INCLUDED IN THIS ISSUE"
K  210 VTAB 8: HTAB 1: PRINT "PLACE DISK IN DRIVE 1, THEN": PRINT : PRINT "INPUT FILENAME: ";: INPUT "";FL$: IF FL$ = "" THEN 210
M  220 ONERR GOTO 380
F  230 PRINT D$;"VERIFY ";FL$;",D1"
P  240 PRINT D$;"OPEN ";FL$;",D1"
T  250 VTAB 11: HTAB 1: PRINT "OUTPUT TO:": PRINT " 1. SCREEN": PRINT " 2. PRINTER IN SLOT 1"
V  260 GET K$: IF K$ < "1" AND K$ > "2" THEN 260
L  270 PRINT K$: IF K$ = "1" THEN 290
A  280 PRINT D$;"PR#1"
Q  290 PRINT D$;"READ ";FL$
K  300 GET C$
B  310 IF C$ = "" THEN 300
X  320 IF C$ = CR$ GOTO 340
H  330 A$ = A$ + C$: GOTO 300
I  340 IF A$ = "" THEN 300
N  350 IF LEFT$ (A$,1) = SP$ THEN A$ = RIGHT$ (A$, LEN (A$) - 1)
B  360 LN = VAL (A$):B$ = STR$ (LN): GOSUB 500
D  370 PRINT SP$ + B$ + SP$ + OT$:A$ = "": GOTO 290
C  380 POKE 216,0:EN = PEEK (222):EL = PEEK (218) + PEEK (219) * 256
Z  390 IF EN < > 5 THEN 430
U  400 PRINT "*** DONE *** WITH ";FL$
Y  410 PRINT D$;"CLOSE ";FL$
A  420 PRINT D$;"PR#0": END
S  430 ON ((EN = 5 OR EN = 6) + 2 * (EN = 8) + 3 * (EN = 16)) GOTO 440,450,460
Z  440 PRINT FL$ + " IS NOT ON DISK": GOTO 470
K  450 PRINT "I/O ERROR-CHECK DRIVE": GOTO 470
W  460 PRINT "ILLEGAL FILE NAME-TRY AGAIN"
Q  470 CALL - 3288: PRINT "PRESS ANY KEY TO CONTINUE": GET K$: RUN
E  480 PRINT "ERROR #";EN;" DETECTED AT LINE ";EL: PRINT "TRY STARTING AGAIN."
A  490 END
G  500 CK = 0:C1 = 0: FOR I = 1 TO LEN (A$):CH = ASC ( MID$ (A$,I,1)):C1 = C1 + CH
Q  510 IF I / 2 = INT (I / 2) THEN CK = CK - CH: GOTO 530
Y  520 CK = CK + CH
A  530 NEXT I:CK = ABS (CK) * C1:I = INT (CK / 26):CK = CK - I * 26
V  540 OT$ = CHR$ (65 + CK): RETURN
```

HCM

BUG-OUT

ATARI 800/800XL/130XE

```
N  100 REM ***********
Z  110 REM * BUG-OUT *
F  120 REM ***********
M  130 REM COPYRIGHT 1985
R  140 REM EMERALD VALLEY PUBLISHING CO.
O  150 REM BY THE HCM STAFF
N  160 REM HOME COMPUTER MAGAZINE
V  170 REM VERSION 5.5.1
T  180 REM ATARI BASIC FOR THE 800, 800XL, 130 XE
U  190 DIM FILE$(16),IN$(255),BL$(7):BL$="       ":POKE 82,0
B  200 ? "ESC CTRL <   HCM BUG-OUT":?"  **Program must be LISTed file**":?
N  210 ? "ENTER DEVICE & FILENAME";:INPUT FILE$:IF FILE$="" THEN 330
C  220 TRAP 340
A  230 OPEN #1,4,0,FILE$:? "LIST TO SCREEN OR PRINTER (S/P)?":POKE 764,255
V  240 K=PEEK(764):IF K<>62 AND K<>10 THEN 240
E  250 ? :? "READING...":TRAP 330
O  260 INPUT #1;IN$:P=1
W  270 IF IN$(P,P)<>" " THEN P=P+1:GOTO 270
A  280 CK=0:CK1=0:FOR I=1 TO LEN(IN$):CK1=CK1+ASC(IN$(I,I)):IF I/2=INT(I/2) THEN CK=CK-ASC(IN$(I,I)):GOTO 300
X  290 CK=CK+ASC(IN$(I,I))
Y  300 NEXT I:CK=ABS(CK)*CK1:I=INT(CK/26):CK=CK-I*26
Z  310 IF K=10 THEN LPRINT CHR$(CK+65);BL$(1,6-P);IN$(1,P):GOTO 260
D  320 PRINT CHR$(CK+65);BL$(1,6-P);IN$(1,P);"  ";:GOTO 260
F  330 CLOSE #1:?:? "DONE":POKE 764,255:END
G  340 ? :STATUS #1,ST:? "** I/O ERROR ";ST;" **":SOUND 1,50,10,8:FOR I=1 TO 1000:NEXT I:SOUND 1,0,0,0:CLOSE #1
T  350 GOTO 200
```

HCM

PROGRAMMER'S WINDOW

Card-Shuffler

REMARKS

The most important routine in the TI version of *Card-Shuffler* is the Get-A-Folder routine, located in lines 340-480. This routine searches various folders and creates a record of each *Card-Trix* card that meets the search parameters' specifications. Without this routine, the program is useless.

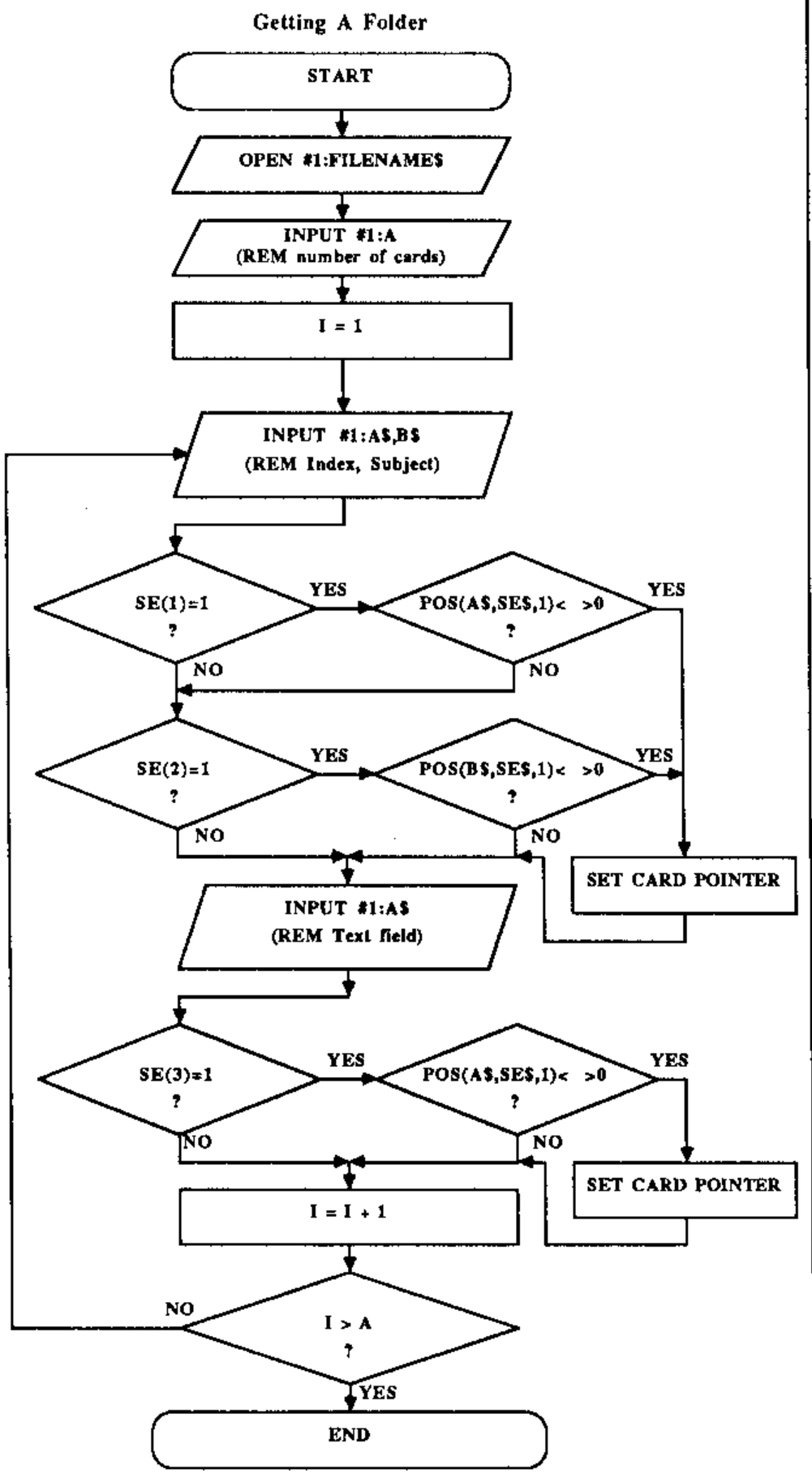
When the Get-A-Folder routine finds a card, it does not store the actual card in memory—it stores a pointer to the card. As a result, memory can hold as many as 4 card folders at one time. The C(,) array contains the card pointers.

We use search parameters to specify what the routine is to search for. The search parameters consist of the item being searched for and the fields to be searched through. The SE$ variable contains the item being searched for. The three elements of the SE( ) array act as flags indicating which, if any, of the three fields should be searched. If SE(1) equals one, the routine searches through the first field: the Index field. If SE(2) equals one, the routine searches through the second field, the Subject field. Similarly, SE(3) controls the third field, the Text field.

The Design Focus on this page presents a flow chart of this routine. First, the routine OPENs the desired file and INPUTs the first item. The first item of a *Card-Trix* file informs us of the number of cards that are in the folder. The variable A stores this number. Next, we INPUT the first two fields of a card—the Index and Subject fields. Using the SE( ) array, the program checks whether it should search through these first two fields. If it should, the program searches through each field using the POS( ) function. When the POS( ) function returns a non-zero value, the routine sets a pointer to the current card. Next, the routine INPUTs the Text field. The program checks SE(3) to see if this field should be searched. The program uses the POS( ) function to search the Text field, as well. Once again, if the POS( ) function returns a non-zero value, the routine sets a pointer to the current card. Using a FOR-NEXT loop, this process repeats the number of times specified by the variable A.

HCM Glossary terms: array, element, field, function, parameter.

DESIGN FOCUS



LISTING ANNOTATIONS

Line Nos.	
100-180	Program header
190-200	Initialize program
210	Title screen
220-280	Main menu
290-330	Search cards menu
340-480	Get a folder
490-570	Toss a folder
580-660	Set search parameters
670-860	Print folder index
870-1090	Save indexed folder
1100-1120	Exit program
1130	Erase bottom of screen
1140-1170	Display search parameters
1180-1190	Draw a box
1200-1260	Input file name
1270	Print "WORKING"
1280	Error routine
1290	Character data

DIRECTORY OF VARIABLES

Variables	Functions
A	General utility
B	General utility
C(,)	Pointers to the cards that are found
C$(,)	Data for one card
FN$( )	Folders' file names
FS$( )	Folders' search item
I	Loop counter
J	Loop counter
K	ASCII of keyboard input
MX	Maximum number of cards
MXF	Maximum number of folders
NF	Current number of folders
PR	Flag for Printer/Screen output
S	Keyboard status
S$	General utility
SE$	Item searched for
SE( )	Fields searched through

Serf City

REMARKS

Efficient allocation of the workforce is critical to the success of a *Serf City* kingdom, as it is to any economy. The TI version of *Serf City* has a built-in algorithm that allocates workers to various occupations according to a specific hierarchy. The Design Focus illustrates the structure of this algorithm.

First the program automatically deducts soldiers from the workforce. It then tries to fill the fields with sufficient workers to harvest the wheat. To determine the number of people to allocate to field work, the program takes the number of bushels of wheat to be planted, then places one person in the fields for each bushel—if the population is sufficient. If the population is not large enough to plant all the seed, the program places all of the available workers in the fields, however, it reduces the number of bushels planted to equal the number of workers.

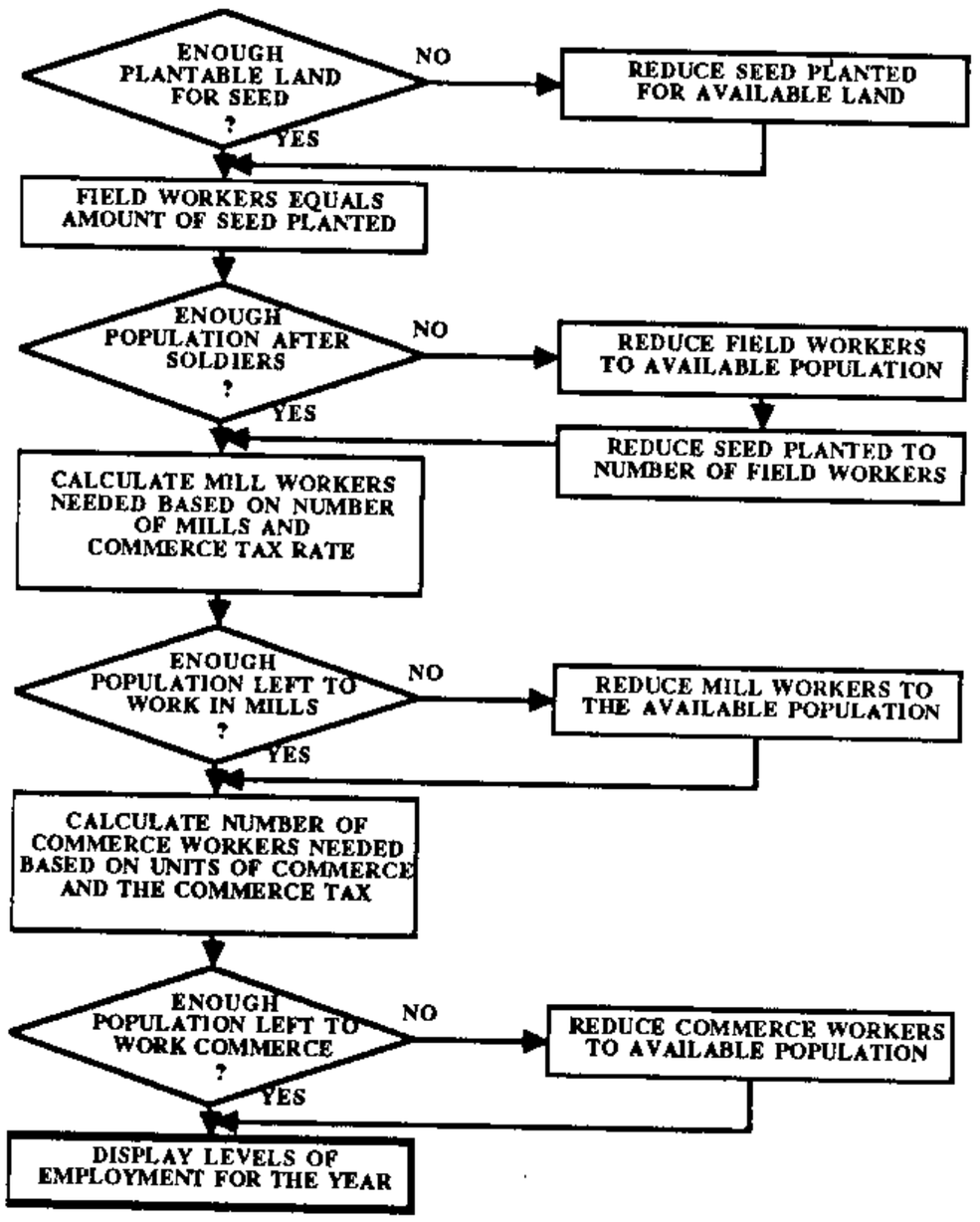
After the program pulls field workers and soldiers from the population, it staffs the mills. The amount of people that the program can allocate to the mills depends on the remaining population, the number of mills, and the commerce tax rate. A high commerce tax rate restricts the number of people that the program can employ in each mill.

After fully staffing the mills, the program employs any remaining population in commerce. The program determines the amount of people that can be employed in commerce according the the remaining population, the number of units of commerce, and the commerce tax rate.

If the kingdom has additional population after the fields, mills, and commerce are fully staffed (in that order), the program considers this remainder unemployed.

DESIGN FOCUS

Serf City Employment



LISTING ANNOTATIONS

Line Nos.	
100-190	Program header.
200-340	Title screen and initialization
350-390	Players' names
400-510	Main control loop
520-620	Main data screen
630-820	Map screen
830-980	Move cursor
990-1250	Add unit of land to map
1260-1460	Resolve conflict
1470-1570	Add mill
1580-1680	Add unit of commerce to map
1690-1770	Update string array S$( )
1780-1850	Wheat option menu
1860-1930	Feed people
1940-2010	Plant seed
2020-2090	Store wheat
2100-2170	Buy wheat
2180-2250	Sell wheat
2260-2380	Input routine
2390-2440	Bad input messages
2450-2730	Hire/fire soldiers
2740-2970	Set tax rate routine
2980-3760	Year-end report routine
3770-3910	End-of-game routine
3920-4020	Print main data screen
4030-4080	Print map screen
4090-4270	Initialize program variables
4280-4290	Map screen format data
4300-4310	Character definition data
4320-4340	Time delay routine
4350-4370	Key scan routine

DIRECTORY OF VARIABLES

Variables	Functions
P$( )	Players' names
S$( )	Map screen
K(,)	Kingdom information
A$	Input routine utility
B$	Input routine utility variable
I1$, I2$	Legal keys list
A	Value returned from input routine
A0, A1, A2, A3	Constants for 0, 1, 2, and 3
A4, A5, A6, A7	Constants for 4, 5, 6, and 7
A8, A9	Constants for 8 and 9
C	Character under cursor
C1	Character above cursor
C2	Character below cursor
C3	Character left of cursor
C4	Character right of cursor
CH	Character placed into map
DN	End-of-game flag
E	Employment in fields
F	Employment in mills
G	Employment in commerce
P	Prosperity of people
P1	Current player
P2	Opposing player
R	Random number
S	Status from scanning keyboard
VW	Value of wheat during year
W	Key returned from key scan
X	Screen coordinates
Y	Screen coordinates
YR	Current year of simulation
Z	Utility loop counter

Cell Mates

REMARKS

One of the most complex algorithms in the TI version of *Cell Mates* is the one that determines the level of protein. Protein can either be produced by the ribosomes, or it can be imported from outside the cell. At the start of the simulation, the protein level is at 50% of maximum. The protein supply must reach this maximum for the cell to reproduce. It is not easy to double the supply of protein, however, because protein must be exported. The Design Focus illustrates the algorithm that we use to control these processes.

Ribosomes are the primary source of protein. Each ribosome can produce a limited amount of protein, however, so the number of ribosomes present determines the amount of protein that can be produced in any given period. This control factor is represented in the Ribosome Level parameter in the flow chart. The routine multiplies the number of ribosomes by the Protein Energy (the amount of ATP available for protein production) and calculates the square root of the result to arrive at the optimum protein production value. The routine then takes other controlling factors into consideration. These factors include the cell's age, the internal temperature deviation from optimum, and the amount of available nutrients—all of which have a detrimental effect on the cell's ability to produce protein. The routine multiplies the optimum increase in protein production by adjustments based on these three factors to determine the increase in protein level.

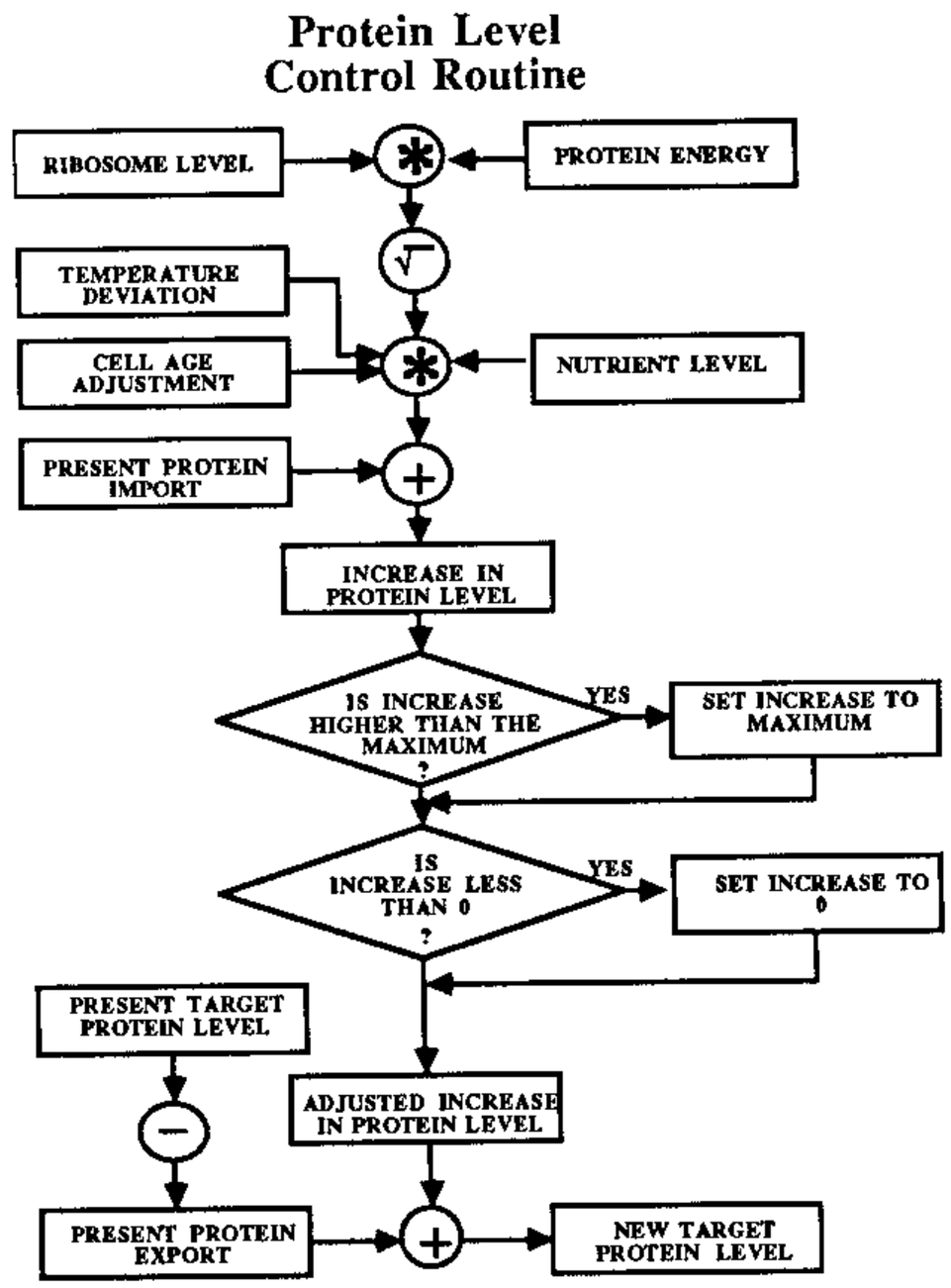
The routine then adds the present protein import and executes a check on the result for the maximum and minimum protein increases in order to avoid unrealistic results.

The value we end up with after all of this calculating is the amount of change that is to take place in the target protein level, or the level that protein would reach if change in level were instantaneous. The routine adds this value to the present target protein level to determine the new target protein level.

Three equations contain the entire algorithm and are located in lines 440, 470, and 480.

HCM Glossary terms: ATP, ribosomes

DESIGN FOCUS



LISTING ANNOTATIONS

Line Nos.	
100-190	Program header
200-240	Program initialization
250-280	Main control loop
290	Branch to end-of-program routine
300-320	Option to exit game
330-480	Vital statistics calculations
490-530	Check vital status
540-670	Cell dies from various causes
680-710	Cell reproduces
720-730	Option to play again
740-770	Update meter positions
780-790	Update mitochondrion motion
800-850	Administer ATP
860	Key scan routine
870-950	Character shape data
960-1020	Screen format data
1030-1040	Initial values for the C( ) array
1050-1090	Playing screen set-up
1100-1110	Initialize variables

DIRECTORY OF VARIABLES

Variables	Functions
C( )	Graphics information
A$	Input routines
I$	Legal keys
A	Age deviation factor
AGE	Age of the cell
ATP	Amount of ATP administered
D	Flag for reason for game termination
DST	Destination of mitochondrion
DX	X coordinate of mitochondrion's destination
DY	Y coordinate of mitochondrion's destination
K	ASCII value of the key pressed
MF	Motion indicator
MX	The speed of the mitochondrion on the X axis
MY	The speed of the mitochondrion on the Y axis
P	The position of the key pressed within the list of legal keys
RN	Random factor: nutrient supply
RT	Random factor: internal temperature
ST	Keyboard status
TA	Temperature deviation multiplier
Z	Utility loop counter

NanoAssembler

REMARKS

The TI version of the *NanoEditor* uses the POS function to control program flow when the user selects a particular editing command. The POS function searches through a string for a smaller string within it, and returns (gives) the numeric value of the POSition of the smaller string. In this way, line 430 sets OP equal to the POSition of the S$ (the user's selected command) in CMD$ (the legal commands—"ADIEL").

As shown in the Design Focus, if OP is not a 1, the program prompts the user for the number of the line to be Deleted, Edited, Inserted, or Listed (see lines 460-600 in the program). If OP is set to 1 (for the Add command), the program already knows the line number to be added, so it branches around this section. In either case, OP then controls the ON-GOSUB in line 610 according to the chosen command.

HCM Glossary terms: branch, string.

LISTING ANNOTATIONS — *NanoEditor*

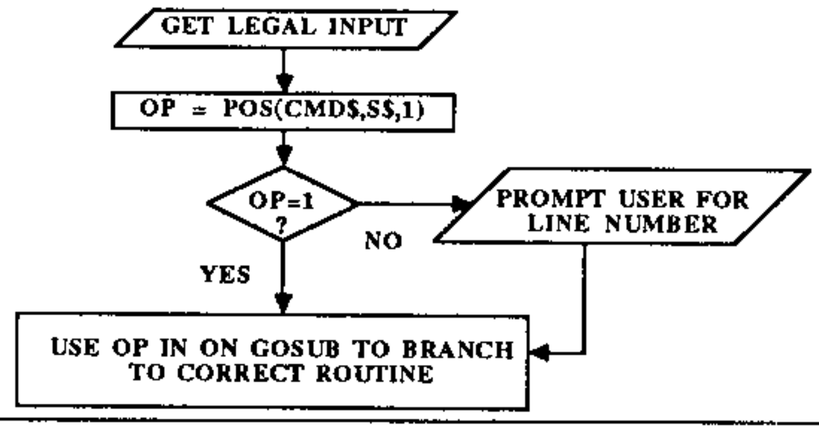
Line Nos.	
100-190	Program header
200-230	Initialize program
240-290	Main menu
300-620	Main editing loop
630-830	Add a line
840-950	Delete a line
960-1140	Edit a line
1150-1340	Insert a line
1350-1570	List lines
1580-2010	File routines
2020-2140	Print-lines routine
2150-2240	End program
2250-3100	Input routine
3110-3190	Print-menu routine
3200-3220	Assure number is three digits
3230-3270	Set parameters for input routine
3280-3310	Print line number
3320-3360	Menu data
3370-3400	Read keyboard
3410-3430	Delay loop
3440-3470	Initialize-program subroutine

LISTING ANNOTATIONS — *NanoAssembler*

Line Nos.	
100-190	Program header
200-270	Initialize program
280-450	Program driver
460-820	First pass
830-1120	Second pass
1130-1170	Get keypress and return to main driver
1180-1310	Error routines
1320-1460	File routines
1470-1520	Prepare assembler output
1530-1590	Strip line of remarks and spaces
1600-1740	More file routines
1750-1820	Look for a legal mnemonic
1830-1930	Set up printer
1940-1960	End program
1970-2700	Check mnemonics and directives
2280-2330	Convert a number from binary
2340-2390	Convert a number from hexadecimal
2400-2700	Evaluate data field
2710-2830	Menu routine
2840-2880	Get a Y or N keypress
2890-2950	Delay loop
2960-3030	Initialize-program subroutine
3040-3050	Mnemonics and assembler directives

DESIGN FOCUS

Using the POS Function For Program Flow



DIRECTORY OF VARIABLES — COMMON

Variables	Functions
A	Store the tab position
A$	Utility variable
FL$	File name
I,IT	Loop counters
K	Keyboard-input variable
LI$( )	Array that stores source-code lines
LN	Current line of text
NI	Number of items in current menu
NL	Current number of source-code lines
PR$	Printer parameter
S	Keyboard status

DIRECTORY OF VARIABLES — *NanoEditor*

Variables	Functions
B	Lowest legal ASCII value for input
C	Screen column position during input
C$,CN$	String of possible ASCII inputs
CMD$	Possible editing commands;"ADIEL"
DC	Disk/Cassette flag
E	Position of last character input
ESC	Flag used for escape key—(FCTN) 9
G	Character read from screen
L	Maximum length of input
NS	Used in input routine
OP	Option chosen from editing prompt
R	Screen row position during input
RW	Limit on row postion
S$	Utility string
SL	Save/Load flag
T	Highest legal ASCII value for input
X	Current cursor postion
CA,KS,LN$,Y,Z	Utility variable

DIRECTORY OF VARIABLES — *NanoAssembler*

Variables	Functions
AD( )	Address array
CA	Current address
EA	Ending address of a line of code
HX$	String of hex characters
LB	Current label number
LB$( )	Array of labels
LFLAG	Flag used to spot label
LL	Length of source-code string
N$	Used to convert numbers to decimal
NM$	Legal numeric-character string
NN( )	Number of nibbles for mnemonics
OP	Current mnemonic (or op-code)
OP$( )	Array of mnemonics (or op-codes)
OT$	Output string
OU$	Output string
PR	Printer flag
T$	Temporary string
TL$	Current source-code line
V	Value of data in right-most field
X,Y	Utility variable

HCM Glossary

VOLUME 5 NUMBER 6

algorithm – A set of rules or procedures used to solve a problem.

amplitude – The height of a waveform measured from peak to peak.

argument – A parameter that is passed by a BASIC function.

ASCII – (American Standard Code for Information Interchange) The computer code most commonly used to represent upper- and lower-case letters, numbers, symbols, and punctuation marks.

assembler – A program that translates or assembles source code into machine language.

ATP – (Adenosine Triphosphate) A product of the mitochondria that serves as the energy source for all cells.

attack – Rate at which a signal reaches its peak level on a traditional ADSR envelope generator.

azimuth – The angular distance of an object measured in degrees clockwise (to the right) from true north.

BCD – (Binary Coded Decimal) A method for representing decimal numbers in binary.

binary numbers – A base-2 numbering system in which the only symbols used are 0 and 1.

bit – (binary digit) The most basic unit of information that the computer uses. Each bit is an electronic impulse, that, combined with other such impulses fed into the computer's circuitry, forms data.

branch – A departure from the sequential execution of program instruction—usually due to the test of a condition.

byte – A sequence of 8 bits used to represent one character.

call – To transfer control to a specified closed subroutine.

component – One of the sine waves that add up to produce a complex periodic waveform.

CONFIG.SYS file – A file on a PC DOS disk that, when DOS boots, configures the way that the operating system views memory, I/O devices, etc.

conjunction – The apparent meeting of two celestial bodies in the same degree of the zodiac.

cross-reference – An organized list of items from several different sources that refers you to the item's folder, card number, or whatever applies.

cursor – A character or mark shown on the video screen to designate where the next character will be printed.

cycle – One complete alternation of a periodic waveform.

decay – Rate at which a signal falls from peak level to sustain level on a traditional ADSR envelope generator.

declination – The angular distance of a celestial body south or north of the celestial equator (the plane of Earth's equator extending to infinity).

dither – To use a combination of dots to simulate a tone in black-and-white or color.

DNA – (deoxyribonucleic acid) Nucleic acids localized in the nucleus of eucaryotic (nucleus-bearing) cells or in the lining of procaryotic (without nuclei) cells as the molecular basis of heredity.

DOS – Disk Operating System.

dummy argument – An argument passed purely to satisfy the syntax of a statement.

ecosystem – An ecological community together with its environment, considered as a unit.

endoplasmic reticulum – A convoluted membrane extending out from the nucleus of a living cell.

envelope – A graphic representation of the envelope generator's output as a function of time.

enzyme – Any of complex proteins produced by living cells that catalyze specific biochemical reactions.

field – A specified area for a particular category of data.

file – A collection of related records (data items), treated as a unit.

flag – A variable or memory location that contains a value that represents a condition the program needs to test.

frequency – The rate of repetition of a waveform. It is perceived as pitch in the audio range.

function – In BASIC, a statement used to return a value. The value often depends on the type of argument passed to it.

fundamental – The principal frequency of a waveform. In the audio range, it is the perceived pitch.

Golgi apparatus – An organelle that packages and transports materials in a living cell to be secreted to the plasma membrane for internal use or expulsion.

hard-code – To specify the values of variables or constants within the program code itself so that they are not subject to change by user input.

harmonic – A component of a periodic waveform. Its frequency is a positive integral multiple of the fundamental frequency.

hexadecimal numbers – A base-16 numbering system using decimal digits 0 through 9 and the letters A through F.

high bit – The left-most bit of a binary number; e.g., the eighth bit of a byte.

horizontal plot position – The horizontal position on the screen at which the next point will be plotted.

index – To selectively choose and organize a group of data (or an organized list of items that refers you to the item's folder, page number, or whatever applies).

interrupt request – A signal that directs the computer away from the program sequence.

iteration – One complete cycle through a process that is repeating.

level of recursion – A routine's depth into a recursive process.

linear – An even progression from one state to another with no deviation in course.

logic operator – A BASIC function used to perform a logical operation.

lysosome – A cell's waste disposal unit—basically a sac pinched off of the Golgi apparatus.

machine language – A native language of the microprocessor in a computer expressed in terms of binary ones and zeros.

mask – A fixed group of characters used to control the retention or elimination of parts of another pattern of characters.

merge – To combine two or more files, or portions of files, into one.

metabolism – The physical and chemical processes of an organism that produce energy and result in the production, maintenance, and destruction of life.

MIDI – (Musical Instrument Digital Interface) Specified protocols for transmitting digital information from one synthesizer, sequencer, computer, signal processor, etc., to another.

mitochondrion – An organelle that produces ATP.

mnemonic – A code or symbol that helps people remember something specific—usually made up of letters from the word or phrases it represents.

nibble – 4 binary digits (bits) of data—a half byte.

noise channel – The channel responsible for producing noises.

nucleus – The central core of a cell containing both the cell's heriditary DNA, and the message-bearing RNA (ribonucleic acid).

organelle – A specialized part of a cell with a specific organ-like function.

overtone – A frequency component of a waveform, that is higher than the fundamental.

op-code – (operation code) The part of a machine instruction that tells the computer which function to perform next.

parameter passing – Sending (giving) a parameter to a particular process.

periodic noise – One of 4 pulse waves produced by the TI-99/4A sound chip.

precession – A gradual change in the angle of the Earth's axis due to "wobbling" in the Earth's rotation.

proper motion – Apparent movement over time of a star in the celestial sphere due to the actual motion of both the star and our sun through space.

protein – Any of several complex compounds that are required for all life processes.

pulse wave – A wave that alternates instantaneously between the two DC volage levels.

pulse width – The percentage of time that a cycle of a pulse wave spends in the higher of its two voltage states.

RAM – See Random Access Memory.

Random Access Memory (RAM) – The set of hardware locations in a computer where programs and data are stored.

random number generator – A circuit within the computer that generates pseudo-random numbers.

record – A collection of information consisting of one or more related items.

recursion – A seemingly circular process in which a procedure calls itself repeatedly.

release – Rate at which a signal falls from sustain level to zero level on a traditional ADSR envelope generator.

respiration – The process of yielding energy from oxidative reactions in living cells.

resultant – The complex periodic waveform resulting from the addition of two or more sine wave components.

ribosome – An organelle consisting of protein and ribonucleic acid (RNA) that catalyzes the construction of proteins.

right ascension – The east/west coordinate of a celestial body along the celestial equator.

ring modulation – Mathematically combining two frequencies by outputting their sum and difference and suppressing the original frequencies.

sector – On magnetic storage medium, a section of a track constituting a unit of data storage.

SID - Sound Interface Device chip in the Commodore 64 computer.

sine wave – A single frequency oscillation that produces a pure fundamental waveform without harmonics.

statement separator – A character that separates BASIC statements.

string – A consecutive set of similar data items—usually bits or characters.

subroutine – A program segment or module that performs a specified function (e.g., debugging).

sustain – The level of a signal after attack and decay, and before release on a traditional ADSR envelope generator.

symbiotic – Two or more different organisms in a close association that is of mutual benefit.

synchronization – The linking of two oscillators such that the start one oscillator cycle triggers the start of the second oscillator's cycle. One oscillator acts as the master and the other acts as a slave.

text file – A file containing textual information that is stored in ASCII format.

tile – A technique employed in conjunction with the IBM BASIC **PAINT** command that allows the user to place a specified graphic pattern onto a particular area of the screen.

TOD – An acronym for the Time-Of-Day clock on the Commodore 64 computer.

video buffer – A section in memory reserved for video processing.

waveform – 1. A graphic representation of the shape of a wave. 2. A signal with periodic fluctuations produced by an oscillator.

weight – The value of a bit determined by its position.

white noise – Random acoustical or electrical signals in which the intensity is the same at all frequencies within a given band.

zodiac – An imaginary belt in the sky about 18 degrees wide along which the sun and all planets except Pluto appear to travel.

SENSATIONAL SOFTWARE GIVEAWAY

3 FANTASTIC OFFERS

1 YOUR CHOICE OF THIS ISSUE'S PROGRAMS AT A REMARKABLE PRICE— ONLY $4.95!

plus $1.00 shipping & handling

Price will change to $6.95 upon becoming a "Back Issue."

FOR SUBSCRIBERS & NEWSSTAND PURCHASERS ONLY*

To participate in our monthly Software Giveaway, you need to be a bonafide purchaser of ***Home Computer Magazine***.
You will receive all the programs ON TAPE™ or ON DISK™ (which have versions for your selected machine) whose listings appear in this issue.

IMPORTANT: *Please complete the order form located in the center of this magazine. Enclose it in an envelope along with $5.95 ($6.95 in Canada, $10.95 Foreign Airmail). Payment must be made by check, money order, or VISA/MasterCard. Proof of purchase (subscriber label number, sales receipt or any reasonable facsimile thereof) must also accompany this form.*

**Non-subscriber and non-purchaser price is $12.95 in the U.S.*

SORRY, WE CANNOT ACCEPT TELEPHONE ORDERS FOR THIS SERVICE.

*FREE SOFTWARE!

When Subscribing To

HOME COMPUTER® magazine

Subscribe or Renew today, and with your paid subscription you will receive **FREE** software—**ON TAPE™** or **ON DISK™**. With a 1-year subscription to ***Home Computer Magazine*** you get **2 FREE issues** (a $11.90 Premium Value) of **ON TAPE™** or **ON DISK™**. Subscribe for 2 years and receive **4 issues** (a $23.80 Premium Value). And with a 3-year subscription we'll give you **6 full issues** (a $35.70 Premium Value) of this convenient software on cassette tape or floppy disk.

*DON'T HAVE A COMPUTER? TAKE A RAINCHECK!

We'll give you a raincheck for the FREE software so that when you buy a computer, we will send you your choice of ON TAPE™ or ON DISK™ as a FREE BONUS as stated in this offer.

And You Save Up To 40% Off The Single-Copy Price of the Magazine!

SAVE EVEN MORE!

3 AND ENJOY THE CONVENIENCE OF A PROGRAM SUBSCRIPTION*

By subscribing to **ON TAPE™** or **ON DISK™** you will save money off the single-copy price **and** receive the same high-quality programs published in each issue of the magazine—**delivered right to your door!**

This cassette tape or floppy disk program service is the convenient, accurate, and affordable way to save hundreds of typing hours.

The Perfect Addition To Your Magazine Subscription!

- Program Subscribers Get Disks & Tapes First To Coincide With Magazine Arrival.
- Program Subscribers Get Special Updates As Available.

*DOES NOT INCLUDE MAGAZINE

TO ORDER: Please review these 3 Fantastic Offers and use the ORDER FORM located in the center of this magazine.

1. **RECEIVE ALL** the programs in this issue on magnetic media (see below). Indicate your choice of computer media on the order form and enclose $5.95 ($6.95 in Canada, $10.95 Foreign Airmail).
   THIS PRICE GOOD FOR THE CURRENT ISSUE ONLY
   Proof of purchase must be included with order.

2. **FREE Software** with your subscription to ***Home Computer Magazine*** for the terms listed below:
   - 1-yr (10 issues) $25 — ***PLUS 2 FREE*** ON TAPE or ON DISK
   - 2-yr (20 issues) $45 — ***PLUS 4 FREE*** ON TAPE or ON DISK
   - 3-yr (30 issues) $63 — ***PLUS 6 FREE*** ON TAPE or ON DISK

   Canada add $7 per year; Foreign Surface add $21 for 1-yr magazine subscription. Free software offer available in U.S. & Canada only.

3. **SAVE MONEY** and time with a **Program Subscription*** to **ON TAPE™** or **ON DISK™** for the terms listed below:
   - 8 ISSUES—*ONLY $44.95* "An Extension To Your 2 Free Issues of Software"
   - 10 ISSUES—*ONLY $49.95* "A Full Year Of Program Convenience"

   *Does not include magazine.

Canada add $5 for software subscription.
Software subscription not available in other countries at this time.

PLEASE LOCATE THIS BOX ON THE CENTER BOUND-IN ORDER FORM AND INDICATE YOUR CHOICE OF COMPUTER MEDIA

DISK VERSION						CASSETTE VERSION		
APPLE	ATARI	C-64	IBM PC	IBM PCjr	TI	ATARI	C-64	TI

Or ask for a raincheck for your FREE software.

SGV1550885

ALL PROGRAMS IN THIS MAGAZINE

ONLY $4.95* ON DISK OR TAPE!

The same high-quality Apple, Atari, Commodore, IBM, and Texas Instruments programs with type-in-and-RUN listings in this issue are now available ON DISK™ or ON TAPE™ to newsstand purchasers or subscribers of this magazine.

For only $4.95* plus $1.00 for shipping & handling, you receive all the programs for your particular brand of computer—rushed right to your door by First-Class mail.

—Truly A "Software Giveaway!"

For More Information, See ① Inside Rear Cover.

To Order, Use The Bound-In Order Form At Center Of Magazine. Offer & Prices Subject To Change Without Notice.

* Current Single-Issue Price Only — See Center Bound-In Order Form For Back-Issue Prices & Special Program Subscription Offer.