SAVE SUPER DEALS SALE

(312) 382-5244 TO ORDER

COMMODORE 64 COMPUTER

SALE $98.00* List $299

*You pay only $98.00 for the Commodore 64 when you purchase a **Pocket Writer 64** word processor for only $39.95. List $299.00 **SALE $98.00**

1541 DISK DRIVE

SALE $179.95 List $249

To run all that software and add to the prestige of your system, you'll want the 1541 disk drive. Runs all of the 3,000 plus pieces of software for the Commodore 64. List $249.00 **SALE $179.95**

COMMODORE 128 COMPUTER

SALE $299.00 List $399

This powerful 128K computer has three modes. One for C-64 software, one for C-128 software, and one for CP/M business software. List $399.00 **SALE $299.00**

340K 1571 DISK DRIVE

SALE $259.00 List $349

Double sided, single disk drive for the C-128 allows you to use the C-128 mode plus CPM mode. 17 times faster than the 1541, and runs all the 1541 formats. List $349.00 **SALE $259.00**

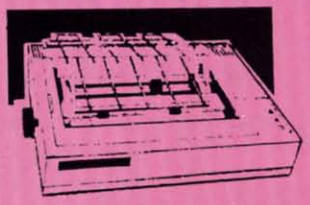
BIG BLUE PRINTER

SALE $ 39.95 List $199

This is the affordable printer you have been waiting for! This 8½" letter size, 80 column dot matrix, heat transfer printer features upper and lower case, underline, graphics, word processing, and much more. List $199.00 **SALE $39.95**

PREMIUM QUALITY 150-170 CPS COMSTAR AERO 160 PRINTER

SALE $199.00 List $499



The Comstar Aero 160 gives you a 10" carriage, 150-170 CPS, 9x9 dot matrix with double strike capability for 18x18 dot matrix (near letter quality), high resolution bit image (120x144 dot matrix), underline, back spacing, left and right margin setting, true lower descenders with super and subscripts, prints standard, block graphics and special characters. It gives you print quality and features found on printers costing twice as much! (Centronics Parrallel Interface) List $499.00 **SALE $199.00**

PRINTER/TYPEWRITER COMBINATION

SALE $199.95 List $299

Superb Silver Reed letter quality daisy wheel printer/typewriter combination, just a flick of the switch to interchange. Extra large carriage, typewriter keyboard, automatic margin control, compact, lightweight, drop in cassette ribbon! (90 day warranty) Centronics Parrallel Interface is built in. List 299.00 **SALE $199.95**

COMSTAR 1000 PRINTER

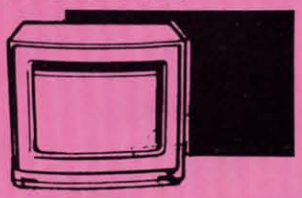
SALE $179.95 List $349

This is one of the best values today. Print letters, documents, ect., at 100 characters per second or 20 cps in Near Letter quality mode. Features are dot addressable graphics, adjustable tractor and friction feed, margin settings, pica, elite, condensed, italics, super & subscript, underline, bold, and double strike. Commodore Interface Included. List $349.00 **SALE $179.95**

12" 80 COLUMN MONITOR

SALE $ 79.95 List $129

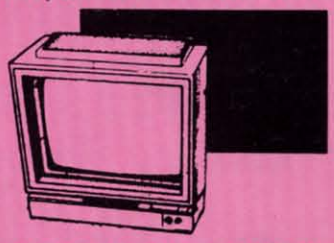
Super High Resolution green screen monitor. 80 columns x 24 lines, easy to read, plus easy reach adjust knobs. Supreme Quality. Fantastic value. (cable $9.95) List $129.00 **SALE $79.95**



13" COLOR MONITOR

SALE $139.95 List $329

This premium quality monitor comes with speaker and audio hook up, High Resolution, clear screen, 40 columns x 24 lines, front panel controls and more. Also be use with VCR. One year Ltd. warranty. (cable $9.95) List $329.00 **SALE $139.95**



14" RGB & COMPOSITE COLOR MONITOR

SALE $237.00 List $399

Must be used to get 80 columns in color with 80 column computers (C128 - IBM - Apple) (RGB cable $19.95) Add $14.50 shipping. List $399.00 **SALE $237.00**

TV TUNER

SALE $ 49.95 List $130

This tuner has dual UHF/VHF selector switches, mute, automatic fine tuning and computer/TV selector switches. Hook it up between your computer and monitor! Inputs included for 300 ohm, 75 ohm, and UHF. List $129.95 **SALE $49.95**

CALL BEFORE YOU ORDER:
• OUR PRICES MAY BE LOWER & AND WE OFFER SPECIAL SYSTEM DEALS •

ATTENTION

COMPUTER CLUBS • DEALERS

We offer big volume discounts!

CALL TODAY

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

(312) 382-5244 to order

Reader Service No. 269

Commodore 64

MODEMS

Commodore 64

Sale

SAVE

300 & 1200 BAUD

Reach Out and Access Someone

Educational Courses• Financial Information • News Updates and Information
Banking at Home • Popular Games • Electronic Shopping • Research and Reference Materials

300 BAUD
List $99.00
Sale
$29.95
Coupon $24.95

Telecommunications

300 BAUD
List $99.00
Sale
$29.95
Coupon $24.95

Lowest Price In the U.S.A.

FOR CHILDREN·ADULTS·BUSINESS

Complete Auto Dial Telecommunications Package

"The only telecommunications package you will ever need."

(Exclusive Easy To Use Features)

• 300 Baud Modem • Auto Dial • Auto Answer • Upload & Download

The Complete Telecommunications Package offers you all this plus...

- Dialing from Keyboard
- On-line Clock
- Capture and Display High Resolution Characters
- Download Text, Program or Data Files
- Stores on Disk Downloaded Files
- Reads Files from Disk and Uploads Text or Program Files
- Select Any Protocol (access almost any computer or modem)
- Plus Much, Much More

We are so sure this is the only telecommunications package you will need we will give you 15 days Free Trial.

300 Baud Modem List for $99.00 on Sale for $29.95 with Coupon only $24.95

1200 BAUD MODEM
Sale
$79.95
List $199.00

Volks 6480

1200/300 Baud Auto Dial/Auto Answer

By Anchor Automation

A Leader in Modem Technology

Features Same as Above Plus...

• AutoCom 64 and 128 Software • File & Program Transfer • Tone and Pulse Dialing • Monitor Speaker

5 YEAR
Limited Warranty

1200 BAUD MODEM
Sale
$79.95
List $199.00

5 YEAR
Limited Warranty

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6¼% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA. Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Prices & availability subject to change without notice.
VISA — MASTER CARD — C.O.D. C.O.D. on Phone orders only

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 269

Famous National Brand

Commodore® • Atari® • Apple® • IBM®

We Like this Printer so much
We Bought Out the Factory.

List $199.00 **Sale $39.95**

80 Column Printer Sale

List $199.00 **Sale $39.95**

• Word Processing • Program Listings • Graphics • Quiet Operation • Upper and Lower case • All points addressable Graphics • Underline • Enlarged • Much much More

Complete your set up with a Printer Stand **SALE $14.95** and Complete Printer Care Kit **SALE $19.95**
(Care Kit will add time to your printers life span.)

Super Print Quality

This printer was made by **Canon®** for the IBM PC and PCjr. The Big Blue printer comes ready to hook up to the serial port of the **IBM®** PC jr. Plus with low cost adapter cables you can connect the Big Blue printer to the **Commodore®** 64 ,128, **IBM®** PC, **IBM®** XT, **IBM®** AT, **IBM Compatibles , Apple®** II,IIe,IIc, **Apple Compatibles, Atari®**, plus many more.

• 90 Day Immediate Replacement Warranty **• 15 Day Free Trial-Satisfaction or Money Back**

Now you can have a full fledged 8½'' letter size 80 column printer for less than the cost of a large box of paper. This printer uses advanced dot matrix, heat transfer technology to print upper and lower case (with true lower descenders), underline, enhanced, all points addressable graphics (works with Printshop) plus More. Print out pictures, program listings, wordprocessing pages, graphics and more. Perfect for the homeowner or student and definitely affordable. Fantastic Printer at an unbeatable price. List $199.00 **Sale $39.95**

Paper (2 Rolls) List $19.95 **Sale $5.95**

Intelligent Commodore Interface— Allows you to connect the Big Blue printer to the printer port of the Commodore 64 and 128 computer. Print Commodore graphics, use Printshop, Word processors and more... List $49.95 **Sale $19.95**

Intelligent Atari Interface— Allows you to connect the Big Blue printer to Atari computers (except 1200). Print Atari graphics, Printshop ,word processors, and more... List $49.95 **Sale $19.95**

RS-232 Adapter— Adapts the Big Blue printer to be used with any IBM® PC, AT, XT.PCjr. Apple® II series RS-232 port. List $19.95 **Sale $9.95** **Laser128, Apple IIc** interface— List $24.95 **Sale $12.95**

Printer Stand List $24.95 **Sale $14.95** - Complete Printer Care Kit List $29.95 **Sale $19.95**

IBM, Apple, Canon, Commodore, Atari are trademarks of International Business Machines, Apple Computer, Canon Inc, Commodore Business Machines, Atari Inc, Respectively.

Add $7.50 for shipping, handling and insurance. Illinois residents please add 6½% tax. Add $15.00 for CANADA, PUERTO RICO, HAWAII, ALASKA and APO-FPO orders. All orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Prices & Availability subject to change without notice.
VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 269

Color Monitor Sale

Sale

16 COLORS

(Premium Quality)

- Built in Speaker & Audio
- For Video Recorders
- For Small Business Computers
- Apple - Commodore -Atari - Laser 128 -etc.
- One Year Limited Warranty

(Premium Quality)

- Beautiful Color Contrast
- High Resolution
- Sharp Clear Text
- Anti-Glare Screen
- 40 Columns x 24 Lines
- Front Panel Controls

List $329.00

Sale $139.95*

Add $14.50 Shipping

80 Columns **RGB** Super High Resolution

13" Color Computer Monitor

*C64/Atari composite cable $9.95

* C128 RGB/Composite 80 column cable $19.95.

14" RGB & Composite Color Monitor

Allows use of C-128 and C64 mode - composite and 80 column RGB mode. Must be used to get 80 columns in color with 80 column computers. Specially designed for use with the C128's special composite video output, plus green screen only option switch. (add $14.50 shipping)

List $399.00 **Sale $237.00***

12" 80 Column Green Monitor

Super high resolution composite green or amber screen monitor. 80 columns x 24 lines, easy to read. Fantastic value. Limited Quantities.

List $129.00 **Sale $79.95***

9" Samsung Hi Res Green Screen Monitor

Super High Resolution 80 column monitor perfect for Apple & Laser 3000/128 computers. Fantastic Value. Very Limited Quantities.

List $129.95 **Sale $59.95***

Turn Your Monitor into a TV Set Without Moving Your Computer

Elegant TV Tuner with dual UHF/VHF selector switches goes between your computer and monitor. Includes mute, automatic fine tuning and computer-TV selector switches. Inputs included for 300 ohm, 75 ohm, and UHF. Can be used with cable TV and VCR's. Fantastic Value. Limited Quantities. (Includes loop antenna for UHF & RCA connecting cables) (Add $3.00 Shipping. Plus $3.00 for APO/FPO).

List $99.95 **Sale $49.95**

Coupon $39.95

15 Day Free Trial - 90 Day Immediate Replacement Warranty

• LOWEST PRICES • BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6⅓% tax. Monitors can only be shipped to Continental U.S. addresses only. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Prices & Availability subject to change without notice.

VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only

No Monitors A.P.O.-F.P.O.

PROTECTO

We Love Our Customers

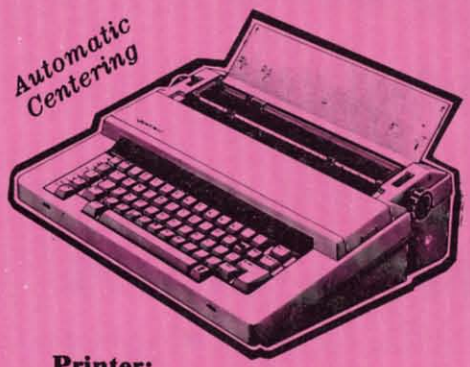
22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 269

Daisy Wheel Printer/Typewriter

• Superb computer business printer combined with the deluxe electronic typewriter • Two machines in one — just a flick of the switch • Superb Letter Quality correspondence • 12" Extra Large carriage • Drop in cassette Ribbon • Precision Daisy Wheel printing • Key in Buffer • Centronics Parallel interface



Sale $199.95

List $299

Now you can have the advantages of a letter quality Daisy Wheel printer and the convenience of a typewriter for one low cost. Use your wordprocessor to type the letters, then with just a flick of the switch, your typewriter can type the envelopes. This is a fantastic Printer and a fantastic Typewriter. Every student and home business needs this machine. List $299.00 **Sale $199.95**

Printer:

Paper width	12 inches
Printing width	10 inches
Cassette ribbons	Correctable film, one-time film, and fabric
Correcting tapes	Lift-off and cover-up
Line space lever	0,1,1-½,and 2
Keyboard selector	I and II
Print pitch	10,12, and 15

Keyboard:

Keys/characters	44/96
Automatic carrier return	
Automatic correction	1 Line
Automatic underline	
Relocate key	
Tabulation	
½ back space key	

Students • Teachers • Business • Home Users

Electronic Compact Printer/Typewriter Now with "Spell Check"

• Supreme quality printer/typewriter with all the features of the above model plus more • 90,000 word dictionary • LED readout flags mistakes before they reach print • 1 line/240 character automatic correction • Free 4K memory card • Edit 2 to 4 pages from typewriter memory •

Sale $299.95

List $399

Get the edge over classmates, colleagues and competitors with this fantastic printer/typewriter with **"Spell Checker"**. With over 90,000 words in its dictionary and the capacity to add 510 more of your choice, you may never misspell again. No more re-typing, when an error is identified, three beeps sound. The LED readout has a one line buffer so mistakes are corrected before they reach print. This means you're virtually ensured of nothing less than perfectly typed letters and documents with no misspellings! Also, you receive a **free** 4K memory card with your purchase. This card equips the typewriter with 4,000 characters of storage which is equal to approximately 2 to 4 pages of typing. Edit right on your typewriter using your LED screen. A $45.00 list value for FREE! Fantastic! List 399.00 **SALE $299.95**

Extra Replacements for both:

Ribbons $9.95

Daisy Wheels $12.95

FANTASTIC BUYS ON BOTH ITEMS FROM SILVER REED

90 Day Manufacturer Warranty $12.00 Shipping

Reader Service No. 269

Famous Comstar National Brand

Universal Printer Stand $14.95

10" Printer Sale

Complete Printer Care Kit $19.95

Includes Commodore Interface

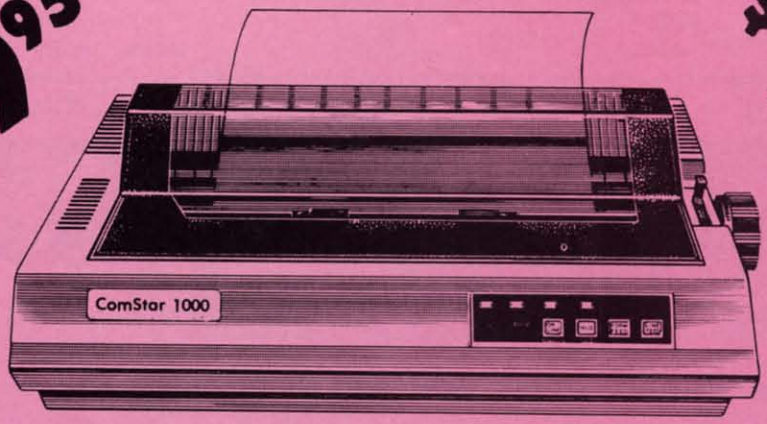
Near Letter Quality *Near Letter Quality*

Best Value in the USA

• 100 CPS draft/20CPS near-letter quality • Dot Addressable Graphics • Adjustable Tractor and Friction Feed • Automatic Paper Loading • Right and Left Margin settings • Pica, Elite, Condensed, Italics • Superscript • Subscript • Underline, Bold print, Double Strike • Superb NEAR LETTER QUALITY

$179.95

$179.95



Easy to Use

Fantastic Graphics

2 Year Limited Warranty

The Comstar 1000 is one of the best values in the United States today. Print your letters, documents, programs, pictures, and more at a blazing 100 Characters Per Second or 20 cps in the Near Letter quality mode. (Looks just like it came from a typewriter.) Plus, choose your printing mode (NLQ, Draft) from your wordprocessor or by simply pushing the controls on the front panel. Fantastic Quality at a Fantastic Price.List $349.00 SALE$179.95.

Print Method
Serial impact dot matrix (9 pin)

Print Speed
Draft- 100 CPS NLQ- 20 CPS

Character Sets
96 ASCII Characters, Marker, Symbols (includes italic font)

Ribbon (Life exp.)
Black: cassette (2.5 million characters)

Dimensions
15.4 (W) x 10.9 (D) x 4.7 (H) inch

Weight
Approx. 10 lbs

Character Spacing
Fixed

Line Spacing
1/6, 1/8, 7/72, and 1/216 inch

Paper Feed
Adjustable tractor and friction feed

Paper feeding Direction
Bi-directional

Copies
2 plus original

Supply is Limited so Buy Today

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6½% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Prices & Availability subject to change without notice.
VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 269

BREAK THROUGH

The TIME BARRIER

Ahoy! uses only
maxell
premium quality disks.

with **Ahoy!**
PROGRAM DISKS AND CASSETTES

BEST OF '85
- Music Tutor
- Midprint
- Fontasia
- Telelink
- Futurewar!
- Salmon Run
- Auto Line Numbers
- Home Budget
- Tile Time
- Screen Dump
- Ahoy! Dock
- File Lock
- Lightning Loader
- Directory Manipulator
- Gameloader
- Jewel Quest

BEST OF '84
- Multi Draw 64
- Address Book
- VIC Space Lanes
- 64 Space Lanes
- Name that Star
- Lunar Lander
- Air Assault
- Math Master
- Biorhythm
- Checklist
- The Alpiner
- Post Time
- Salvage Diver
- DOS
- Micro Minder '85
- VIC 40
- Emerald Elephant
- Lawn Job
- Construction Co.
- Space Patrol
- Multi Draw Boot

BEST UTILITIES
- Booter
- BASIC Trace
- Two Column Directory
- Commodore Font Editor
- Screen Magic
- Screen Bit Dump
- Printat
- Auto Append
- Formatter
- Windows
- Start & End File Address
- Fastnew
- Maestro
- 1541 Cleaning Utility
- DOS
- Base Conversions
- Micro Minder '85
- Lucky Lottery

BEST GAMES
- Name That Star
- Post Time
- Alpiner
- Salvage Diver
- Lawn Job
- Emerald Elephant
- Construction Co.
- Space Patrol
- Tile Time
- Duck Shoot
- Script Analysis
- Wizard of Im
- Disintegrator
- Gators N Snakes
- Dragon Type
- Gravinauts
- Mystery at Mycroft Mews

Why waste time typing? All the programs in this issue of *Ahoy!* are available on disk or cassette! Use the coupon below to order a single disk or tape or a subscription... as well as the special disks described at left.

SINGLE ISSUE AND SUBSCRIPTION PRICES

(Postage and Handling Included)

	U.S.A.	CANADA	ELSEWHERE
❑ December '86 *Ahoy!* Disk ❑ December '86 *Ahoy!* Cassette	$ 8.95	$ 12.95	$ 13.95
❑ 12-Month Disk Subscription ❑ 12-Month Cassette Subscription	$ 79.95	$ 99.95	$124.95
❑ 24-Month Disk Subscription ❑ 24-Month Cassette Subscription	$149.95	$179.95	–

$10.95 EACH (disk only): ❑ Best of '84 ❑ Best of '85 ❑ Best Utilities ❑ Best Games

BACK ISSUES $8.95

In Canada add $2.00 per issue; outside US and Canada add $4.00 per issue.

❑ _____	$8.95	❑ _____	$8.95
❑ _____	$8.95	❑ _____	$8.95
❑ _____	$8.95	❑ _____	$8.95
❑ _____	$8.95	❑ _____	$8.95

SPECIFY
❑ *DISK* ❑ *CASSETTE*

Make check or money order payable to **Double L Software.** N.Y. State Residents must add 8¼% sales tax.

Send coupon or facsimile to:
***Ahoy!* Program Disk/Cassette**
Ion International Inc.
45 West 34th Street, Suite 407
New York, NY 10001

NAME_____

ADDRESS_____

CITY_____ **STATE**_____ **ZIP**_____

LAZY SOURCE CODE

Mnemonic Shorthand for Phlegmatic Typists

By Michael Bennett

Those of you who enjoyed my *LazyBASIC* program (*Ahoy!*, June '86) will find a familiar format in this program which does for assembly programming what *LazyBASIC* did for BASIC programming.

The onscreen menu has the same A-Q rows topped by the shiftless function keys (f1-f7) that must be hit prior to these letters in order to get a word to print out. But instead of BASIC keywords, the 56 mnemonic assembly instructions (such as "LDA" or "JSR") are listed alphabetically and can be accessed by two strokes instead of three. This program can be used alone or with a symbolic assembler program such as PAL or LADS, which allows you to type in your assembly as if it were BASIC.

If you think that a two for three savings of keystrokes is not impressive enough, how about two for thirteen? That is the maximum length of a labelname that you can define on the menu and then print out with two strokes. So instead of typing out "Spritepointer" every time you enter "JSR Spritepointer", you can define one of the nine labelnames on the menu to be "Spritepointer" and have it print out with two strokes. Defining the labelnames is a very simple process.

When you first load the program, the area of the menu where your labelname definitions will go is filled with my mailing address for feedback on this or *LazyBASIC*. If you were to hit f7 followed by the letter H at this point, the words "sendfeedback" would print out, with the space removed from between the words. Spaces are always removed, because that is how I achieved variable length labelnames on an easily altered menu.

Let's assume that you want to use that labelname "Spritepointer". First hit the RETURN key so that the whole menu is visible on the screen. This is an important step because the program recopies everything on the screen where the menu is supposed to be into the memory area where the menu data is stored. If only part of the menu is visible when you do this, only part of the menu will appear every time you hit the RETURN key from that point on. So hit the RETURN key as the first step in defining a labelname.

As your second step, cursor up to one of the labelname lines which are in column f7 in the rows H through P. Type in the labelname "Spritepointer" starting with the first space after the dash that follows the mnemonic instruction in the next column over. This space is the first of thirteen spaces that are available for your labelname. "Spritepointer" happens to have the maximum number of letters that you are allowed. After typing it in, cursor down below the menu and hit the English pound key. It causes everything on the menu area of the screen to be recopied into the menu data area, and from now on when you hit the RETURN key you will see "Spritepointer" on the menu. In addition, whenever you hit f7 followed by the letter corresponding with "Spritepointer" 's row, "Spritepointer" will print out.

Lazy Source Code has most of the other features of *LazyBASIC*, including automatic line numbers which are turned on by answering the question that appears when f6 is accessed followed by answering the question that appears when f8 is accessed and then hitting RETURN twice. For those of you who missed out on *LazyBASIC*, those questions are "JUMP between lines (1-255)", and "What's your line#". You turn off auto line#'s by hitting RETURN on a line which only has a line number on it.

F2 and f4 still toggle the screen and border colors as they did in *LazyBASIC*, and hitting the wrong key still gets you a message "You hit the wrong key—try again!!" Turning off the program is still accomplished by the morbid "Die!!" choice of f7 followed by Q.

My aversion to the shift key lingers in this program with the inclusion of "#" and "$" under f7. These frequently used symbols in assembly programming seem slightly easier in two strokes than they do with a held-down shift key and a stretched finger.

I hope this program helps make the world of machine language as easy and as fun for you as BASIC. □

SEE PROGRAM LISTING ON PAGE 136

SEASONS GREETINGS

ABACUS
Ada Training Course	$29.95
Basic 128	42.95
Basic 64	29.95
Cadpak 128	45.95
Cadpak 64	29.95
Cadpak 64 w/Light Pen	69.95
Chartpak 128	29.95
Chartpak 64	29.95
Cobol	29.95
Personal Portfolio Manager	29.95
Powerplan	29.95
Super C Compiler 64/128	42.95
Super Pascal	45.95
Techinical Analysis System	45.95

BRODERBUND
Where in the World is Carmen Sandiago	29.95
P/S Library 1-2-3	15.95
Print Shop	26.95
Print Shop Companion	21.95

MICRO LEAGUE
General Managers Disk	$24.95
Micro League Baseball	24.95
Team Disks	16.95

RIBBON
Axiom Elite 5 LCD	$ 5.95
Brother 15, 25, 35	5.95
Commodore 1525	6.95
Commodore 1526	6.75
Epson 185	6.95
Epson 85	4.95
Epson mx, fx, rx 100	6.95
Epson mx, fx, rx, 1x 80	4.75
Gorilla Banana	6.95
Juki 6000	4.95
Juki 6100	3.95
Legend 880, 1080, 808	6.75
MPS 801	5.75
MPS 803	6.95
OKI 182, 192	9.75
OKi 82, 92, 83, 93	2.25
SG 10-15	2.25

(Min. 6 per order)

INFOCOM
Ballyhoo	$24.95
Enchanter	21.95
Hitchhicker's Guide	21.95
Invisiclues (all)	6.00
Spellbreaker	26.95
Zork 1-2-3	25.95
Trinity 128	24.95

PRECISION
Superbase 128	$56.95
Superbase 64	46.95
Superscript 128	47.95
Superscript 64	46.95

TIMEWORKS
Data Manager 128	$42.95
Data Manager II	29.95
Partner 128	42.95
Swift Calc 128 w/Sideways	42.95
Swift Calc 64 w/Sideways	29.95
Word Writer 128 w/Speller	42.95
Word Writer 64 w/Speller	29.95

NEW RELEASES
Partner 1 2 8	43.95
Ace	14.95
Graphic Expander	21.95
Murder on the Mississippi	21.95
Where in the World is Carmen Sandiago	21.95
World Games	24.95
Top Gunner	16.95
Mercenary	17.95
Hacker II	21.95
Battlefront	26.95
The Infinity Machine	15.95
Ace of Aces	17.95
Ninja	7.95

DIGITAL SOLUTIONS
Pocket Filer 128	$31.95
Pocket Filer 64	24.95
Pocket Planner 128	31.95
Pocket Planner 64	24.95
Pocket Writer 128	31.95
Pocket Writer 64	24.95

MICROPROSE
Conflict in Vietnam	$24.95
Crusade in Europe	24.95
F-15 Strike Eagle	21.95
Silent Service	21.95

SPRINGBOARD
Clip Art I	$20.95
Clip Art II	24.95
Newsroom	30.95
Graphic Expander	21.95

ACCESS
Leader Board	$24.95
Leader Board Tournament Disk	14.95
Mach 128	30.95
Mach 5	21.95

ELA
Bard's Tale	$24.95
Lords of Conquest	24.95
Movie Maker	24.95
Chessmaster 2000	26.95
Mind Mirror	21.95
Ultimate Wizard	19.95

HOT SELLERS
Geos	$37.95
Commando	21.95
Koala Pad	42.95
Koala Printer Utility	16.95
Print Shop Companion	21.95
Int'l Hockey	16.95
Kung Fu II	19.95
Gato	19.95
Fontmaster II	31.95
Ultima III & IV ea	32.95
Super Bowl Sunday	19.95
Super Bowl 1985 Teams	12.95

FIREBIRD
Elite	$18.95
Colossus Chess IV	21.95
Frankie Goes To Hollywood	21.95
Music Systems	24.95
Advance Music Systems	49.95

ACTIVISION
Alter Ego	$29.95
Cross Country Road Race	19.95
Mindshadow	19.95
Murder on the Mississippi	21.95

ACCESSORIES
WICO Boss Joystick	$11.50
WICO Bat Handle Joystick	15.50
WICO Three Way Joystick	18.95
Kraft Joystick	6.95
Xetek Graphic Interface	54.95
Xetek Graphic Jr	34.95
RS 232 Interface	34.95
1200 Baud Modem	99.95
Master Modem	34.95
Flexidraw Light Pen	CALL
Disk Notcher	5.95
AICP Disks (10)	7.95
Drive Cleaner	7.95
Printer Head CLeaner	7.95
Disk Storage Case (50)	7.95

SPECIAL SAVINGS
FREE
PRINTER HEAD CLEANER KIT
with every
RIBBON ORDER

BATTERIES INCLUDED
Consultant 64/128	$39.95
Paper Clip 64/128	39.95
Paper Clip w/spell 64/128	49.95

ACCOLADE
Dambusters	$18.95
Fight Night	18.95
Hard Ball	18.95
Law of The West	18.95
PSI 5 Trading Comp	18.95

SUBLOGIC
Flight Simulator II	$34.95
Football	27.95
Jet	27.95
Scenery Disks 1-6	14.95 ea
Pure Stat Baseball	34.95

KOALA
Koala Pads 64	$42.95
Koala Printer Utility	16.95
Programers Tool Kit	24.95

MINDSCAPE
Bop and Wrestle	$18.95
Infiltrator	18.95
Halley Project	24.95

EPYX
Fast Load	23.95
Multiplan 64/128	40.95
Programmers Tool Kit	27.95
Vorpal	21.95
World Karate Championship	18.95
Worlds Greatest Baseball	23.95
Super Cycle	24.95
World Games	24.95
Worlds Greatest Football	24.95

COMMODORE HARDWARE
C-128	CALL
C-1571	CALL
C-1902A	CALL
C-1670 Modem 1200	CALL
C-1350 Mouse	CALL
64 C	CALL
1541 C	CALL

2 Games In One
Gerry The Germ/ Microcosm	13.95
Willow Pattern Chimera	13.95
UnderWurlde Sabre Wulf	13.95
Booty/Cylu	13.95
Runestone/The Helm	13.95
Chicken Chase/Rasputin	13.95
Battle of Britian Battle for Midway	13.95
ARC of Yesod Nodes of Yesod	13.95

AMIGA SOFTWARE
Borrowed Time	$27.95
Hacker	27.95
Mindshadow	27.95
Music Studio	37.95
Aegis Animator w/Images	89.95
Aegis Draw	124.95
Archon	29.95
Deluxe Paint	72.95
Skyfox	29.95
Rogue	24.95
Temple of Apshai Trilogy	24.95
Master Type	24.95
Halley Project	29.95
Racter	32.95

SSI
Battle Of Antietam	$29.95
Carrier Force	36.95
Kampfgruppe	36.95
Phantasie II	24.95
Rings of Zelphin	24.95
Wings of War	24.95
Wizards Crown	24.95

Solid State Software
VizaStar 128	$79.95
VizaWrite 128	59.95

GREAT GIFT IDEA
Buy 2 Packs of AICP Disks
Receive A
FREE
Disk Storage Case
store 50 disks

CALL US 1-800-634-AICP TOLL FREE

EXTENDED HOLIDAY ORDERING HOURS
10AM TO 8PM E.S.T ORDERS ONLY

Customer Service
718-351-1864
10AM to 6PM
E.S.T.

AMERICAN INTERNATIONAL COMPUTER PRODUCTS
P.O. Box 1758
Staten Island, N.Y. 10314
Reader Service No. 251

ORDER INFO

In stock items shipped within 24 hours for prepaid orders (cert. check, visa, m/c.) C.O.D. orders accepted, please add $4.00, must be cash or cert. check ONLY. NY State Residents add 8¼% sales tax. Advertised prices reflect a 4% discount for cash, credit card add 4%. All orders add $3.50 shipping and handling within the Continental U.S. International orders add $15.00 + shipping & handling. Personal checks allow 2 wks. for clearance. Availability and Pricing subject to change without notice. Orders over $150.00 receive free shipping in Continential U.S. School inquiries invited. Make all checks payable to American International Computer Products. DEFECTIVE EXCHANGES ONLY, REQUIRE RETURN AUTHORIZATION NUMBER.

ART GALLERY

Art Gallery Disk Sale

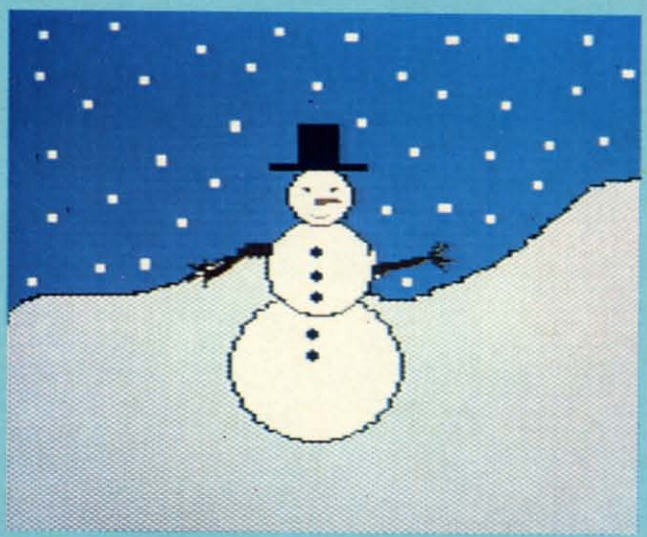
The images on these pages are now available on a monthly disk. Multicolor images are supplied in *Koala* format, while high-resolution images are in *DOODLE!* format. Included are a slide show for easy viewing, along with a bit map dump for your 1525 printer or properly interfaced equivalent. Annual (12 month) subscriptions are $89. Individual disks are $12 per month. A sample *Art Gallery* disk with slide show and printer dumps is $10; or send a stamped and self-addressed envelope (business size) for a listing of available *Art Gallery* collection disks. Prices shown are for US and Canada. All others add $3 per disk. New York State residents please add appropriate sales taxes. Disks may be ordered from Morton Kevelson, P.O. Box 260, Homecrest Station, Brooklyn, NY 11229.

Contribute to Ahoy!'s Art Gallery

The *Ahoy! Art Gallery* offers the opportunity for fame and fortune to any and all aspiring Commodore artists. Simply send Morton (see address above) your work on disk indicating the drawing package or file format of the images. All graphics produced on the C-64/C-128 and Plus/4 computers are eligible. In exchange your work will receive the opportunity for display in these pages. All published works will receive royalties based on the monthly *Art Gallery* disk sales. In addition, both published and unpublished images may be included on the various *Art Gallery* collection disks.

Note that the *Art Gallery* is not a contest. Published pictures are selected in an arbitrary and capricious fashion by the *Ahoy!* Art Director based solely on the artistic merit of the individual images.

By gosh, by golly—though it's still September as we send this issue to press, this holiday installment of the *Art Gallery* has us looking up the chimney. At top left is *Barn* by Marcy Willbrandt (Battle Creek, MI), drawn with Koala. Only the stockings hung with care are missing from *Fireplace* at top right by Daryl Maksymec (Regina, Massachusetts), also on Koala. Fully accoutred, however, is the fireplace and tree in Marcy Willbrandt's *Christmas* in the middle row. Beside it is *Santa* by Michael Mikottis (Berwyn, IL), done on Koala. Parting with tradition, though, a mouse is stirring in *Christmas Tree* by Sandy Steele (Rockford, IL) at bottom left, done on *DOODLE!* Finishing up are three Koala images by Pamela and Richard Winters (Shreveport, LA): *Sleigh*, *Happy New Year*, and *Frosty*.



BUILD A FIVE VOLT PERIPHERAL POWER SUPPLY

By Morton Kevelson

With the exception of the joystick, every Commodore peripheral requires a source of electrical power. Major peripherals, such as disk drives and printers, will invariably contain their own power supplies. Minor peripherals, in particular printer interfaces, form a symbiotic relationship with the host computer. That is, the power is drawn from the computer's power supply via one of the peripheral ports.

With regard to most printer interfaces, the five volt DC supply is obtained from the computer's cassette port. In this case a single wire from the interface is terminated in a six pin edge card connector which mates with the cassette port. A small printed circuit board is attached to this connector to extend the cassette port so that the tape deck may still be used.

While this hookup does work, we have always viewed it as a workable solution to an intractable problem. The extension board projects beyond the computer's case, where it is subject to contact with stray objects. If you double up peripherals, such as *Serial Box* (reviewed last month) used with a printer interface, we end up with an awkward pairing of two of these extension boards. We have also found that excessive movement of the connecting wire invariably causes it to break. Of course, users of the SX-64 and Plus/4 computers lack the appropriate cassette port entirely.

Some interfaces, Micro R & D's MW-350 in particular, have opted for a joystick port connection to obtain their five volt stimulation. The problem with this arrangement is the tenuous hold the joystick connector has on the port. The real estate to the right of our computer is invariably occupied by pads, pencils, disks, manuals, and other paraphernalia. These are always being moved about, especially during those intervals of printer activity. Somehow the joystick port connector always attracts a glancing blow with disastrous consequences. Aside from all this, our Plus/4, with its peculiar joystick and cassette connectors, is still out of luck.

On top of all this we have encountered an overriding consideration beside which all of the above pales by comparison. We have found that the Commodore 64 power supplies are a finicky lot. In spite of claims to the contrary, the C-64 power pack has little tolerance for external electrical loads. This supply provides two voltages for the computer's use. The critical one is the five volt source. Although the computer requires less than half of the supply's rated output, we have found that the slightest extra current drain causes the built-in voltage regulator to shut down. Symptoms of this activity are a breaking up of the video display, along with a total lack of

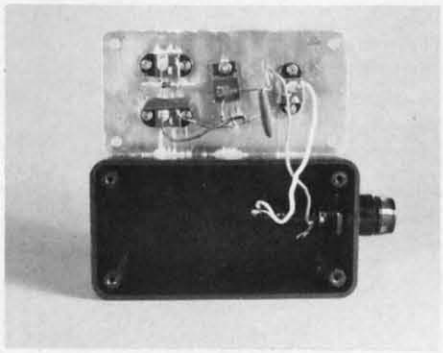


FIGURE 1

The completed peripheral power supply. The center screw mounts the 7805 voltage regulator IC.

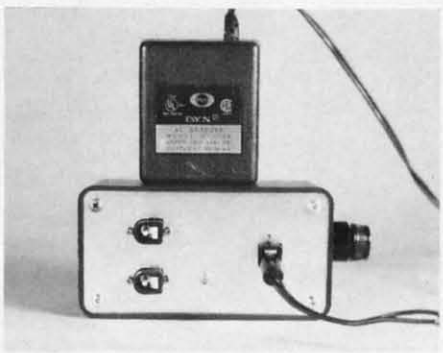


FIGURE 2

View of the inside of the peripheral power supply. The fuse holder is mounted to the side of the case.

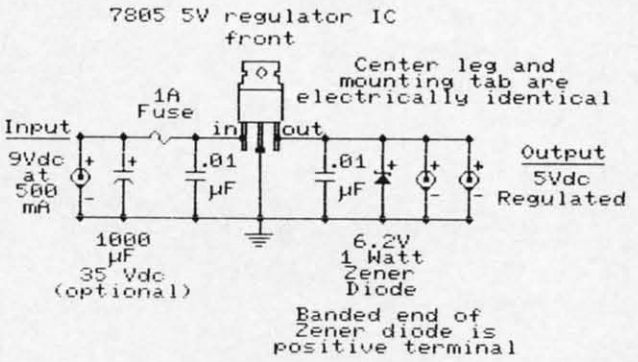


FIGURE 3
Schematic of the five volt DC peripheral power supply.

FOR COMMODORE 128 AND C-64 OWNERS ONLY:

BLITZ!™

NOW SHOWING AT SKYLES ELECTRIC WORKS

If your programs are slow, BLITZ!™ them

It's Spectacular
Bob has discovered a spectacular compiler for the **Commodore 64** and the **Commodore 128** that he calls ***BLITZ!***

It's Fast
BLITZ! is faster than *PET SPEED*, faster than any other Commodore compiler that has come down the pike. Your ***BLITZ!***-compiled program will run from *five to 20 times* faster than it did before you blitzed it.

It Translates
This disk program translates your slow BASIC programs into a much, much faster code. The ***BLITZ!*** compiler significantly improves the performance of your BASIC routines. It reads the entire program, reduces that program's size by deciding which operations need to only run once, interprets the operations, and then re-writes the program into its own special P-code.

C-64 or C-128

It's a Treasure
Using this compiler, you can now create a string of BASIC programs such that one automatically loads the next. You can also pass information from one program to another. You will find the ***BLITZ!*** compiler a treasured tool in your programming kit.

It's Popular
Reviewers and users have heaped the following praise on the ***BLITZ!***: "The ***BLITZ*** basic compiler is a dream come true for BASIC programmers." Compute!'s Gazette "worth its weight in gold." TPUG "Wonderful" "Fantastic" "I can't imagine programming without it" User comments. It is even more popular with professional programmers. Many commercial programs now say "***BLITZ!***" when they are listed. The ultimate accolade for ***BLITZ!***.

	WINTER SALE
***BLITZ!* C-128, Disk**	**69.95***
***BLITZ!* C-64, Disk**	**49.95***

QUICKSILVER 128™ now available for Commodore 128

Is the fastest IEEE-488 Interface available for the **Commodore 128.** How fast? As fast as the external IEEE-488 device will allow. For example, a program or file will load over four times faster from a Commodore 4040 or 8050 disk drive as compared to a 1541 disk drive. ***QUICKSILVER 128*** loads 5 times faster than a serial MSD drive. When used with printers speed is only limited by the printer.

QUICKSILVER 128 is transparent. ***QUICKSILVER 128*** does not occupy any program memory. Almost all programs are completely compatible with ***QUICKSILVER 128.*** A word of caution, some IEEE disk drives have different formats, and 1541 formatted programs are not going to load. Some programs protect the disk by various operations in the 1541 disk memory. These programs will not load on non-1541 disk drives. However, for most uses ***QUICKSILVER 128*** is very quick and transparent.

C-64 or C-128

QUICKSILVER 128 will interconnect your **Commodore 128** to Commodore SFD 1001, 2031, 2040, 3040, 4040, 8050, 8250, 9060, and 9090 Disk Drives. ***QUICKSILVER 128*** will connect your **Commodore 128** to Commodore 2022, 2023, 4022, and 4023 Printers.

QUICKSILVER 128 is easy. Easy to plug into the cartridge port. Easy to plug additional cartridges into the built-in horizontal, cartridge port extender. Easy to locate your IEEE device within 6 feet of ***QUICKSILVER 128*** using the built-in 6 foot IEEE cable. Easy to select which device numbers you want ***QUICKSILVER 128*** to interface to the IEEE bus. Easy to use with almost all programs. Easy to buy, at an amazingly low price, from Skyles Electrics Works or your local dealer.

	WINTER SALE
***QUICKSILVER 128*, C-128 Cartridge**	**$119.95**
***IEEE Flash! 64*, C-64 Cartridge**	**99.95**

So many cartridges, so little space . . .

Skyles Electric Works to the rescue, with our **2+1**™ and **4+1**™ expansion boards for the C-64. Briefly the **2+1** gives you 2 vertical and 1 horizontal fully switchable cartridge ports. The **4+1** continues the tradition with 4 vertical and 1 horizontal cartridge ports. See the inside front cover for more details.

	SUMMER SALE
Skyles 2+1 Expansion Bd. C-64	**$49.95** / **39.95***
Skyles 4+1 Expansion Bd. C-64	**$89.95** / **69.95***

*There is an additional $4.50 U.S. and Canada, $15.00 Europe and Asia, shipping charge per order. California residents add sales tax.

Skyles Catalogue Page 3

From Skyles Electric Works, the oldest and largest professionals in the business.

This is just a few of the 200+ bargains from the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full splendor, and the other pages overflowing with over 200 bargains in peripherals, software, and books that will make your Commodore 128 or C-64 computer even nicer to live with.

So, if we missed sending you your very own copy within the last few weeks, call us at 1-800/227-9998, unless you live in California, in which case call 1-415/965-1735.

Skyles Electric Works
231E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

Reader Service No. 239

"MAXI-DISK"...
A MUST FOR YOUR LIBRARY

"Programs by Users, for Users"

THE NEW MONTHLY PUBLICATION ON DISK
FOR YOUR C-64 AND C-128

10 powerful and exciting programs for your Commodore computers not seen in any other publication at a price hardly worth mentioning. Disk comes with a booklet with complete instructions for all programs.

MUSIC 64
A complete and powerful music making program for your Commodore 64.

SPRITE RITE
Single or Multicolored and creates data statements for you. Make a whole library with this one.

DATABASE MANAGER
Try this easy to use DB Mgr. Handles your data easily and has all the tools.

SUPERBOWL HELPER
Keep getting nipped by those spreads...This will help even your odds....

ALSO ON THIS DISK: Super 5 Card Poker, Program Packer, Journey, Cross Reference, Disk Jacket, Space Pirates.

MICRO-TEK
P.O. BOX 1861
STATEN ISLAND, NY 10314

Please *RUSH* me the following order

(postage and handling included):	**USA**	**FOREIGN**	**Back Issue**
☐ Volume 2 (Dec. '86) issue of the "MAXI-DISK"	$ 12.95	$ 15.95	☐ Vol. 1
☐ 1 Year Subscription of the "MAXI-DISK"	$119.95	$139.95	

Make checks payable to Micro-Tek Software. New York residents add 8.25% sales tax.

Name_____ Address_____

City_____ State_____ Zip_____

ARE YOUR PROGRAMS GOOD ENOUGH?

MICRO-TEK Software is looking for quality games, utilities or business programs written by you, the computer enthusiast, to share with your fellow hackers. At the same time, you can make extra cash and get the satisfaction and recognition you deserve.

MICRO-TEK publishes the **"MAXI-DISK"** **"MAXI-DISK"** is a C-64 program disk with 10 quality programs, not published by any other publication. A new and exciting program disk is produced each and every month. Our premier disk will be available November 1986.

If you are interested in submitting your programs, please observe the following rules:

1. **All programs must be written and owned by you and submitted on C-64 formatted disk.**
2. **Typed, double-spaced documentation must accompany the program.**
3. **Rejected submissions will only be returned with S.A.S.E.**
4. **Prices subject to quality and length of program.**
5. **Upon acceptance, you will be sent a contract to which you must sign and return.**

MICRO-TEK Software pays generous royalties to programmers based on the number of disks sold for as long as the disk is for sale.

response from the keyboard. If the external load is disconnected and the power supply is allowed to cool down, normal operation resumes.

The solution to all of the above is to simply provide our peripherals with their own source of power. It turned out to be a simple construction project based on parts available at our local Radio Shack. Before we begin we must present the following:

WARNING!

TO PROPERLY COMPLETE THIS PROJECT YOU WILL REQUIRE BASIC SKILLS IN SOLDERING ELECTRONIC COMPONENTS. ACCESS TO A VOLT/OHMETER AND THE KNOWLEDGE TO USE IT WILL BE HELPFUL FOR THE COMPLETION OF THIS PROJECT. MINOR MODIFICATION OF *SERIAL BOX* AND THE PRINTER INTERFACE WILL BE REQUIRED. THIS WILL VIOLATE THE MANUFACTURER'S WARRANTY. PROCEED AT YOUR OWN RISK.

We hate to seem melodramatic, but such is life.

The project is based on a readily available battery eliminator-style power supply. For our purposes we chose a nine volt unit rated at 500 ma. A milliamp, abbreviated ma, is 1/1000 of an ampere, which is the basic unit of electrical current. The interface requires a regulated source of five volts DC. This is obtained by using a 7805 integrated circuit voltage regulator.

The electrical connections are shown in the accompanying schematic. The remaining components perform various auxiliary functions. The .01 micro Farad capacitors, C1 and C2, decouple the 7805 from the inductance of the connecting leads. The 1000 micro Farad electrolytic capacitor was not used in our prototype. It is required only if the battery eliminator lacks its own filter capacitor. Virtually all general purpose units are so equipped.

The 6.2 volt zener diode serves as a surge protector. It should also cause the fuse to blow in the unlikely event of the complete failure of the 7805. The 7805 voltage regulator has built-in protection against external short circuits and thermal overload of the chip.

The entire project is built into a small utility box. The metal cover of the box also serves as a heat sink for the 7805. A small dab of thermally conductive silicon compound should be applied to the 7805's mounting tab against the box cover. We chose the DC coaxial connectors as they do not short circuit when the plugs are inserted or removed.

Your printer interface will have to be slightly modified to access the ground connection. The positive five volt hookup should be made to the existing wire which goes to the cassette port connector. The ground connection can be traced off pin 2 on the six pin serial bus DIN plug. The pin layout for this plug is shown on page 142 of the *Commodore 64 User's Guide* or page 350 of the *Commodore 128 System Guide.* Just remember that the view shown in the manual is looking from the back at the port in the computer. The actual peripheral connector plug will be a mirror image. Once you have found the correct wire, just solder on the ground lead to a convenient point in the interface.

Interface Current Requirements

Interface	Current
Serial Box	95
Xetec Supergraphix	70
Xetec Graphix Jr.	65
Xetec GPI (obsolete)	250
Micro R & D MW-350 with 10K	190
Cardco ?/A	250
Cardco G-Wiz	90
Cardco Super G	180
Tymac Connection	250

All values are approximate at rated five volts DC. Specific values may vary from sample to sample.

Power Supply Parts List

Item	Radio Shack Stock No.	Description
Battery Eliminator	273-1651	120 Vac to 9Vdc, 500 ma
C1, C2	272-131	.01 uF Capacitor
C3	272-1019	1000 uF, 35 volt, electrolytic capacitor (optional; see text)
J1, J2, J3	274-1565	Coaxial DC jack
P1, P2	274-1567	Coaxial DC plug
(See Description)	270-231	4"×2⅛"×1⅝" utility box
VR	276-1770	7805, 5 volt regulator IC
D1	276-561	1N4735, 6.2 volt, 1 watt zener diode
(See Description)	276-1732	Heat sink grease
(See Description)	270-362	Fuse Holder (5 × 20 mm)
F1	270-1250	1 amp fuse (5 × 20 mm)

Power requirements for interfaces vary. The table above shows the current draw, at five volts DC, for several interfaces we have looked at. The power supply presented here should be able to deliver five volts at 500 ma. This should be enough to drive any two of these peripherals. □

REBELS AND LORDS

For the C-64

By Terry Bryner

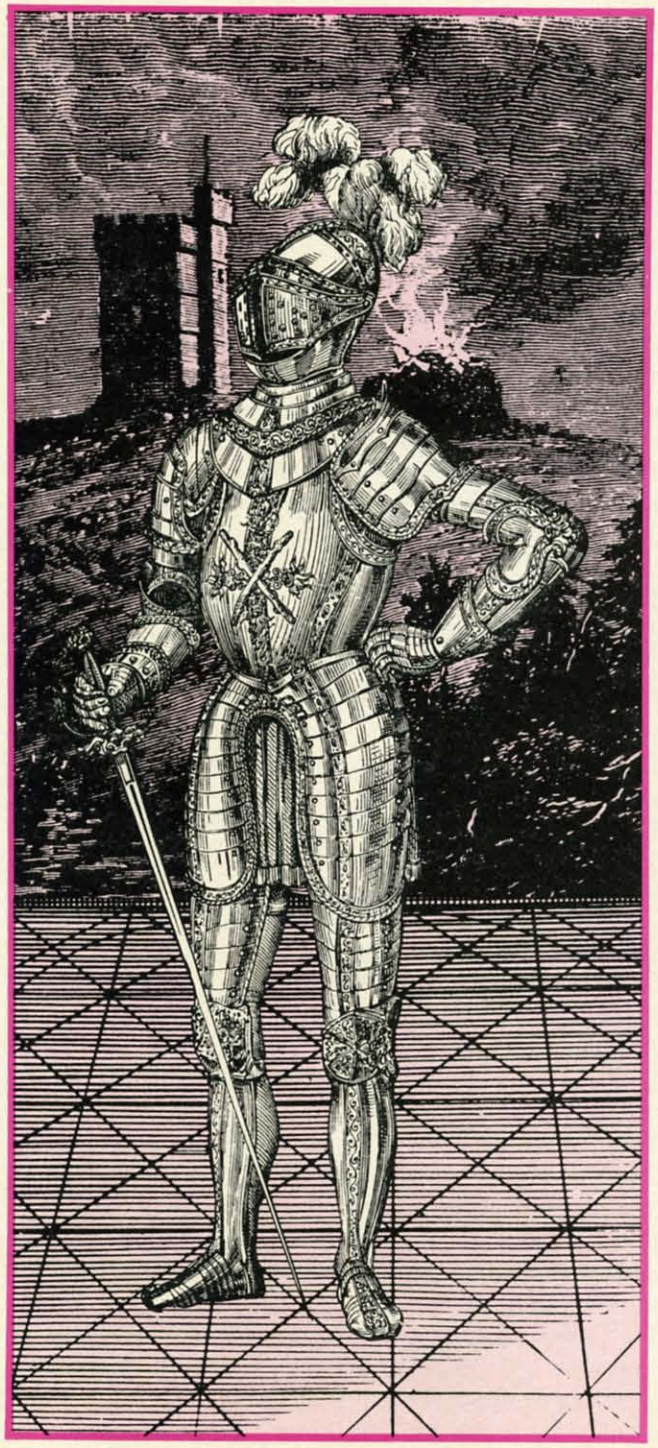
The peasants have rebelled, and the King has made you Lord of a castle, charged with subduing them! In this graphic wargame, you (and up to three other players) pit your armies against the wild rebels on the graphic map to win back the castles for the King. Three levels of difficulty are provided, and even the solitaire game can be challenging.

When you run the game, there will be a delay while the program initializes itself. Then you will be asked to specify the number of Lords (players), the number of castles, and the type of terrain. The program will then generate a map for this game. Most of it will be grassy fields, with castles placed randomly around. The red castle is the first Lord's, the yellow the second's, etc., while the rebel hordes own the black ones. Trees (which hide armies) and rocks (impassable) are also shown. If the first map is unacceptable, reject it and another will be generated.

The game is run in weeks: each Lord issues his armies movement orders, and then all armies move and attack. Follow the prompts, and a square window will appear on the map, and the contents will be displayed below the board. The window can be moved around with the joystick or cursor keys (interchangeable; use f7 for the fire button), to get more information. For your castles you will see the troops inside, and also the 'levy'– the number of additional troops you will receive here each week as long as you hold this castle. You may also see some armies listed below—an identification number and a troop strength (they already have orders to go elsewhere). Non-castle squares may also contain armies; the ones listed at the bottom with an identification number already have orders, while idle troops will be counted above. (Note that information is only available for your own forces; you will not even be told if enemies are hiding in the trees.)

To send unoccupied armies somewhere, press the fire button, use the joystick to set the size, and press the fire button/f7 again. A destination cross will appear: Move this with the joystick/cursor keys to the destination and press the fire button, and the army has its orders!

A word about the troops. They are typical peasants: loyal to their leader, furious fighters, but not overly skilled at marching. They only march five squares a week, and cannot be recalled. In order to move to the southeast,



Ahoy! STILL #1

Everyone respects *INFO* Magazine as the independent observer of the Commodore marketplace. So we were especially proud when *INFO's* September-October '85 product roundup awarded *Ahoy!* a **– rating: the highest given any Commodore magazine. The only fault publisher Benn Dunnington found with *Ahoy!* was that it "seems to fluctuate from issue to issue."**

We promised to stop—and apparently we have, because the August/September '86 *INFO* awarded us four stars—the highest rating of any Commodore monthly!

***INFO's* review of *Ahoy!* is printed here in its entirety, along with those of the three other major Commodore publications. One has retained its previous rating, and two have lost points—while *Ahoy!* moves ahead!**

hoy!
2.75 Ion International
★★★★
Monthly for C64/C128, and considering Amiga coverage. Ahoy! offers type-in programs in BASIC, so you can actually learn from them, articles that are informative and even challenging, and reviews that are somewhat cautious, but sometimes critical. Dale Rupert, Morton Kevelson, and others on the Ahoy! staff are real experts. The best of the 'big three,' with more 'hacker flavor' than the others. (—B.D.)

Compute!'s Gazette
$2.95 Compute Publications
★★★
Slick monthly publication with coverage of Commodore's 64, 128, and some on the Amiga. Emphasis is on type-in games and occasional utilities. Reviews are generally candy-coated but reasonably current. Goes with the flow, avoiding controversy and originality. Both the quality and the accuracy of the type-in programs has been way down lately.(—B.D.)

Commodore Microcomputers
$2.50 Commodore Publications
★★★+
Recently merged with 'Power Play,' this is Commodore's own magazine. Some good informative articles, type-in programs in BASIC (so you can learn something), candy-coated 'reviews' that read like press releases, and good photos. Don't expect to read any criticism of Commodore from this gang: their paychecks are still signed in West Chester, PA. Monthly. (—B.D.)

RUN
$2.95 CW Communications
★★★+
This slick monthly Commodore magazine is chock-full of ads, lots of color and photos, occasional decent articles. Reviews are often out of date and, worse, not to be trusted. Though they've toned it down somewhat lately, they are still treating the Plus/4 seriously. Best features are the MAGIC column, which is best obtained in its indexed, all-in-one-place form in the annual *RUN* Special, and Strasma's C= Clinic. (—B.D.)

Back Issues

SPECIAL ANNOUNCEMENT!

A small cache of *Ahoy! #2* (Feb. '84) has been uncovered, and is available at $10 per copy while limited supplies last.

SORRY... ISSUES #1, 2, AND 4 SOLD OUT!

ISSUE #3—MAR. '84 $4.00
Anatomy of the 64! Printer interfacing for VIC & 64! Educational software series begins! And ready to enter: Address Book! Space Lanes! Random Files on the 64! Dynamic Power!

ISSUE #5—MAY '84 $4.00
Future of Commodore! Inside BASIC storage! Memory management on the VIC & 64! Guide to spreadsheets! And ready to enter: Math Master! Air Assault! Biorhythms! VIC Calculator!

ISSUE #6—JUNE '84 $4.00
Game programming column begins! Program generators! Rupert on inputting! Memory management continues! And ready to enter: Post Time for the 64 & VIC! Alpiner! Sound Concept!

ISSUE #7—JULY '84 $4.00
MSD dual disk drive! Database buyer's guide! Training your cursor! Screen displays! File Sleuth! Users Groups! And ready to enter: Renumbering! Checklist! Math Defender! Brisk!

ISSUE #8—AUG. '84 $4.00
Choosing a word processor! Computational wizardry! Creating your own word games! Sound on the 64! And ready to enter: Micro-Minder! Directory Assistance! The Terrible Twins!

ISSUE #9—SEPT. '84 $4.00
Program your own text adventure! Build a C-64 cassette interface! Video RAM! And ready to enter: Salvage Diver! DOS! Sound Explorer! The Castle of Darkness! Base Conversions!

ISSUE #10—OCT. '84 $4.00
C-64 graphics programs! Bit-mapped graphics! Joystick programming! And ready to enter: VIC 40 Column Operating System! BAM Read & Print! Emerald Elephant! Lawn Job!

ISSUE #11—NOV. '84 $4.00
Music programs & keyboards for the 64! Graphics feature continues! And ready to enter: PTE word processor! Block Editor! Alternate Character Set for the 64! The Tunnel of Tomachon!

ISSUE #12—DEC. '84 $4.00
Buyer's guide to printers! 1525 printer tutorial! Custom characters! User Guide to KMMM Pascal! Diving into BASIC! And ready to enter: Construction Co.! Space Patrol! Cross Ref!

ISSUE #13—JAN. '85 $4.00
VIC/64 OS exposed! Sprites! 1541 device # disconnect switch! Ghostbusters! And ready to enter: Ultra Mail! Music Tutor! Alice in Adventureland! Midprint! To the Top! Tape/Disk Transfer!

ISSUE #14—FEB. '85 $4.00
Printer interfacing! Multicolor sprites! Modems! Bulletin boards! Theory of game design! And ready to enter: Futurewar! Fontasia! VIC Eraser! Insurance Agent! Flankspeed! Telelink 64!

ISSUE #15—MAR. '85 $4.00
Making multiscreen gameboards! Inside the Plus/4! Commodore DOS! And ready to enter: Old Routine! Programmable Functions! Automatic Line Nos.! Home Budget! Salmon Run!

ISSUE #16—APR. '85 $4.00
Assembly language column begins! Programming the joystick! 1541 disk drive alternatives! And ready to enter: Hop Around! Faster 64! Booter! Elecheck! BASIC Trace! Space Hunt!

ISSUE #17—MAY '85 $4.00
Disk drive enhancements! Install a reset switch! Assembler escapades! And ready to enter: Super Duper! Two-Column Directory! DSKDU! Raid! DOS Plus! Font Editor! Tile Time!

ISSUE #18—JUNE '85 $4.00
Music & graphics entry systems! How modems work! Inside the 6510! And ready to enter: Quad-print! Mapping 4.4! Towers of Hanoi! Speedy! Duck Shoot! The 6510 Simulator!

ISSUE #19—JULY '85 $4.00
PROM programming! 3-part harmonies on VIC/64! Speeding pixels! And ready to enter: Auto-Append! Script Analysis! Wizard of Im! Lucky Lottery! Brainframe! Etch! Printat!

ISSUE #20—AUG. '85 $4.00
Inside the 128! Read-world simulations! Sound effects! And ready to enter: Windows! Formatter! Sound-a-Rama! Screen Dump! Selectachrome! Disintegrator! Fidgits! Gators N Snakes!

ISSUE #21—SEP. '85 $4.00
Inside the 1571 drive and 128 keyboard! Sprite programming! And ready to enter: Fastnew! Go-Lister! File Lock! Dragon Type! Superhero! Auto-Gen! Moxey's Porch! Fish Math!

ISSUE #22—OCT. '85 $4.00
Create cartoon characters! Infinitesimal intrigue! Inside copy protection! And ready to enter: Shotgun! Maestro! Solitaire! Mystery at Mycroft Mews! Gravinauts! 1541 Cleaning Utility! ShadeyDump!

ISSUE #23—NOV. '85 $4.00
Adventure gaming! ML sprite manipulation! BASIC for beginners! And ready to enter: Lightning Loader! Knight's Tour! Chopper Flight! Rhythmic Bits! Instant Bug Repellent! File Scout! Slither!

ISSUE #24—DEC. '85 $4.00
Speech synthesizers! The IBM Connection! The year's 25 best entertainments! And ready to enter: Gypsy Starship! Directory Manipulator! Cloak! Gameloader! Jewel Quest! Lineout! Santa's Busy Day!

ISSUE #25—JAN. '86 $4.00
Build a speech synthesizer! Survey of sports games! And ready to enter: The Martian Monsters! Streamer Font! Microsim! The Haunted Castle! Knockout! Infraraid! Alarm Clock! Memory Check!

ISSUE #26—FEB. '86 $4.00
Windows! Build an auto-exec cartridge! Align your 1541! Survey of flight simulators! Structured programming! And ready to enter: Arena! Head to Head! Crabfight! Treasure Wheel! Character Dump!

ISSUE #27—MAR. '86 $4.00
Programming educational games! Memory dumpers! Choosing a copy program! Custom characters! And ready to enter: Ahoy! Term 128! Trivia Game Maker! Brickbusters! Easy Lister! Programmer's Aid!

ISSUE #28—APR. '86 $4.00
Comet catching! Survey of action and strategy games! Screen dumping! And ready to enter: Chrono-Wedge! Mr. Mysto! Air Rescue! Notemaker! Screen Window! JCALC! Hidden Cavern! Swoop!

ISSUE #29—MAY '86 $4.00
128 graphic bit map! Epyx strategy guide! 128 commands! ML music programming! And ready to enter: Bigprint! Star Search! Failsafe! English Darts! Ski Folly! Free RAM Check! Alchemist's Apprentice!

ISSUE #30—JUNE '86 $4.00
Debugging dilemmas! Public domain software! Winning at Ultima! Computer Aided Design! And ready to enter: LazyBASIC! Got A Match? Star Strike! Queen's and Bishop's Tours! Shaker! Trackdown!

ISSUE #31—JULY '86 $4.00
Inside the Amiga! Conditional branching! Chess programs! 128 and 64 DOS! And ready to enter: Screen Sleuth! Escape from Skull Castle! Head-On! Nebergall Run! Wordcount! Crazy Joe!

ISSUE #32—AUG '86 $4.00
Inside the Amiga, part II! Approaching infinity! C-64 war simulations! Pascal for beginners! And ready to enter: Reversi! Highlight! Disk Cataloger! Meteor Run! Trim! Step On It! Flap!

ISSUE #33—SEP. '86 $4.00
Windows and viewports! Sound & music on the 64! COMAL! And ready to enter: The Last Ninja! Speech64! Multi RAM! Dogcatcher! Trapped! Matchblocks! Variable Manager! Dual Dump!

ISSUE #34—OCT. '86 $4.00
Build a digital oscilloscope! ML speed techniques! And ready to enter: Vault of Terror! Quick Change! Penguins! Attack Force! Disk Checkup! Dvorak Keyboard! Mountaineer Mack! 128 Autoboot!

ISSUE #35—NOV. '86 $4.00
C-128 shadow registers! Data file handling! PROMAL! Habitat! And ready to enter: Teleporter! 128 RAM Check! Discs of Daedalus! Guardian! Tenpins! Syntax Patrol! Deluxe List! Long Lines!

Use coupon or facsimile. If ordering more than three issues, list choices on separate sheet.

If you have a modem and want more complete information on any of the back issues listed, call *Ahoy!*'s Bulletin Board Service at 718-383-8909.

Ahoy!
BACK ISSUES
ION
INTERNATIONAL INC.
45 West 34th Street
Suite 407
New York, NY 10001

Please Send Me The Follcwing:

_____ Copies of issue number _____
_____ Copies of issue number _____
_____ Copies of issue number _____

Enclosed Please Find My Check or Money Order for $_____
(Outside the USA please add $1.00 for every copy)

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP CODE _____

they may march due south, then due east, or east then south, or even alternate—but they'll get there, unless they run into a rock (they'll stop and tell you). They attack anything not loyal to you, and may even capture a few castles on the way to their destination. When an army meets another (or a castle) you will see the starting strengths, and watch the numbers dwindle as they slug it out.

When a player has ordered all his armies, he hits the Commodore key and the next player directs his armies. Then the armies all move and fight. When all armies have moved five times, the Lords again direct their troops (including new levies that appeared in the castles you hold). A player may also drop out of the game, or all may quit. Each castle starts with a few troops. Yours will have a goodly number, but some of the rebel strongholds may be significant. The strongest ones will occasionally send out armies of their own—which weakens that castle's defense, but those marauding rebels may take back your castles, too. Your task, of course, is to take all the castles on the board.

This is a challenging and unpredictable game. Not only is the initial setup of castles, levies, and troops random, but the result of any battle is also. The defender always attacks first, and he may be much more effective than the attacker, so use lots of troops...if you have them.

I hope you enjoy this game as much as I do, either solitaire or in competition. If you have any questions, please write me at 247 Savannah Round, Summerville, SC 29483.

TECHNICAL NOTES

This program is interesting in a couple of areas. It is basically a game of having more armies in the right place than the "other guy," and taking control of the castles to get more armies. The display is the heart of the game. The board is made of custom characters read in from data statements (Lines 2890-2910; switched in 2970-2980; board printed 230-310). Sprites for the window, cross, and battle are also read in. Messages may be printed at the top by prefacing with H$ or at the bottom of the board with D$. The choice of joystick or keyboard is easy with lines 90-130: JF is non-zero if the fire button/f7 is hit, and JD contains direction. Lines 140-160 position sprite SN at row R, column C. This makes it easy to "roam" the board, looking at the territory (the information for an area is printed out by lines 770-1010). The board is too big for all the information to be printed, so this roaming window idea was used for information, and also for sending armies, rather than the usual "row, column" format. Most of the rest of the programming is, alas, bookkeeping! □ *SEE PROGRAM LISTING ON PAGE 139*

CADET'S COLUMN

Christmas Presents for You and Your Commodore

By Cheryl Peterson

December is usually the best month of the year for home computer sales, and I'm sure 1986 will continue the tradition. For me, that means a whole crop of beginning computer users. Some of our newcomers will be using the new C-64C. While I don't have one of the new machines, Commodore claims it is completely compatible with the old C-64, so this column should be useful to our new computing friends.

I am a bit concerned by Commodore's offering *GEOS* with the C-64C. While it will help new users get comfortable with the computer, I hope that those who use *GEOS* won't think they're limited to it. Those who struggled through learning about their computers can testify to the myriad tasks a computer can perform. *GEOS* touches only two of them.

The most difficult part of using a computer is learning that you can't hurt it (short of dropping it several feet to the floor). You may confuse and confound it by typing strange things on the keyboard, but you can't break it that way. Once you get that firmly entrenched in your mind, the computer becomes a vast new territory to be explored. A program like *GEOS* can help overcome the initial fear.

While *GEOS* does open the door, remember that it only leads into one room of the house. There is much more hidden underneath those icons and menus that won't be found until you find your way out of *GEOS*.

For those just buying their first computer, I would recommend getting a collection of *Ahoy!* back issues. You will find valuable programs to type in, programming hints, lessons on how a computer works, software reviews, and inside looks at the innards of these beasts. The articles can get a little technical, but most can be read and understood even by novice computerists.

There are a few other very good magazines. I highly recommend *INFO* magazine for its reviews and other product information. Another publication, *Transactor,* tends to be more technical and focuses on programming techniques and hints.

A valuable resource for anyone who is considering expanding his computer system is *Computer Shopper.* This "want ads" for the computer world includes many bargains and also carries articles on software, hardware, and applications of computer technology. They recently started a column on CP/M for the Commodore 128 and regularly run articles devoted to the C-64, C-128, and Amiga.

CHRISTMAS PRESENTS

For those who already have a computer, Christmas is a great time to expand your system. After all, the computer's a member of the family too.

One difficult decision is whether to expand the present system or upgrade to a new one. I can offer one bit of advice. If you have a computer that doesn't have a printer, a disk drive, and a modem hitched up to it, you are better off buying these items than getting a new computer. If you upgrade to a machine with more memory, higher screen resolution, etc., you will still have limited capability.

FREE!

BONUS GIFT WITH ANY SOFTWARE ORDER — AN EXCELLENT PUBLIC DOMAIN VERSION OF AN ARCADE HIT GAME ON ITS OWN DISK.

ARCADE

A.C.E.	$14.95
Bop and Wrestle	19.95
Commando	23.95
Dambusters	19.95
Elite	19.95
Fight Night	19.95
Flight Simulator II	32.95
Scenery Disks (Set of Six)	75.95
Star Scenery Disks	16.95
GBA Two on Two Basketball	23.95
Gary Kitchen's Game Maker	26.95
Great American Road Race	19.95
Hardball	19.95
Leader Board	26.95
MicroLeague Baseball	26.95
1984 or 1985 team disk	13.95
World Series or All Star disk	13.95
General Manager's disk	26.95
Box Scores/Stat Disk	16.95
Racing Destruction Set	12.95
Spitfire 40	23.95
Star Rank Boxing	21.95
Superbowl Sunday	20.95
1984 & 1985 Data Disks	13.95
Super Cycle	26.95
World Karate Championship	21.95

ADVENTURE AND MIND GAMES

Alter Ego (Male or Female)	$33.95
Astrology Horoscope Maker	15.95
Bard's Tale	25.95
Battlefront	26.95
Chessmaster 2000	28.95
Hacker II	21.95
Heart Of Africa	12.95
Hitchhiker's Guide	24.95
Lords of Conquest	26.95
Modern Computer People	23.95
Movie Maker	25.95
Nato Commander	24.95
Phantasie (or Phantasie II)	26.95
Rings of Zelfin	24.95
Seven Cities Of Gold	12.95
Strip Poker	20.95
Ultimate Wizard	19.95
Where is Carmen Sandiego	23.95
Wizard's Crown	27.95

EDUCATIONAL PROGRAMS

Bank Street Storybook	$27.95
Cave Of The Word Wizard	18.95
Donald Duck's Playground	16.95
Evelyn Wood's Dynamic Reader	35.95
Gertrude's Secrets	20.95
Grammer Examiner	30.95
Perfect Score S.A.T.	49.95
Reader Rabbit	26.95
Sky Travel	27.95
The Game show	23.95
Grade Level Subject Disks	13.95
Tic Tac Show	23.95
Subject Diskettes	13.95
Typing Tutor III	26.95
Winnie the Pooh	16.95

WORD PROC. - DATA BASES SPREAD SHEETS

Batteries Incl. - Home Organizer Series	
Audio/Video Catalog	$13.95
Checkbook	13.95
Electronic Address Book	13.95
Home Inventory	13.95
Mail List	13.95
Photos/Slides/Home Movies	13.95
Recipes	13.95
Stamps	13.95
Fleet System 2 (Limited Quantity)	29.95
Mirage Data Base Manager	38.95
Multiplan	42.95
Pocket Planner or Writer or Filer	27.95
Sideways	19.95
Superbase 64	52.95
Swiftcalc w/Sideways	35.95
Sylvia Porter's Pers. Finance	42.95
Visawrite 128	51.95

GRAPHICS & MUSIC

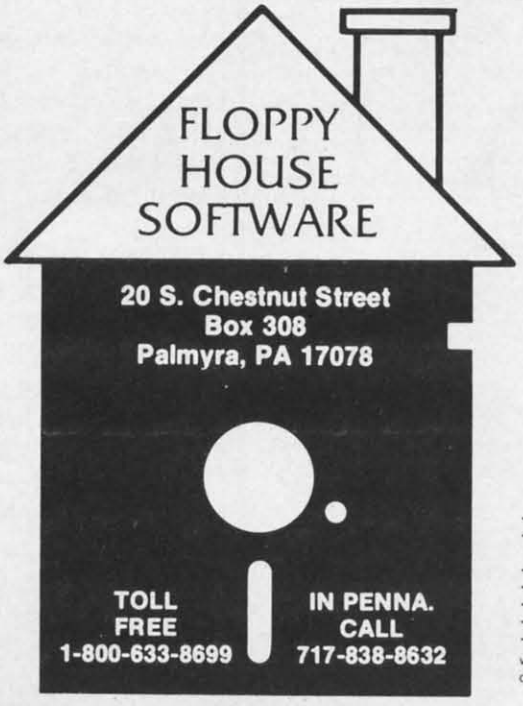
Adv. Music Sys. (MIDI compatible)	$49.95
B-Graph (limited quantities)	19.95
Billboard Maker	29.95
Computer Eyes	99.95
Comp. Eyes Compatibility Disks	12.95
Disney Card & Party Shop	19.95
Disney Comic Strip Maker	19.95
Doodle	26.95
Flexidraw	119.00
GEOS	40.95
Graphics Expander	26.95
Icon Factory	29.95
Music Studio	20.95
Newsroom	34.95
Newsroom Clip Art #1	21.95
Newsroom Clip Art #2	23.95
PrintMaster (Program)	26.95
PrintMaster Art Gallery 1 & 2	15.95
Print Shop (Program)	29.95
Print Shop Companion	23.95
Print Shp. Gr. Lib. 1 or 2 or 3	16.95
Screen Dumper 64	16.95

MISCELLANEOUS

1541 Physical Exam	$33.95
Clone Machine (Super Clone)	21.95
Copy II 64/128 V 2.8	26.95
CSM Disk Alignment	32.95
Disks (Boxes of 10 DS/DD)	8.95
Fast Load	26.95
Pick' Em '86	43.95
Snapshot	40.95
Super Kit 1541	23.95

COMMODORE 128 SOFTWARE

CAD Pack 128	$45.95
Mach 128 (accellerates 128 loads)	34.95
Paperback Planner 128	34.95
Paperback Writer 128	34.95
Partner 128	34.95
Superbase 128	69.95
Timeworks Data Manager 128	49.95
Timeworks Swiftcalc 128	49.95
Timeworks Word Writer 128	49.95



AMIGA SOFTWARE

Aegis Animator	$94.95
Aegis Draw	135.00
Aegis Images	54.95
Analyze (Spreadsheet)	69.95
Arctic Fox	29.95
Brataccus	35.95
Delta Patrol	16.95
Deluxe Paint	59.95
Art Disk	22.95
Deluxe Print	69.95
Art Disk	22.95
Deluxe Video	69.95
Hacker	31.95
Halley Project	31.95
Infocom Titles in stock	CALL
Marble Madness	35.95
Mind Shadow	31.95
Music Studio	42.95
One On One	29.95
Rogue	24.95
Scribble (Word Processor)	69.95
Seven Cities of Gold	29.95
Skyfox	29.95

HARDWARE AND PERIPHERALS

Anchor 6480 300/1200 BAUD Modem	$129.00
Disk Notcher (Interex)	5.95
Estes Power-Pack (Serviceable)	41.95
McPen (High-Res Light Pen)	40.95
Messinger Modem (300 BAUD)	49.95
MicroStuffer Buffer (64K)	59.95
Mighty Mouse	44.95
Omni-Tronics RS-232 Interface	39.95
PPI Interface	39.95
Power Pack (Maxtron or Commodore)	28.95
Pro-Tech Vinyl Cover C64 or C128	5.35
Vinyl Cover - 1541 or 1571	5.35
Vinyl Cover - Most printers	6.50
Vinyl Cover - Most Monitors	10.95
Ribbons - 1525/801 Printers	7.95
Ribbons - 1526/802 printers	8.95
Ribbons - Star SG10 1.95 - Color	3.95
Ribbons - Other	CALL
Six Outlet Surge Protector	21.95
Tac 2 Joysticks	9.95
Underwear Transfer Ribbons	10.95
Underwear Colored Pens (Set)	10.95
Xetec Super Graphic Interface	62.95

LEROY'S CHEATSHEETS **$2.95**

Basic	Practicalc
Blanks (3)	Practicalc II
Calc Result Adv.	Printers
Calc Result Easy	Epson RX80
Comal (1.4)	Gemini 10X
The Consultant	1525/801
Disk 1541	Microline
Doodle	Quick Brown Fox
Easy Script	Simon's Basic
Fleet System II	Sky Travel
Flight Simulator II	Speedscript
For the Beginner	Sprites Only
Hes Writer	Superbase 64
Logo Sheet 1	Terminals
Logo Sheet 2	Smart64
The Manager	Vidtex
Multiplan	VIP
Omniwritter	Word Pro 3+
Paperclip	Word Pro 64
GEOS	Elite

MAILING CHARGE ON LEROY'S CHEATSHEETS $1.00

ORDERING & TERMS

- VISA & Master Card - no additional charge
- $2.50 Shipping charge - Shipping free on orders over $100.00
- LEROY'S total mailing charge $1.00 for any quantity
- C.O.D. orders - add $1.90 extra
- PA residents add 6% Sales Tax
- FPO & APO - no additional charge
- We try to keep our prices as low as possible to serve our valued customers. In order to do this we are prosecuting credit card fraud to the full extent of the law

Reader Service No. 240

A computer without a disk drive is like a car running on two cylinders. It'll get you there—eventually. The wait for programs to load from tape can seem interminable, and most programs aren't even offered on tape anymore.

For those contemplating upgrading from a C-64 to a C-128 in the near future, you might consider buying a 1571 disk drive instead of a 1541. The 1571 will work with the C-64, and you won't need to upgrade your drive later.

After buying my initial computer system (computer, monitor, and disk drives), the first peripheral I added was a printer. At that time, a printer that could provide graphics and near letter quality print cost $1500. I've never regretted that purchase. Admittedly, the majority of you aren't writers. But now you can buy the same printer for less than $400.

Having such a printer will let you print out letters, labels for addressing your Christmas cards (but then you've done that already this year, haven't you?), database information (reports of your household items for insurance forms, for instance), school reports and papers, and drawings created with graphics packages. If you do your taxes by computer, some programs will even print your IRS forms.

If you decide to go for a printer neither manufactured by Commodore nor designed to be Commodore-compatible, you'll need to buy a printer interface as well. A number of companies market these. One, Cardco, recently went bankrupt, so think twice about buying any of their interfaces unless you can get a very good price on it.

While these first two add-ons (disk drives and printers) seem obvious to many, the wonders of adding a modem to your computer almost have to be experienced to be appreciated. The simple explanation that attaching a modem to your computer and a phone line will let you communicate with other computers just doesn't convey the extent to what you can do with it.

Telecommunicating can put you in touch with hundreds of other users who can offer tips and advice on getting more from your computer. By signing up with a commercial system like PlayNET, QuantumLink, or CompuServe, you can type messages on your keyboard that will be seen by a dozen or more people all over the country almost instantly. You can make friends and play games with other people without leaving your own home. In most large cities it only takes a local telephone call to get online.

Public domain programs can be transferred to your computer through a modem, allowing you to do many tasks more easily. These programs can be found in special sections of the commercial services or on local BBS's (Bulletin Board Systems). If you can find a copy of *Computer Shopper*, they run an updated listing of such BBS's in each issue. The numbers are listed by area code, making it easy to find those that are near you.

Another good buy for Christmas is a starter kit or registration for one of the online services. Most have a toll-free number you can call to sign up if you have a major credit card (MasterCard, Visa, and sometimes American Express). The sign-up charges vary for each service. In some cases, you can buy a starter kit at computer stores.

Each system has a schedule of charges for different services. In some cases, the speed of your modem determines the rate. Some charge extra for 1200 baud. While PlayNET costs less than $3 an hour, CompuServe, The Source, Delphi, and GEnie run $5 to $8.40 an hour at 300 baud during non-prime time hours. Prime time on these services is generally Mon-Fri 8 a.m. to 5 p.m., local time. The charges during prime time are usually twice that of non-prime, if the service is available during the day.

OTHER PERIPHERALS

Some alternate input devices make nice presents for your computer. A new joystick or two can add a lot more fun for the whole family. Since the Commodore 1350 mouse will work with both the C-64 and the C-128, it's another good choice. Some mailorder houses sell the mouse for as little as $39.

Koala Pads are quite inexpensive these days. With one of these and the right graphics software, you can draw pictures on the pad using a stylus. There are even a few children's games that have overlays for the Koala Pad. Other drawing tablets are also readily available.

Another input device that can be fun to use is the light

pen. Like the drawing pads, they are mostly used for drawing pictures. Special software is usually required, but frequently a drawing program is sold with the light pen.

PROTECTING YOUR BEST FRIEND

Several companies offer power strips or power filtering devices that will help protect your computer from electric surges and glitches. I use a MasterPiece Plus, which is a swivel base that sits under the monitor. The back panel has AC power connectors into which you plug the computer, monitor, printer, and two auxiliary items. I use the two extras for my modem and disk drive. The MasterPiece Plus then plugs into the wall. A master switch on the front panel is used to turn everything on. There are also five switches that make it easy to turn off any of the five items. For instance, if I'm not using the printer or modem, I can disable them by pressing their switch.

The MasterPiece Plus also has a telephone line filter. You plug the phone line into the unit and then run another line to your modem. If you get a power surge down the phone line, this should protect your modem.

If you live in an area where thunderstorms and power surges are a problem, this $150 add-on could save you from some expensive repairs. While it won't protect you against a direct lightning strike to your power pole, it will handle most surges. Ordinary filtering or surge-protecting power strips run as little as $30.

SOFTWARE

There are so many good software packages out there, it is hard to list them all. Rather than slight anyone, I would recommend checking reviews in back issues of *Ahoy!* to find the ones that are really outstanding.

As long as you're picking presents for the computer, don't forget to pick up something fun for yourself. *Ahoy!'s* entertainment software section covers many enjoyable and exciting games. Besides shoot-em-ups aimed at space-happy teenagers, there are strategy and simulation packages that challenge intellect as well as reflexes. Text adventure games can be extremely devious.

CARTRIDGE PRODUCTS

Anyone using a Commodore disk drive should consider getting one of the cartridge products that increase the speed of disk loads and saves. Epyx's *Fast Load* and Access Software's *Mach 5* make the drive work up to five times faster. Since the Commodore's disk drives are notoriously slow, this is one of the best Christmas presents you can get.

Access also markets the *Mach 128* cartridge. While it is designed to work with a C-128 and 1571 drive, a switch on the cartridge's top makes it compatible with

the C-64 and 1541. If you plan to upgrade to a C-128 someday, it makes sense to buy the slightly more expensive *Mach 128* cartridge.

Timeworks' recently announced *Partner 128* cartridge (reviewed last month) looks like a great collection of frequently needed utilities. It includes a quick loading routine, but also offers a memo pad, appointment calendar, telephone and address book, text-only screen printer, and an address label printer that can use the address book's entries. All of these reside on the cartridge and are supposed to be accessible from inside most C-128 programs. Timeworks has plans to release a C-64 version, but it wasn't ready at press time. It is scheduled to reach stores in November, so you should be able to try one out for Christmas.

Online Services

CompuServe
Starter Kit: $39.95
5000 Arlington Centre Blvd.
P.O. Box 20212
Columbus, OH 43220
Phone: 800-848-8199 (in OH: 614-457-0802)

Delphi
Starter Kit: $49.95
3 Blackstone St.
Cambridge, MA 02139
Phone: 800-544-4005
(in MA: 617-491-3393)

GEnie
Registration Fee: $18
401 North Washington Street
Rockville, MD 20850
Phone: 800-638-9636
ext. 21

PeopleLink
Starter Kit: $10
3215 North Frontage Road,
Suite 1505
Arlington Heights, IL 60004
Phone: 800-524-0100

PlayNET
Starter Kit: $15
200 Jordan Road
Troy, NY 12180
Phone: 800-PLAYNET

The Source
$49.95 plus $3 shipping and handling
1616 Anderson Road
P.O. Box 1305
McLean, VA 22101-9881
Phone: 800-336-3366 (in VA: 703-821-6666)

Magazines

Computer Shopper
407 S. Washington Ave.
P.O. Box F
Titusville, FL 32781
Phone: 305-269-3211
1 yr./12 issues: $21

INFO
P.O. Box 2300
Iowa City, IA 52244
1 yr./6 issues: $18

Transactor
Subscription Dept.
500 Steeles Ave.
Milton, Ontario, Canada L9T 3P7
1 yr./6 issues: $15
Phone: 416-878-8438

CONCLUSION

As you see, there are any number of ways to expand and improve a computer system. You could probably do half your family Christmas shopping at your local computer store. A copy of *Print Shop* and *Hitchhiker's Guide to the Galaxy* for Mom; *Newsroom, Quink, The Little Computer People Discovery Kit, Rescue on Fractulus, Mindwheel, Questron, Music Studio,* and maybe a few modules of *Success With Math* for the children; and for Dad, *Super Bowl Sunday, Flight Simulator II,* and *MultiPlan.* Add a piece or two of hardware, magazine subscriptions, and maybe a couple of books on computer use, and you won't have room for socks, pajamas, and neckties. In all honesty, wouldn't you rather have a few fresh printer ribbons than another necktie? □

TIPS AHOY!

By Michael R. Davila

CAREFREE ANIMATION

The most difficult problem I have encountered while programming games in BASIC is keeping the sound, program logic, and action going simultaneously. To simplify this, I wrote this ML routine that will animate sprite #7 using the three sprite definitions in blocks 13-15.

To use this in your own programs, simply place your sprite data in blocks 13-15, run the machine code loader, and type SYS 679. To vary the speed of the animation, POKE 738 with a number between 3 and 10. The lower the number the faster the animation. (The default value is 10.) Enter and run the example program to see a mythical beast take a carefree stroll.

—Bob Ash
APO, NY

```
10 REM - CAREFREE ANIMATION -
20 V=53248:S=7:GOSUB100
30 POKE53281,0:POKE53280,15:POKE646,11:P
RINT CHR$(147)
40 POKE V+23,(2^S):POKEV+29,(2^S)
50 POKEV+39+S,7:POKE V+(S*2+1),150
60 POKE V+21,PEEK(V+21) OR (2^S)
70 FOR I1=255TO50STEP-1:FOR I2=1TO8
80 POKE V+(S*2),I1:NEXTI2:NEXT
90 END
100 REM ** SPRITE DATA LOADER **
110 FOR I1 = 832 TO 1023: READ SP: POKE
I1,SP: NEXT
120 DATA000,000,000,000,000,000,000,096
130 DATA000,001,096,000,003,032,000,003
140 DATA120,000,003,244,000,063,252,000
150 DATA001,096,000,000,113,224,000,127
160 DATA252,000,255,246,000,127,246,000
170 DATA096,227,000,112,097,000,112,112
180 DATA000,096,232,000,192,080,000,096
190 DATA096,000,064,064,000,000,000,000
200 DATA000,000,000,000,000,000,000,096
210 DATA000,000,096,000,001,032,000,003
220 DATA120,000,003,248,000,255,240,000
230 DATA003,096,000,001,113,224,000,127
240 DATA252,000,255,246,000,127,242,000
250 DATA224,230,000,192,098,001,192,112
260 DATA001,064,112,001,032,144,001,096
270 DATA216,001,064,160,000,000,000,001
280 DATA000,000,000,000,000,000,000,096
290 DATA000,001,096,000,003,032,000,003
300 DATA120,000,003,244,000,063,252,000
310 DATA001,096,000,000,113,224,000,127
320 DATA252,000,255,246,000,127,246,000
330 DATA096,227,000,112,097,000,112,112
340 DATA000,096,232,000,192,080,000,096
350 DATA096,000,064,064,000,000,000,000
360 REM *** MC LOADER ***
370 FOR I1 = 679 TO 753: READ MC: POKE I
1,MC: NEXT:SYS 679
380 DATA 169,2,141,54,3,141,53,3,120,169
390 DATA 188,141,20,3,169,2,141,21,3,88
400 DATA 96,238,54,3,173,54,3,201,1,240
410 DATA 12,201,10,144,37,169,0,141,54,3
420 DATA 24,144,29,238,53,3,173,53,3,201
430 DATA 3,208,10,169,0,141,53,3,169,12
440 DATA 141,52,3,238,52,3,173,52,3,141
450 DATA 255,7,76,49,234,0
460 RETURN
```

POKE AND FIND MESSAGES

These two short programs may help when programming in machine language or discovering messages in your machine. It should work on any machine, but I've only tried it on the C-64. The first, *POKE Message,* allows you to type in a message and it will POKE in the ASCII values of the message into the appropriate memory locations. This is useful when you want to put replace a message somewhere in memory. It would be very tedious if you had to POKE each individual character.

The second program, *Find Message*, will try to find a message in memory that is between the starting and ending addresses which you specify. It will tell you if it is not found. You could look for BASIC keywords or error messages in memory, or look for other messages. If you look for a BASIC message, the start address would be 40960 and the end address would be 49152. When you type in the message, capitalize the last letter. For example, if you were looking for 'NEXT', type: nexT. This is how BASIC knows where the end of a message is.

In both programs, when you input the numbers, input them in decimal. If you want to use special characters when typing in the message (such as CLR/HOME or color keys), type a quote mark before you enter the message.

—John Chong
Syracuse, NY

POKE MESSAGES

```
0 REM POKE MESSAGES
1 INPUT"ADDRESS: ";A:INPUT"MESSAGE: ";B$
:L=LEN(B$)
2 FORI=1TOL:POKEA-1+I,ASC(MID$(B$,I,1)):
NEXT
3 PRINT"MESSAGE RESIDES FROM"A"-"A - 1+L
:PRINT"AND IS "L"CHHARACTERS LONG"
```

FIND MESSAGES

```
0 REM FIND MESSAGE
1 INPUT"START";S:INPUT"END";E:INPUT"STRI
NG TO SEARCH FOR";X$:L=LEN(X$)
```

```
2 FORI=1TOL:IFPEEK(S-1+I)=ASC(MID$(X$,I,
1))THEN NEXT:PRINTX$,S,S+L:END
3 I=L:NEXT:S=S+1:IFS=ETHENPRINT"NOT FOUN
D":END
4 GOTO 2
```

ONE MORE FUNCTION KEY!

Okay folks, hold your breath on this one. I am a two-drive programmer who flicks disks in and out like an epileptic centipede—and I get tired of typing "CATALOG D0,U9" every time I want to see a directory of drive 9.

The problem is that the currently defined function keys are so well chosen for their initial bootup contents that it is a shame to have to redefine one.

Wa-lah! A quick look at Jim Butterfield's memory maps shows the HELP key has a five byte ASCII buffer of its own! Since I hardly ever need the HELP command, I figured out a bare bones, no frills abbreviated CATALOG call to device 9 that just squeezes in with a carriage return. Once this commnd is in the buffer, it is impervious to RUN STOP/RESTORE!

In the program listing are a few other useful definitions, depending on your preference. Each one is all on one line number by itself, so load in the program when you boot up your 128, list the program on the screen, and type NEW.

Then cursor up to the line number of the command you want to define, hit return (so the line is the only line number in memory, and clear the screen and RUN.

If you need the HELP key for debugging, you can still use the ultra-abbreviated form for the Directory on device 9—"C(SHIFTA)U9"

This tip has been a real convenience to me in using two drives.

—Cleve Blakemore
Richmond, VA

```
10 REM REDEFINES THE HELP KEY ON THE C-1
28 TO READ ERROR CHANNEL OFF THE DRIVE
20 FORX=4168 TO 4171:READA$:POKEX,ASC(A$
):NEXT:POKE4172,13:DATA ?,D,S,$
30 REM REDEFINES THE HELP KEY ON THE C-1
28 TO DISPLAY THE DIRECTORY ON DRIVE 9
40 FORX=4168 TO 4171:READA$:POKEX,ASC(A$
):NEXT:POKE4172,13:DATA C,"[SHIFT A]",U,
9
50 REM REDEFINES THE HELP KEY ON THE C-1
28 TO GOTO C-64  MODE
60 FORX=4168 TO 4171:READA$:POKEX,ASC(A$
):NEXT:POKE4172,13:DATA G,0,6,4
70 REM REDIFINES THE HELP KEY ON THE C-1
28 TO ENABLE PROGRAM TRACING
80 FORX=4168 TO 4171:READA$:POKEX,ASC(A$
):NEXT:POKE4172,13:DATA T,R,O,N
```

HIGH RESOLUTION ML ASSISTANCE

When using high-resolution graphics on the Commodore 64, a total of 9000 bytes of memory locations must be POKEd. It is very time consuming to POKE each individual location using BASIC. The following ML routine using a BASIC program loader would be faster and more practical to handle this tedious job. The routine will set up a high resolution screen and change the background and border colors to black when you type "SYS 49152". To change the border color, you simply change the "0" in line 30 to the POKE code of the color you desire. If you would like to alter the background color, change the second "0" in line 60 to the color POKE code of your choice. By typing "SYS 49241", the screen will clear and revert back to text mode. If you change the "14" or the "6" in line 90 to another POKE code, the screen background or border in text mode will produce the corresponding colors, respectively. Be sure to change the correct value of the variable "DT" in line 20 by adding or subtracting the corresponding value of any revised POKE codes or an 'ERROR IN DATA STATEMENTS' message will occur.

—Stanley C. Evans
Bahama, NC

```
10 FOR LOC=49152TO49272:READ PKODE:POKE
LOC,PKODE:DT=DT+PKODE:NEXT
20 IF DT<>15710THENPRINT"ERROR IN DATA"
30 DATA 173,24,208,9,8,141,24,208,173,17
,208,9,32,141,17,208,169,0,141,32,208
```

```
40 DATA 169,0,133,178,169,32,133,179,162
,32,160,0,169,0,145,178,136,208,251
50 DATA 230,179,202,48,2,208,244,162,96,
32,80,192,169,0,133,178,169,4,133,179
60 DATA 162,3,160,0,169,0,145,178,136,20
8,251,230
70 DATA 179,202,48,2,208,244,162,236,160
,0,145,178,200,202,208,250,96
80 DATA 173,24,208,41,247,141,24,208,173
,17,208,41,223,141,17,208
90 DATA 169,14,141,32,208,169,6,141,33,2
08:REM SET BACKGROUND & BORDER COLORS
100 DATA 169,147,32,210,255,96:REM CLEAR
 SCREEN
```

NO PRINT

Here is a little BASIC utility for the C-128 that enables a programmer to give the user instructions, menu options, or documentation without PRINT or the weird ol' REM. The trick is to turn off the line numbers is Line 20 (POKE24,27), since the 128 doesn't halt the program execution with a LIST command. It's handier than listing REM statements, or running PRINT with data statements. Other 128 features can be used when programming instructions in this style also, such as the escape codes (Delete, Insert, etc.), or functions such as auto line numbering. I've found it handy, especially in longer applications. Combined with the other features of the 128, when programming instructions or documentation, it is reminiscent of a very simple word processor.

—Barri Olson
Madison, WI

```
10 REM NO PRINT/REM DOCUMENTATION//BARRI
 OLSEN//80 COL VER
20 POKE24,37:LIST30-50:POKE24,27:RUN60
30 SEE HOW EASY THIS MAKES CREATING TEXT
 ON THE C-128. A LIST WITHIN
40 A PROGRAM DOESN'T STOP IT. FOR 40 COL
. OR WINDOW USAGE ADJUST SPACING
50 TO SEND TO YOUR PRINTER, OPEN A CHANN
EL AND USE 'CMD' -LIST-LINE# IN LINE 20
60 PRINT"SEE! -AND IT'S HANDY FOR LONG A
PPLICATIONS ":END
```

FAST SIMULATOR

For 64 programmers there exists a method of simulating the 128's FAST mode, at least to a degree. To demonstrate, type in and run the following lines. Remember that there are 60 jiffies to 1 second.

```
10 TI$="000000"
20 FOR T=1TO1000:NEXT
30 PRINT TI$; " JIFFIES"
```

Now add these lines to the program. Do not be alarmed when the screen blanks when you run the program again.

```
5 PRINT"STANDBY...":FORT=1TO500:NEXT
6 POKE 53265,0: POKE 56325,255
25 POKE 53265,27: POKE 56325,46
```

Note that the FOR loop in Line 20 took a shorter time to execute with the added lines. The VIC chip refreshes the screen display 60 times a second. POKE 53265,0 disables the VIC chip, giving the 6510 chip more time to process BASIC. POKE 56325,255 gives the 6510 less time to scan the keyboard and more time to process BASIC. POKE 53265,27 and POKE 56325,46 set things back to normal.

These POKEs would be most helpful in a portion of a program in which the user needs to neither see the screen nor use the keyboard, such as a sort routine or a READ...DATA routine. The program should warn the user about the blank screen before the VIC chip is disabled.

—Mark Bersalona
Philadelphia, PA

PERFECT CIRCLE

For *Ahoy!* readers with C-128's: When using the graphics in hi-res to produce a perfect circle, the proper ratio between the X radius (Xr) and the Y radius (Yr) is 5:4. If Xr is known, Yr can be calculated by Yr=(Xr/5)*4. If Yr is known, Xr can be calculated by Xr=(Yr/4)*5. Some sample commands are "CIRCLE 1,100,90,30,(30/5)*4" or "CIRCLE 1,100,90,(30/4)*5,30". —Andy Hatchell
Durham, NC

SAVE SCREEN

Did you ever wish to save your game screen directly to disk? If the answer is yes, here is a short utility you'll enjoy. Simply type the line shown below or insert it in your own program.

To LOAD your game screen type LOAD"filename",8,1 and then type POKE 53281,color and your screen will appear.

—Marco Gauthier
Fitch-Bay, Canada

```
SYS 57812"filename",8:POKE 193,0:POKE
194,4:POKE 174,231:POKE 175,7:SYS 62954
```

COMBINER

To combine two or three programs into one larger program, a lot of work is needed. The easier way is to use an append routine. This will save your fingertips from doing unnecessary work.

To append, load the first program into memory. You can list it if you want. Then type (in direct mode):

```
POKE 43,PEEK(45)-2 : POKE 44,PEEK(46)
```

When you try to list it, no program lines are listed, because the computer tries to start at the end of the program.

Then load the second program. You can list it if you want. (Note: The first line number of the second pro-

gram must be larger than the last line number of the first program.) Then type (in direct mode):

```
POKE 43,1 : POKE 44,8
```

When you type LIST and RETURN, you will find that the first and the second program have been appended.

Note: When you get 'OUT OF RANGE ERROR', try to add an unnecessary line in the program such as:

```
0 REM
```

—Adzhar Hamdan
Syracuse, NY

BASIC SCREEN COPIER SUBROUTINE

The following subroutines for the C-128 and C-64 can be used within BASIC programs to copy the screen to a printer. The normally slow and complex task of converting screen codes to ASCII code is speeded up very simply by using the first screen location as an input device. The screen codes from the entire screen are POKEd into the first location in turn and the GET#3 command is used to convert the screen code into its ASCII equivalent. The variable L$ is used to hold a screen line of text, and then it is sent to the printer. The somewhat slow rate of the copy is made up for by how simple and short the routine is. To use within a program, simply print your output to the screen as desired, then set the variable SA to 0 or 7 for UPPER CASE/GRAPHIC or UPPER/LOWER CASE, then call the subroutine and the screen will be sent to the printer (example: SA=0:GOSUB10 or SA=7:GOSUB10). You may of course change the line numbers to fit your own program. —Ernest R. Hunter
Long Beach, MS

C-128 VERSION

```
10 B=1024:A=PEEK(B):OPEN3,3:OPEN4,4,SA:F
AST:FORJ=0TO24:REM"FOR THE C-128"
20 L$="":FORI=0TO39:POKEB,PEEK(B+J*40+I)
:POKE244,0:PRINT"[HOME]";:GET#3,A$
30 L$=L$+A$:NEXT:PRINT#4,L$:NEXT:POKEB,A
:PRINT#4:CLOSE4:CLOSE3:SLOW:RETURN
```

C-64 VERSION

```
10 B=1024:A=PEEK(B):OPEN3,3:OPEN4,4,SA:F
ORJ=0TO24:REM"FOR THE C-64"
20 L$="":FORI=0TO39:POKEB,PEEK(B+J*40+I)
:POKE212,0:PRINT"[HOME]";:GET#3,A$
30 L$=L$+A$:NEXT:PRINT#4,L$:NEXT:POKEB,A
:PRINT#4:CLOSE4:CLOSE3:RETURN
```

1571 SINGLE SIDED MODE

If your commercial software worked with your 128/1541 system and won't with an upgraded 128/1571 configuration, try changing the 1571 disk drive to single sided mode. This can be done in direct mode, before loading, or added as a line in the BASIC loader used by most store-bought software. The loader is almost always the first program on a commercial disk. The following command accomplishes this. The drive must be on.

—Edward Horgan
Coatesville, PA

```
OPEN15,8,15: PRINT#15,"U0>M0":CLOSE15
```

OUTER SPACE WITH JUST FOUR LINES

This four line program shows how powerful the C-128 BASIC 7.0 graphic and sprite commands are. The program draws a universe complete with planets orbiting around a sun along with stars and shooting asteroids. It even adds a bit of space sound. It would require many BASIC lines to do this on the C-64, and you would probably have to resort to machine language to get the speed. You can study the four very simple lines for ideas for forming screens for games or other programs requiring graphics, sound, and animation. —Ernest R. Hunter
Long Beach, MS

```
10 COLOR4,1:COLOR1,8:COLOR0,1:GRAPHIC1,1
:FORJ=0TO90 STEP 30:CIRCLE1,25,22,0,8,,,
J:NEXT:SSHAPEA$,11,10,34,31:SPRSAVA$,7:S
PRSAV7,8:GRAPHIC1,1:MOVSPR7,260 #9
20 MOVSPR8,145 #12:SPRITE7,1,6:SPRITE8,1
,9:GRAPHIC1,1:CIRCLE1,25,22,6,6:PAINT1,2
6,26:SSHAPEB$,11,10,34,31:GRAPHIC1,1:FOR
J=1TO100:H=RND(0)*(35):V=RND(0)*24
30 CHAR0,H,V,".    .":NEXT:CIRCLE1,150,10
0,10,10:PAINT1,151,101:FORJ=1TO6:SPRSAVB
$,J:SPRITEJ,1,J+3,1:MOVSPRJ,160+J*20,140
:CIRCLE1,150,100,J*20,J*20:NEXT
40 VOL8:SOUND1,55535,250,0,42768,1000,2,
2600:FORJ=0TO2*[PI] STEP .5:FORI=1TO6:X=
160+(I*20)*COS(J-I*20):Y=140+(I*20)*SIN(
J-I*20):MOVSPRI,X,Y:NEXT:NEXT:GOTO40
```

INSTANT RUN

Here is a tip that eliminates the need to type RUN. Lines 10 and 20 copy BASIC and Kernal into RAM memory. Line 30 switches the 64 from ROM BASIC to RAM BASIC. Line 40 prevents RAM BASIC from being switched out by RUN STOP/RESTORE. Lines 50 and 60 change the READY. prompt to the word RUN and puts the cursor on the word RUN. To run a program after loading it in, simply hit RETURN. This is especially helpful in debugging programs, where you must run a program several times to check for bugs and to test for corrections. —Mark Bersalona
Philadelphia, PA

```
10 FOR X=40960TO49151:POKEX,PEEK(X):NEXT
20 FOR X=57344TO65535:POKEX,PEEK(X):NEXT
30 POKE1,53:REM SWITCH IN RAM BASIC
40 POKE64982,53:REM PROTECT FROM RESTORE
50 FOR X=41848TO41853:READN:POKEX,N:NEXT
60 DATA 82,85,78,141,145,145
```

COMMODARES
PROGRAMMING CHALLENGES

By Dale Rupert

Each month, we'll present several challenges designed to stimulate your synapses and toggle the bits in your cerebral random access memory. We invite you to send your solutions to:

Commodares, c/o *Ahoy!*
P.O. Box 723
Bethel, CT 06801

We will print and discuss the cleverest, simplest, shortest, most interesting and/or most unusual solutions. Be sure to identify the *name* and *number* of the problems you are solving. Also show sample runs if possible. Be sure to tell what makes your solutions unique or interesting, if they are.

Programs on diskette (1541 format only) are welcome, but they must be accompanied by listings. You must enclose a stamped, self-addressed envelope if you want any of your materials returned. Solutions received by the middle of the month shown on the magazine cover are most likely to be discussed, but you may send solutions and comments any time. Your original programming problems, suggestions, and ideas are equally welcome. The best ones will become *Commodares*!

PROBLEM #36-1: FANCY FACTORIALS

This problem was submitted by Jim Speers (Niles, MI). Write a program to calculate any factorial (up to 1000 factorial). For example, 5! (five factorial) is 5 * 4 * 3 * 2 or 720. Jim's solution calculates factorials exactly, giving all 869 digits of 400 factorial. Both exact and approximate solutions will be considered.

PROBLEM #36-2: ANIMATED EXPANSION

The user enters a word. The computer displays it at the left of the screen and then proceeds to expand the word to the right, accordion fashion, by inserting a specified number of spaces between letters.

For example, if the number of spaces is two, and the word is Ahoy, the sequence looks like this (all on the same screen line):

```
Ahoy
A-hoy
A--hoy
A--h-oy
A--h--oy
A--h--o-y
A--h--o--y
```

where " - " represents the added spaces. No POKEs allowed. Can your solution be easily modified to reverse the process and squeeze the word back together?

PROBLEM #36-3: LIFE TIMES

Here is one from Bill Okerblom (Providence, RI) that should be interesting even for first-time programmers. Write a program which prints out the dates for the previous 100 lifetimes of the user. If the user is 30, the program displays something like this:

```
1) 1956     2) 1926     3) 1896     4) 1866
... 99) 984 BC     100) 1014 BC
```

Bill suggests that it gives a different perspective to see how many of your lifetimes ago (rather than years or centuries) America was discovered or the pyramids were built.

Since the year 1 AD (not 0 AD) followed the year 1 BC, the "BC" years in the example are off by one. You can fix that problem if you want, although one year doesn't really matter here.

PROBLEM #36-4: CROSSED LADDERS

Some of you may recall, as I do, the agony of trying to solve this classic problem algebraically. Does the computer make it any easier? Two tall buildings face each other across a narrow street. A 30 foot ladder goes from the base of building A and leans against building B. A 20 foot ladder goes from the base of building B to the face of building A. The point at which the ladders cross is 6 feet above the ground. How wide is the street? Sounds awfully easy, doesn't it?

We will begin this month with another solution to a problem discussed last month, *#31-4: Repeated Compression*. The problem is to compress a string which contains repeated adjacent characters and to print it without the duplicate letters. For example, "COMMOOODAAARRESS" becomes "COMODARES" (with one "M"). This

solution from Sol Katz (Lakewood, CO) is unusual since it uses recursion in a COMAL program to do the job.

```
0001 // COMMODARES PROBLEM #31-4
0002 // REPEATED COMPRESSION
0003 // COMAL SOLUTION BY
0004 // SOL KATZ
0005 //
0010 DIM STRING$ OF 80
0020 PRINT
0030 INPUT "ENTER STRING TO COMPRESS>> "
: STRING$
0040 COMPRESS(1) // START WITH FIRST CHA
     RACTER
0050 // END OF PROGRAM //
0060 // THIS IS THE ROUTINE (PROCEDURE)
0070 //      INVOKED BY LINE 40
0080 PROC COMPRESS(POS)
0090  IF STRING$(POS)<>STRING$(POS-1) TH
EN PRINT STRING$(POS),
0100  IF POS<LEN(STRING$) THEN COMPRESS(
POS+1) //THE RECURSIVE CALL
0110 ENDPROC COMPRESS
```

The main part of the program is line 40 which "calls" the routine beginning at line 80. The COMPRESS procedure calls itself (that's the recursion) in line 100, each time with the value of the argument POS incremented by one. Once POS is equal to the length of the input string, the routine branches to line 110 which essentially returns to line 50 to end the program.

Recursion is not necessarily an efficient or desirable solution. Some readers chastised me for my recursive BASIC permutation program a few months ago, which in fact was many times slower than the more straightforward solutions (not to mention the fact that it quickly ran out of stack space). This solution is presented as an interesting programming technique.

The statement COMPRESS(POS+1) in line 100 looks like it should be equivalent to the BASIC statements POS=POS+1 : GOSUB 80. For all practical purposes in this program they are equivalent. In actuality, the recursive COMAL routine does not change the value of POS. Each invocation of the COMPRESS routine has its own value of POS. This is most easily understood if we add line 105 PRINT POS. In the BASIC equivalent, only one value of POS (the last value) would be printed. In the recursive program, each value assigned to POS (from largest to smallest) will be printed.

This BASIC program behaves the way that the recursive COMAL program behaves:

```
·5 REM -- BASIC RECURSION --
·10 S$="TTEE[3"S"]TT"
·20 N=1 : P(N)=1 : GOSUB 80
·30 END
·80 P=P(N)
·90 IF MID$(S$,P,1)<>MID$(S$,P+1,1) THEN
 PRINT MID$(S$,P,1);
·100 IF P<LEN(S$) THEN N=N+1: P(N)=P+1: G
 OSUB 80
·105 PRINT P(N);
·110 N=N-1 : RETURN
```

Here N keeps track of the level of recursion. P(N) keeps the value of P at each level. P in this program is equivalent to POS in the COMAL version. When P=9, the conditional statement in line 100 is false, and line 105 is executed, printing the value 9. Since level 9 was called from line 100 of level 8, the RETURN in line 110 of level 9 brings the program back to line 105 of level 8 which prints 8, and so forth. Consequently the equivalent of recursion is possible in BASIC, but you must keep track of the levels and the variables at each level since BASIC doesn't do that for you.

Problem #32-1: Diligent Decoder was the counterpart to *Problem #28-4: Elegant Encoder*, both from Jim Speers (Niles, MI). Each letter in a word is encoded by replacing it with a letter whose numeric value equals the sum (modulo 29) of the other letters' values in the word. 29 characters are allowed (@, A-Z, [, and <English pound>). Modulo 29 (MOD 29) means to continuously subtract 29's from the value until it is less than 29. Your task was to decode such an encoded word.

As promised in the August issue, here are the encoder/decoder solutions from Matt Shapiro (Fort Lee, NJ) and from David Hoffner (Brooklyn, NY).

```
·1 REM COMMODARES PROBLEM #32-1:
·2 REM    DILIGENT DECODER
·3 REM SOLUTION BY
·4 REM    MATT SHAPIRO
·5 REM (ALSO #28-4:ELEGANT ENCODER)
·6 REM
·10 DIM W(28):DEF FNM(X)=X-29*INT(X/29)
·20 INPUT"WHAT'S YOUR WORD";W$
·30 PRINT"[RVSON]E[RVSOFF]NCODE OR [RVSON
 ]D[RVSOFF]ECODE?  ";
·40 GET A$:IF A$<"D" OR A$>"E" THEN 40
·50 PRINT A$:L=LEN(W$):S=0:IF L<2 OR L>29
  THEN 20
·60 FOR I=1 TO L:W(I)=ASC(MID$(W$,I,1))-6
 4:S=S+W(I):NEXT:W$="":IF A$="E" THEN100
·70 S=FNM(S)
·80 Q=S/(L-1):IFQ<>INT(Q) THEN S=S+29:GOT
 O 80
·90 S=Q
·100 FOR I=1 TO L:W$=W$+CHR$(FNM(S-W(I))+
 64):NEXT:PRINT W$:END
```

```
·1 REM COMMODARES PROBLEM #32-1:
·2 REM    DILIGENT DECODER
·3 REM SOLUTION BY
·4 REM    DAVID HOFFNER
·5 REM (ALSO #28-4:ELEGANT ENCODER)
·6 REM
```

```
·200 INPUTA$:A=LEN(A$):DIMF(A):FORX=1TOA:
 F(X)=ASC(MID$(A$,X,1))-64:C=C+F(X):NEXT
·210 FORX=1TOA:B=C-F(X):B=INT(29*(B/29-IN
 T(B/29))+64.1):B$=B$+CHR$(B):NEXT
·220 PRINT B$:FORX=1TOA:F(X)=ASC(MID$(B$,
 X,1))-64:D=D+F(X):NEXT:L=A-1
·230 IF D/L>INT(D/L) THEN D=D+29:GOTO 230
·240 FORX=1TOA:G=D/L-F(X):G=INT(29*((G/29
 )-INT(G/29))+.1):PRINTCHR$(G+64);:NEXT
```

David's program inputs the word, then displays the encoded value and then decodes it. Matt's program lets the user specify whether the input word is to be encoded or decoded.

These two solutions are very similar. Matt defines a MOD 29 function in line 10. David uses the INT function in lines 210 and 240 to calculate MOD 29. The added .1 in David's program is a standard technique to guarantee proper truncation by the INT function. The .1 may be needed for certain values of G when the division by 29 cannot be performed precisely enough. (For you super programmers, when is the .1 actually needed? For what values of G in line 240 would the results be different without it?)

Problem #32-2: Text Search from Len Lindsay (Madison, WI) was easily solved with COMAL's IN function, as this program from Bill Davies (Downingtown, PA) shows.

```
0001 //  COMMODARES PROBLEM #32-2
0002 //     TEXT SEARCH
0003 //  COMAL SOLUTION BY
0004 //     BILL DAVIES
0010 DIM SEARCH'STRING$ OF 30
0011 DIM FILE'STRING$ OF 30
0012 INPUT "ENTER SEARCH STRING: ": SEAR
CH'STRING$
0013 OPEN FILE 3,"TESTFILE",READ
0014 REPEAT
0015  INPUT FILE 3: FILE'STRING$
0016  IF SEARCH'STRING$ IN FILE'STRING$
THEN
0017   PRINT FILE'STRING$
0018  ENDIF
0019 UNTIL EOF(3)
0020 CLOSE FILE 3
```

The IN function is true if the search string is found in the target string. The EOF function in line 19 is false until the end-of-file marker is found in the search file.

A solution for the C-64 from Jason Simpson (Everett, WA) is listed below:

```
·1 REM COMMODARES PROBLEM #32-2:
·2 REM      TEXT SEARCH
·3 REM SOLUTION BY
·4 REM    JASON SIMPSON
·5 REM
·10 INPUT"SEARCH STRING";A$
·20 A=LEN(A$)
·30 OPEN1,8,0,"TESTFILE,S,R"
·40 IF ST THEN CLOSE1 : END
·50 INPUT#1,B$
·60 B=LEN(B$)
·70 FOR Y=1 TO B
·80 IF MID$(B$,Y,A)=A$ THEN PRINT B$: Y=B
·90 NEXT Y
·100 GOTO 40
```

This program opens the sequential file "TESTFILE". If the file is not found, ST will be unequal to zero and the program ends in line 40. Otherwise each string (up to the next carriage return) is read into the variable B$. Lines 70 through 90 step through B$ looking for a substring equal to A$. If a match is found, the loop variable Y is set equal to the end value B so that the FOR/NEXT loop will be terminated in line 90. This is neater than using a GOTO to branch past the NEXT Y statement.

Kurt Schaeffer (Lebanon, PA) used a FOR/NEXT loop like this:

```
FOR Y=1 TO LT-LS+1
IF MID$(target$,Y,LS)=search$ THEN CT=CT
+1
NEXT Y
```

where LT and LS are the lengths of the target (file) string and search string respectively. CT counts the number of times that search$ is found in target$.

The following solution from Louis Dix (North Hampton, NH) takes advantage of several functions in BASIC 7.0 for the C-128:

```
·1 REM COMMODARES PROBLEM #32-2:
·2 REM       TEXT SEARCH
·3 REM C-128  SOLUTION BY
·4 REM     LOUIS DIX
·5 REM
·10 INPUT "ENTER SEARCH SUBSTRING ";S$
·20 DOPEN#3,"TESTFILE"
·30 DO
·40 INPUT#3,L$
·50 IF INSTR(L$,S$) <> 0 THEN PRINT L$
·60 LOOP UNTIL ST
·70 DCLOSE
```

The INSTR function gives a value corresponding to the starting position of the first occurrence of substring S$ within L$. If S$ is not in L$, then INSTR returns a zero. The LOOP UNTIL ST line causes the program to branch back to line 40 if the STatus variable is not zero. ST is zero until the end of the file is found.

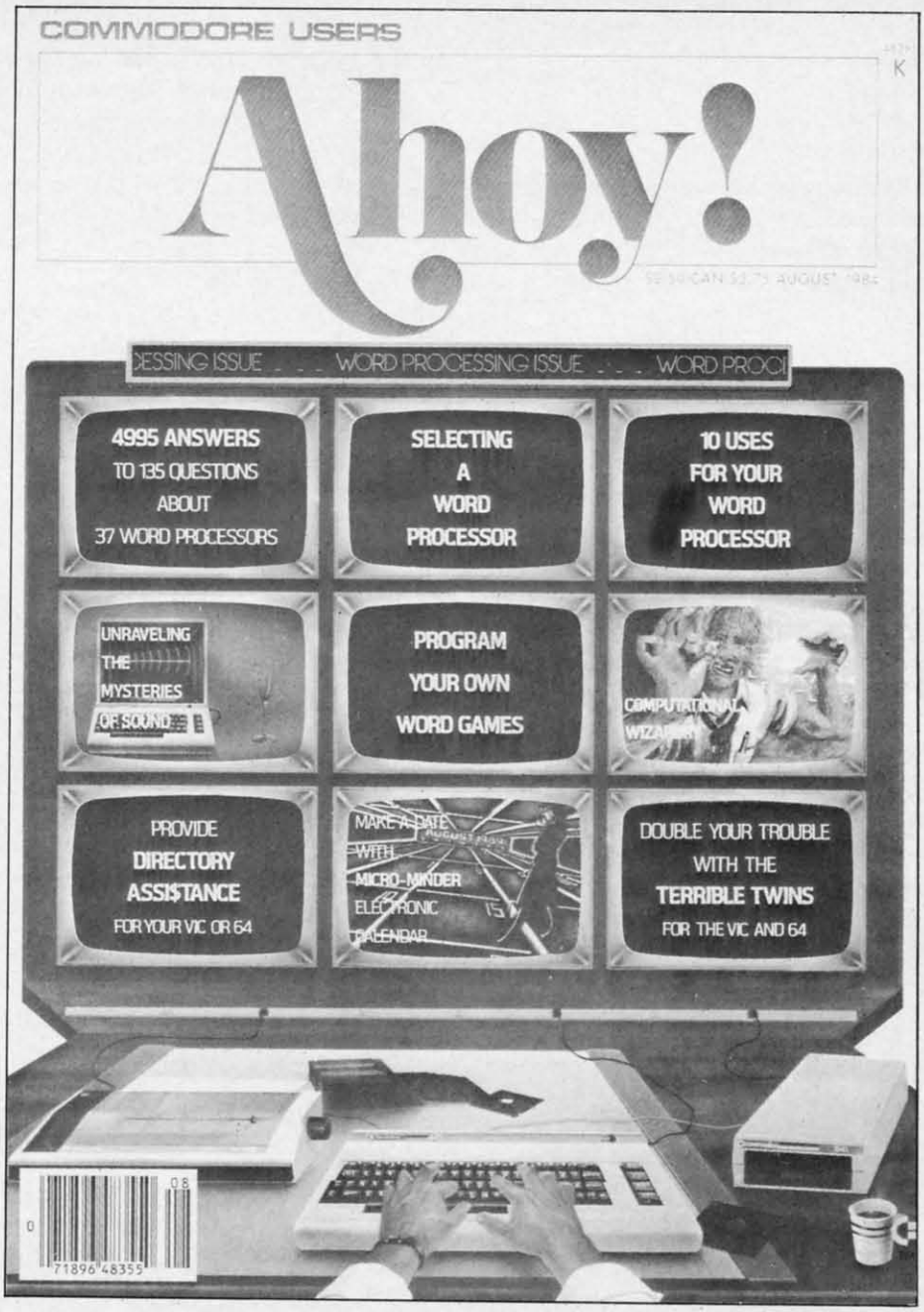
Here is the program to create the sequential file called TESTFILE on the disk. It must be run once before the three programs above are used.

```
·1 REM COMMODARES PROBLEM #32-2:
·2 REM       TEXT SEARCH
·3 REM PROGRAM TO CREATE SEQ FILE
·4 REM       "TESTFILE"
·5 REM
·10 OPEN 8,8,8,"TESTFILE,S,W"
·20 FOR N=1 TO 4
·30 READ L$ : PRINT#8,L$
·40 NEXT N : CLOSE 8 : END
·50 DATA FIRST LINE, SECOND LINE
·60 DATA THIRD LINE, FOURTH LINE
```

Of course you may use these TEXT SEARCH solutions to look through any other sequential files for specified substrings. Change the input filename appropriately. You must modify these programs to use the GET$ function instead of the INPUT# function if you want to search through program files.

Good solutions for *Problem #32-3: Round Robin* came from Douglas Underwood (Walla Walla, WA), Scott Preston (Shillington, PA) in COMAL, and this one from Fred Ransom (Oxnard, CA).

```
·1 REM COMMODARES PROBLEM #32-3:
·2 REM      ROUND ROBIN
·3 REM SOLUTION BY
·4 REM      FRED RANSOM
·5 REM
·10 INPUT "NUMBER OF TEAMS";N
·20 DIM T$(N)
·30 FOR K=0 TO N-1
·40 PRINT"TEAM NUMBER ";K+1;: INPUT T$(K)
·50 NEXT K
·55 IF (N AND 1) THEN N=N+1:T$(N-1)="[3"-
"]"
·60 INPUT "TO (P)RINTER OR (S)CREEN [S]";
FL$
·70 DEV=3 : IF FL$="P" THEN DEV=4
·80 OPEN 4,DEV
·90 FOR J=1 TO N-1
·100 FOR A=1 TO 3 : PRINT#4 : NEXT
·110 PRINT#4,"ROUND ",J
·120 PRINT#4,"HOME", "AWAY"
·130 PRINT#4,"[6"="]", "[6"="]"
·140 FOR K=0 TO (N/2-1)
·150 IF (J AND 1)=1 THEN PRINT#4, T$(K),
T$(K+N/2)
·160 IF (J AND 1)=0 THEN PRINT#4, T$(K +
N/2), T$(K)
·170 NEXT K
·180 A$=T$(1)
·190 IF N < 3 THEN 240
·200 FOR K=2 TO N-1
·210 T$(K-1)=T$(K)
·220 NEXT K
```



HAS EVERYTHING!

SUBSCRIBE TO AHOY!

☐ Twelve Issues for $21.95 ($29.25 Canada and elsewhere)
☐ Twenty-four Issues for $41.75 ($54.25 Canada and elsewhere)

Name_____

Address_____

City_____ State_____ Zip_____

Send coupon or facsimile to:
ION INTERNATIONAL INC.
45 West 34th Street, Room 407, New York, NY 10001

```
•230 T$(N-1)=A$
•240 NEXT J
•250 CLOSE 4
```

The algorithm for this program is to hold one team constant and rotate the others. In order to get a more even distribution of home and away games, "Home" and "Away" are reversed from round to round. For example, if round one is

Home:	1	2	3	4
Away:	8	7	6	5

then round two will be

Away:	1	8	2	3
Home:	7	6	5	4

Team 1 remains fixed and the other teams rotate in a clockwise fashion.

In Fred's program you specify the number of teams and then their names. You may then select the screen or the printer for the output. If there is an odd number of teams, a no-game symbol (---) is added to the roster. The determination of oddness or evenness of N is determined by the (N AND 1) expression. Its value is zero if N is even and one if N is odd. This program will work for two or more teams, limited only by the computer's memory.

Readers had fun with *Problem #32-4: Centipede Scroller.* The problem was to make the word "centipede" to centipede down the screen. This solution from Barry King (Nome, AK) is representative of several others.

```
•1 REM COMMODARES PROBLEM #32-4:
•2 REM     CENTIPEDE SCROLLER
•3 REM SOLUTION BY
•4 REM     BARRY KING
•5 PRINT"[CLEAR][DOWN][DOWN]  HERE IT COM
 ES.
•10 DIM A(959),L(10):S=1025
•20 FOR R=0 TO 24 STEP 2:FOR C=0 TO 37:A(
 R*38+C)=S+R*40+C:NEXT:NEXT
•30 FOR R=1 TO 23 STEP 2:FOR C=0 TO 37:A(
 R*38+C)=S+R*40+(37-C):NEXT:NEXT
•40 FOR P=950 TO 959:A(P)=P+75:NEXT
•50 W$=" CENTIPEDE":FOR N=1 TO 10:L(N)=AB
 S(ASC(MID$(W$,N,1))-64):NEXT:PRINT"[CLEA
 R]"
•60 FOR P=0 TO 949:FOR N=1 TO 10:POKE A(P
 +N),L(N):NEXT:NEXT:GOTO 60
```

This program works on the C-64 and on the C-128 (40 column screen) since both computers have screen memory from address 1024 to address 2023. Lines 20 through 40 fill the array A( ) with the sequential addresses in which to POKE the centipede's characters. It takes a while for this array to be created, so be patient. Line 50 fills the character array L( ) with the values to be POKEd into screen memory. The main loop of the program is simply line 60 which is repeated indefinitely.

Jim Speers (Niles, MI) suggested a graphics centipede which you could add to this program by replacing line 50 with these:

```
50 FOR N=1 TO 10 : READ L(N) : NEXT
55 DATA 32,67,114,114,114,114,114,114,87
,61
```

Jim also suggested replacing the 32 in line 55 with a 46 if you prefer the centipede to leave a trail. The use of an array to store the sequential POKE locations allows you to modify the path of the centipede by changing the FOR/NEXT loops which define A( ).

Douglas Underwood (Walla Walla, WA) sent this machine language routine to POKE 14's into color RAM so that POKEd characters appear as light blue. He said this is necessary for C-64's with Revision 2 ROM's. If your computer displays only invisible centipedes, you may need to add this routine to the program above.

```
•1 REM - FILL COLOR RAM WITH CODE FOR
•2 REM - LIGHT BLUE.  FOR C-64'S WITH
•3 REM - REVISION 2 ROMS ONLY.
•4 REM - ADD TO CENTIPEDE SCROLLER PRGM
•5 REM       DOUGLAS UNDERWOOD
•6 GOSUB 100 : SYS 828
•100 FOR AD=828 TO 856:READ MC:POKE AD,MC
 :NEXT:RETURN
•110 DATA 169,0,133,251,169,219,133,252
•120 DATA 160,0,136,169,14,145,251,192
•130 DATA 0,208,247,166,252,202,134,252
•140 DATA 224,215,208,238,96
```

Note that you must change line 55 above to line 155 if you add this machine language routine to your program.

Congratulations this month to the following readers for their solutions, suggestions, and letters:

J. Callaway (Orange Beach, AL)
William Colman (Hamden, CT)
Bucky Cox (Weeki Wachee, FL)
Ken Critton (New Haven, IN)
Bill Davies (Downington, PA)
Louis Dix (North Hampton, NH)
Craig Ewert (Crystal Lake, IL)
Thomson Fung (San Diego, CA)
John Gilmore (Sacramento, CA)
Charles Grady (Cleveland, TN)
David Hoffner (Brooklyn, NY)
Sol Katz (Lakewood, CO)
Barry King (Nome, AK)
Paul Lalli (McAlester, OK)
Yee Chang Lee (Yonkers, NY)
Wallace Leeker (Lemay, MO)
Dennis McGrath (Cicero, IL)
M. Naylor (N. Miami Beach, FL)
Fred Ransom (Oxnard, CA)
Kurt Schaeffer (Lebanon, PA)
Matt Shapiro (Fort Lee, NJ)
Jason Simpson (Everett, WA)
Paul Sobolik (Pittsburgh, PA)
Jim Speers (Niles, MI)
Steven Steckler (Columbia, MD)
D. Underwood (Walla Walla, WA)

Thanks also to our international writers this month: Mark Breault (Brandon, Manitoba) and Peter Zinterhof (Salzburg, Austria). We look forward to letters from programmers at all levels. If you have solved one or more of the *Commodares*, send us your solutions. Your ideas just might be worthy of publication for the world to see. □

FLOTSAM

Up until now I have not subscribed to *Ahoy!* because I am not all that interested in simple games (not always easy). I am more interested in programming aids and utilities that I can learn from or use in a program.

I want you to know that the money is in the mail. The program VERY-ABLE in the March 1986 issue convinced me that I don't want to do without your magazine. The only thing that program was missing was a choice between the screen and a printer. The program will print to a printer if you use the line below in the direct mode:

```
OPEN 4,4:CMD4:SYS 49152
```

Remember to (PRINT#4:CLOSE4) after the program is finished to close the channel. –William H. Duncan
Middletown, OH

I read with shock and regret that you are looking forward to receiving (and displaying) Amiga pictures in *Ahoy!'s Art Gallery.* To hell with the Amiga! I don't own one, can't afford one, and couldn't care less about seeing the pictures. I subscribe to *Ahoy!* because your fine mag supports my C-128. I understand that C-64/C-128 graphics don't hold a candle to the Amiga's, and most likely *Art Gallery* contents would be completely replaced with Amiga garbage. It is my understanding that nearly one half million C-128's have been sold, and less than 10% of that amount of Amigas are out there. It's fun to see the "best of the best" C-64/C-128 graphics – it's inspirational and sets a goal for us to try to aspire to (knowing our machines can do that). Amiga pictures would be of little interest, perhaps merely frustrating for us "commoners."

Don't get me wrong – I like the Amiga, and I'm very impressed with its capabilities. But with accessories, the Amiga package costs over $2000 – out of the reach of many of us. Please reconsider this decision to include Amiga graphics at the expense of C-64/C-128 pictures. Idea: Perhaps a separate Amiga *Art Gallery* would be feasible, eh?

Other than that, keep up your state-of-the-art gallery, articles, and reviews. –Franz Stephan
Oshkosh, WI

Now that the price of a full-blown Amiga system has dipped to barely above $1000, Franz, we're sure you've jumped on the bandwagon. A separate Amiga Art Gallery *is a possibility for the future, but rest assured that in any event, the C-64 and C-128 artists in our readership will never be nosed out of the feature they made famous.*

...Announcing

THE COMMODORE USERS

Ahoy Access Club!

FREE TO ALL SUBSCRIBERS

An incredible offer from the publishers of *Ahoy!*...The ***Ahoy!* Access Club.** The ***Ahoy!* Access Club** offers members: discounts, rebates, coupons on products and services for your Commodore, a Commodore calendar of events, and much, much more! Members will also receive "The Clipper," an exclusive bi-monthly news bulletin. But the best thing about the ***Ahoy!* Access Club** is that it's **FREE!** to all *Ahoy!* subscribers. Subscribe today! and along with your first issue of *Ahoy!* you'll receive your official ***Ahoy!* Access Club** membership card...your ticket to super savings and bonuses you can only get from *Ahoy!* Don't miss out on this fantastic offer. The ***Ahoy!* Access Club**...if you own a Commodore — this club is for you!

THE COMMODORE MAGAZINE THAT OFFERS YOU MORE

PROGRAM LISTINGS

Attention new *Ahoy!* readers! You must read the following information very carefully prior to typing in programs listed in *Ahoy!* Certain Commodore characters, commands, and strings of characters and commands will appear in a special format. Follow the instructions and listing guide on this page.

On the following pages you'll find several programs that you can enter on your Commodore computer. But before doing so, read this entire page carefully.

To insure clear reproductions, *Ahoy!*'s program listings are generated on a daisy wheel printer, incapable of printing the commands and graphic characters used in Commodore programs. These are therefore represented by various codes enclosed in brackets [ ]. For example: the SHIFT CLR/HOME command is represented onscreen by a heart ♥ The code we use in our listings is [CLEAR]. The chart below lists all such codes which you'll encounter in our listings, except for one other special case.

The other special case is the COMMODORE and SHIFT characters. On the front of most keys are two symbols. The symbol on the left is obtained by pressing that key while holding down the COMMODORE key; the symbol on the right, by pressing that key while holding down the SHIFT key. COMMODORE and SHIFT characters are represented in our listings by a lower-case "s" or "c" followed by the symbol of the key you must hit. COMMODORE J, for example, is represented by [c J], and SHIFT J by [s J].

Additionally, any character that occurs more than two times in a row will be displayed by a coded listing. For example, [3 "[LEFT]"] would be 3 CuRSoR left commands in a row, [5 "[s EP]"] would be 5 SHIFTed English Pounds, and so on. Multiple blank spaces will be noted in similar fashion: e.g., 22 spaces as [22 " "].

Sometimes you'll find a program line that's too long for the computer to accept (C-64 lines are a maximum of 80 characters, or 2 screen lines long; VIC 20 lines, a maximum of 88 characters, or 4 screen lines). To enter these lines, refer to the *BASIC Command Abbreviations Appendix* in your User Manual.

On the next page you'll find our *Bug Repellent* programs for the C-128 and C-64. The version appropriate for your machine will help you proofread our programs after you type them. (Please note: the *Bug Repellent* line codes that follow each program line, in the whited-out area, should *not* be typed in. See the instructions preceding each program.)

Also on the following page you will find *Flankspeed*, our ML entry program, and instructions on its use. □

Call *Ahoy!* at 212-239-0855 with any problems.

When You See	It Means	You Type		You Will See
[CLEAR]	Screen Clear	SHIFT	CLR/HOME	♥
[HOME]	Home		CLR/HOME	
[UP]	Cursor Up	SHIFT	↑ CRSR ↑	
[DOWN]	Cursor Down		↑ CRSR ↑	
[LEFT]	Cursor Left	SHIFT	←CRSR←	
[RIGHT]	Cursor Right		←CRSR←	
[SS]	Shifted Space	SHIFT	Space	
[INSERT]	Insert	SHIFT	INST/DEL	
[DEL]	Delete		INST/DEL	
[RVSON]	Reverse On	CNTRL	9	
[RVSOFF]	Reverse Off	CNTRL	0	
[UPARROW]	Up Arrow		↑	
[BACKARROW]	Back Arrow		←	
[PI]	PI		π	
[EP]	English Pound		£	

When You See	It Means	You Type		You Will See
[BLACK]	Black	CNTRL	1	
[WHITE]	White	CNTRL	2	
[RED]	Red	CNTRL	3	
[CYAN]	Cyan	CNTRL	4	
[PURPLE]	Purple	CNTRL	5	
[GREEN]	Green	CNTRL	6	
[BLUE]	Blue	CNTRL	7	
[YELLOW]	Yellow	CNTRL	8	
[F1]	Function 1		F1	
[F2]	Function 2	SHIFT	F1	
[F3]	Function 3		F3	
[F4]	Function 4	SHIFT	F3	
[F5]	Function 5		F5	
[F6]	Function 6	SHIFT	F5	
[F7]	Function 7		F7	
[F8]	Function 8	SHIFT	F7	

BUG REPELLENT By MICHAEL KLEINERT and DAVID BARRON

Bug Repellent is a checksum program used for proofreading BASIC listings typed in from *Ahoy!* magazine. For each program line you enter, *Bug Repellent* will produce a two-letter code that should match the code listed beside that line in the magazine.

Type in, save, and run the *Bug Repellent*. (If you have a C-64, type in the C-64 version. If you have a C-128, you will need to type in the C-64 version for use with C-64 programs, and the C-128 version for use with C-128 programs.) If you have typed in *Bug Repellent* properly, you will get the message BUG REPELLENT INSTALLED; otherwise you will get an error message. If you get an error message, double check the *Bug Repellent* program for typing mistakes. Type NEW and hit RETURN. Then type in and save, or load, the *Ahoy!* program you wish to check. Type in SYS 49152 for the C-64 version or SYS 3072 for the C-128 version and hit RETURN (this will begin execution of *Bug Repellent*). You will see the prompt SCREEN OR PRINTER ? Hit S if you want the codes listed on the screen, or P if you want them listed on the printer. To pause the listing depress and hold the SHIFT key.

Compare the codes your machine generates to those listed to the right of the corresponding program lines. If you spot a difference, that line contains an error. Write down the numbers of the lines where the contradictions occur. LIST each line, locate the errors, and correct them.

COMMODORE 64 VERSION

```
·100 FOR X = 49152 TO 49488:READY:S=S+Y          AB
·110 IF Y<0 OR Y>255 THEN 130                    EA
·120 POKE X,Y:NEXT:GOTO140                       ID
·130 PRINT"[CLEAR][DOWN]**ERROR**":PRINT"[DOWN
 ]PLEASE CHECK LINE"PEEK(64)*256+PEEK(63):END    ID
·140 IF S<>44677 THEN PRINT"[CLEAR][DOWN]**ERR
 OR**":PRINT"[DOWN]PLEASE CHECK DATA LINES 170
 -500":END                                       HJ
·150 PRINT"[CLEAR]":POKE53280,0:POKE53281,6:PO
 KE646,1                                         NP
·160 PRINT"[RVSON][6" "]C-64 BUG REPELLENT INS
 TALLED[6" "]"                                   LF
·170 DATA32,161,192,165,43,133,251,165,44,133    DL
·180 DATA252,160,0,132,254,32,37,193,234,177     DB
·190 DATA251,208,3,76,138,192,230,251,208,2      OF
·200 DATA230,252,76,43,192,76,73,78,69,32        KN
·210 DATA35,32,0,169,35,160,192,32,30,171        CA
·220 DATA160,0,177,251,170,230,251,208,2,230     CE
·230 DATA252,177,251,32,205,189,169,58,32,210    JE
·240 DATA255,169,0,133,253,230,254,32,37,193     CL
·250 DATA234,165,253,160,0,76,13,193,133,253     NB
·260 DATA177,251,208,237,165,253,41,240,74,74    MB
·270 DATA74,74,24,105,65,32,210,255,165,253      EP
·280 DATA 41,15,24,105,65,32,210,255,169,13      GH
·290 DATA32,220,192,230,63,208,2,230,64,230      AN
·300 DATA251,208,2,230,252,76,11,192,169,153     NG
·310 DATA160,192,32,30,171,166,63,165,64,76      BF
·320 DATA231,192,96,76,73,78,69,83,58,32         EP
·330 DATA0,169,247,160,192,32,30,171,169,3       PJ
·340 DATA133,254,32,228,255,201,83,240,6,201     FK
·350 DATA80,208,245,230,254,32,210,255,169,4     FL
·360 DATA166,254,160,255,32,186,255,169,0,133    CL
·370 DATA63,133,64,133,2,32,189,255,32,192       GC
·380 DATA255,166,254,32,201,255,76,73,193,96     NN
·390 DATA32,210,255,173,141,2,41,1,208,249       NH
·400 DATA96,32,205,189,169,13,32,210,255,32      IM
·410 DATA204,255,169,4,76,195,255,147,83,67      KC
·420 DATA82,69,69,78,32,79,82,32,80,82           DC
·430 DATA 73,78,84,69,82,32,63,32,0,76           ML
·440 DATA44,193,234,177,251,201,32,240,6,138     GN
·450 DATA113,251,69,254,170,138,76,88,192,0      JK
·460 DATA0,0,0,230,251,208,2,230,252,96          NA
·470 DATA170,177,251,201,34,208,6,165,2,73       DM
·480 DATA255,133,2,165,2,208,218,177,251,201     JA
·490 DATA32,208,212,198,254,76,29,193,0,169      FM
·500 DATA13,76,210,255,0,0,0                     PA
```

COMMODORE 128 VERSION

```
·100 FAST:FOR X = 3072 TO 3520:READ Y:POKE X,Y
 :S=S+Y:TRAP110:NEXT:SLOW                        IH
·110 SLOW:IF S<>49057 THEN PRINT"[CLEAR][DOWN]
**ERROR**":PRINT"[DOWN]PLEASE CHECK DATA LINE
S 140-390":END                                   JA
·120 PRINT"[CLEAR][DOWN]  C-128 BUG REPELLENT
 INSTALLED"                                      II
·130 PRINT"[4" "]TYPE SYS 3072 TO ACTIVATE"      IN
·140 DATA 32,161,12,165,45,133,251,165,46,133,
 252,160,0,132,254,32,37                         OF
·150 DATA 13,234,177,251,208,3,76,138,12,230,2
 51,208,2,230,252,76,43                          NC
·160 DATA 12,76,73,78,69,32,35,32,0,169,35,160
 ,12,32,80,13,160,0,177                          OL
·170 DATA 251,170,230,251,208,2,230,252,177,25
 1,32,89,13,169,58,32,98                         EF
·180 DATA 13,169,0,133,253,230,254,32,37,13,23
 4,165,253,160,0,76,13                           JO
·190 DATA 13,133,253,177,251,208,237,165,253,4
 1,240,74,74,74,74,24                            LC
·200 DATA 105,65,32,98,13,165,253,41,15,24,105
 ,65,32,98,13,169,13,32                          DE
·210 DATA 220,12,230,65,208,2,230,66,230,251,2
 08,2,230,252,76,11,12                           GM
·220 DATA 169,153,160,12,32,80,13,166,65,165,6
 6,76,231,12,96,76,73,78                         CP
·230 DATA 69,83,58,32,0,169,247,160,12,32,80,1
 3,169,3,133,254,32,107                          HC
·240 DATA 13,201,83,240,6,201,80,208,245,230,2
 54,32,98,13,169,4,166                           GK
·250 DATA 254,160,255,32,116,13,169,0,133,65,1
 33,66,133,250,32,125,13                         LB
·260 DATA 32,134,13,166,254,32,143,13,76,73,13
 ,96,32,98,13,165,211                            JF
·270 DATA 234,41,1,208,249,96,32,89,13,169,13,
 32,98,13,32,152,13,169,4                        GD
·280 DATA 76,161,13,147,83,67,82,69,69,78,32,7
 9,82,32,80,82,73,78,84,69                       PL
·290 DATA 82,32,63,32,0,76,44,13,234,177,251,2
 01,32,240,6,138,113,251,69                      OK
·300 DATA 254,170,138,76,88,12,0,0,0,0,230,251
 ,208,2,230,252,96,170,177                       FJ
·310 DATA 251,201,34,208,6,165,250,73,255,133,
 250,165,250,208,218,177                         GA
·320 DATA 251,201,32,208,212,198,254,76,29,13,
 0,169,13,76,98,13,0,0,32                        FI
·330 DATA 170,13,32,226,85,76,180,13,32,170,13
 ,32,50,142,76,180,13,32                         OF
·340 DATA 170,13,32,210,255,76,180,13,32,170,1
 3,32,228,255,76,180,13,32                       AK
·350 DATA 170,13,32,186,255,76,180,13,32,170,1
 3,32,189,255                                    BP
·360 DATA 76,180,13,32,170,13,32,192,255,76,18
 0,13,32,170,13                                  FP
·370 DATA 32,201,255,76,180,13,32,170,13,32,20
 4,255,76,180,13,32,170                          ID
·380 DATA 13,32,195,255,76,180,13,133,67,169,0
 ,141,0,255,165,67,96                            BJ
·390 DATA 133,67,169,0,141,1,255,165,67,96,0,0
 ,0                                              IF
```

FLANKSPEED FOR THE C-64 By GORDON F. WHEAT

Flankspeed will allow you to enter machine language *Ahoy!* programs without any mistakes. Once you have typed the program in, save it for future use. While entering an ML program with *Flankspeed* there is no need to enter spaces or hit the carriage return. This is all done automatically. If you make an error in a line a bell will ring and you will be asked to enter it again. To LOAD in a program Saved with *Flankspeed* use LOAD "name",1,1 for tape, or LOAD "name",8,1 for disk. The function keys may be used after the starting and ending addresses have been entered.

f1 – SAVEs what you have entered so far.

f3 – LOADs in a program worked on previously.

f5 – To continue on a line you stopped on after LOADing in the previous saved work.

f7 – Scans through the program to locate a particular line, or to find out where you stopped the last time you entered the program.

f7 temporarily freezes the output as well.

```
·100 POKE53280,12:POKE53281,11
·105 PRINT"[CLEAR][c 8][RVSON][15" "]FLANKSPEED[15" "]";
·110 PRINT"[RVSON][5" "]MISTAKEPROOF ML ENTRY PROGRAM[6" "
]"
·115 PRINT"[RVSON][9" "]CREATED BY G. F. WHEAT[9" "]"
·120 PRINT"[RVSON][3" "]COPR. 1984, ION INTERNATIONAL INC.
[3" "]"
·125 FORA=54272TO54296:POKEA,0:NEXT
·130 POKE54272,4:POKE54273,48:POKE54277,0:POKE54278,249:PO
KE54296,15
·135 FORA=680TO0699:READB:POKEA,B:NEXT
·140 DATA169,251,166,253,164,254,32,216,255,96
·145 DATA169,0,166,251,164,252,32,213,255,96
·150 B$="STARTING ADDRESS IN HEX":GOSUB430:AD=B:SR=B
·155 GOSUB480:IFB=0THEN150
·160 POKE251,T(4)+T(3)*16:POKE252,T(2)+T(1)*16
·165 B$="ENDING ADDRESS IN HEX":GOSUB430:EN=B
·170 GOSUB470:IFB=0THEN150
·175 POKE254,T(2)+T(1)*16:B=T(4)+1+T(3)*16
·180 IFB>255THENB=B-255:POKE254,PEEK(254)+1
·185 POKE253,B:PRINT
·190 REM  GET HEX LINE
·195 GOSUB495:PRINT": [c P][LEFT]";:FORA=0TO8
·200 FORB=0TO1:GOTO250
·205 NEXTB
·210 A%(A)=T(1)+T(0)*16:IFAD+A-1=ENTHEN340
·215 PRINT" [c P][LEFT]";
·220 NEXTA:T=AD-(INT(AD/256)*256):PRINT" "
·225 FORA=0TO7:T=T+A%(A):IFT>255THENT=T-255
·230 NEXT
·235 IFA%(8)<>TTHENGOSUB375:GOTO195
·240 FORA=0TO7:POKEAD+A,A%(A):NEXT:AD=AD+8:GOTO195
·245 REM    GET HEX INPUT
·250 GETA$:IFA$=""THEN250
·255 IFA$=CHR$(20)THEN305
·260 IFA$=CHR$(133)THEN535
·265 IFA$=CHR$(134)THEN560
·270 IFA$=CHR$(135)THENPRINT" ":GOTO620
·275 IFA$=CHR$(136)THENPRINT" ":GOTO635
·280 IFA$>"@"ANDA$<"G"THENT(B)=ASC(A$)-55:GOTO295
·285 IFA$>"/"ANDA$<":"THENT(B)=ASC(A$)-48:GOTO295
·290 GOSUB415:GOTO250
·295 PRINTA$"[c P][LEFT]";
·300 GOTO205
·305 IFA>0THEN320
·310 A=-1:IFB=1THEN330
·315 GOTO220
·320 IFB=0THENPRINTCHR$(20);CHR$(20);:A=A-1
·325 A=A-1
·330 PRINTCHR$(20);:GOTO220
·335 REM LAST LINE
·340 PRINT" ":T=AD-(INT(AD/256)*256)
·345 FORB=0TOA-1:T=T+A%(B):IFT>255THENT=T-255
·350 NEXT
·355 IFA%(A)<>TTHENGOSUB375:GOTO195
·360 FORB=0TOA-1:POKEAD+B,A%(B):NEXT
·365 PRINT:PRINT"YOU ARE FINISHED!":GOTO535
·370 REM BELL AND ERROR MESSAGES
·375 PRINT:PRINT"LINE ENTERED INCORRECTLY":PRINT:GOTO415
·380 PRINT:PRINT"INPUT A 4 DIGIT HEX VALUE!":GOTO415
·385 PRINT:PRINT"ENDING IS LESS THAN STARTING!":B=0:GOTO41
5
·390 PRINT:PRINT"ADDRESS NOT WITHIN SPECIFIED RANGE!":B=0:
GOTO415
·395 PRINT:PRINT"NOT ZERO PAGE OR ROM!":B=0:GOTO415
·400 PRINT"?ERROR IN SAVE":GOTO415
·405 PRINT"?ERROR IN LOAD":GOTO415
·410 PRINT:PRINT:PRINT"END OF ML AREA":PRINT
·415 POKE54276,17:POKE54276,16:RETURN
·420 OPEN15,8,15:INPUT#15,A,A$:CLOSE15:PRINTA$:RETURN
·425 REM GET FOUR DIGIT HEX
·430 PRINT:PRINTB$;:INPUTT$
·435 IFLEN(T$)<>4THENGOSUB380:GOTO430
·440 FORA=1TO4:A$=MID$(T$,A,1):GOSUB450:IFT(A)=16THENGOSUB
380:GOTO430
·445 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)*16)+T(4):RETURN
·450 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-55:RETURN
·455 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-48:RETURN
·460 T(A)=16:RETURN
·465 REM ADRESS CHECK
·470 IFAD>ENTHEN385
·475 IFB<SRORB>ENTHEN390
·480 IFB<256OR(B>4096ANDB<49152)ORB>53247THEN395
·485 RETURN
·490 REM ADDRESS TO HEX
·495 AC=AD:A=4096:GOSUB520
·500 A=256:GOSUB520
·505 A=16:GOSUB520
·510 A=1:GOSUB520
·515 RETURN
·520 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):GOTO530
·525 A$=CHR$(T+48)
·530 PRINTA$;:AC=AC-A*T:RETURN
·535 A$="**SAVE**":GOSUB585
·540 OPEN1,T,1,A$:SYS680:CLOSE1
·545 IFST=0THENEND
·550 GOSUB400:IFT=8THENGOSUB420
·555 GOTO535
·560 A$="**LOAD**":GOSUB585
·565 OPEN1,T,0,A$:SYS690:CLOSE1
·570 IFST=64THEN195
·575 GOSUB405:IFT=8THENGOSUB420
·580 GOTO560
·585 PRINT" ":PRINTTAB(14)A$
·590 PRINT:A$="":INPUT"FILENAME";A$
·595 IFA$=""THEN590
·600 PRINT:PRINT"TAPE OR DISK?":PRINT
·605 GETB$:T=1:IFB$="D"THENT=8:A$="@0:"+A$:RETURN
·610 IFB$<>"T"THEN605
·615 RETURN
·620 B$="CONTINUE FROM ADDRESS":GOSUB430:AD=B
·625 GOSUB475:IFB=0THEN620
·630 PRINT:GOTO195
·635 B$="BEGIN SCAN AT ADDRESS":GOSUB430:AD=B
·640 GOSUB475:IFB=0THEN635
·645 PRINT:GOTO670
·650 FORB=0TO7:AC=PEEK(AD+B):GOSUB505:IFAD+B=ENTHENAD=SR:G
OSUB410:GOTO195
·655 PRINT" ";:NEXTB
·660 PRINT:AD=AD+8
·665 GETB$:IFB$=CHR$(136)THEN195
·670 GOSUB495:PRINT": ";:GOTO650
```

THE SHADOW KNOWS, PART II

FROM PAGE 27

Assembler required for program entry! See introductory article.

SPRITE2

```
1 *
2 * SPRITE2
3 *
4           ORG   $1300
5 *
6 COLOR     EQU   $E0
7 *
8 TABLEN    EQU   $800
9 MAPLEN    EQU   1000
10 SCRLEN   EQU   8000
11 SP0ADR   EQU   $4E00
12 COLMAP   EQU   $5C00
13 NEWADR   EQU   $4000
14 *
15 SPRPTR   EQU   $5FF8
16 SPENA    EQU   $D015
17 SP0COL   EQU   $D027
18 SP0X     EQU   $11D6
19 SP0Y     EQU   $11D7
20 MSIGX    EQU   $11E6
21 YXPAND   EQU   $D017
22 XXPAND   EQU   $D01D
23 *
24 INDFET   EQU   $FF74
25 *
26 HMAX     EQU   320
27 VMID     EQU   100-8
28 *
29 R6510    EQU   $0001
30 BASE     EQU   $6000
31 CHRBAS   EQU   $D000
32 SCROLY   EQU   $D011
33 SVMCSB   EQU   $A2D
34 BORDER   EQU   $D020
35 CIACRE   EQU   $DC0E
36 CI2PRA   EQU   $DD00
37 CIADIR   EQU   $DD02
38 *
39 TEMPA    EQU   $C8
40 TMPB     EQU   TEMPA+2
41 TPTR     EQU   TEMPA
42 *
43 MVSRCE   EQU   $FA
44 DEST     EQU   MVSRCE+2
45 BPTR     EQU   DEST+2
46 *
47 TABSIZ   EQU   $0C00
48 *
49 HPSN     EQU   TABSIZ+2
50 VPSN     EQU   HPSN+2
51 CHAR     EQU   VPSN+1
52 ROW      EQU   CHAR+1
53 LINE     EQU   ROW+1
54 BYTE     EQU   LINE+1
55 BITT     EQU   BYTE+2
56 *
57 MPRL     EQU   BITT+1
58 MPRH     EQU   MPRL+1
59 MPDL     EQU   MPRH+1
60 MPDH     EQU   MPDL+1
61 PRODL    EQU   MPDH+1
62 PRODH    EQU   PRODL+1
63 FILVAL   EQU   PRODH+1
64 LENPTR   EQU   FILVAL+1
65 CHCODE   EQU   LENPTR+2
66 HPTR     EQU   CHCODE+2
67 VPTR     EQU   HPTR+2
68 ONEBYT   EQU   VPTR+1
69 COUNT    EQU   ONEBYT+2
70 LTTR     EQU   COUNT+1
71 *
72          JMP   START
73 *
74 TEXT     DB    9,32,32,13
75          DB    25,32,3,15
76          DB    13,13,15,4
77          DB    15,18,5,32
78          DB    49,50,56,0
79 *
80 * BLOCK FILL ROUTINE
81 *
82 BLKFIL   LDA   FILVAL
83          LDX   TABSIZ+1
84          BEQ   PARTPG
85          LDY   #0
86 FULLPG   STA   (TPTR),Y
87          INY
88          BNE   FULLPG
89          INC   TPTR+1
90          DEX
91          BNE   FULLPG
92 PARTPG   LDX   TABSIZ
93          BEQ   FINI
94          LDY   #0
95 PARTLP   STA   (TPTR),Y
96          INY
97          DEX
98          BNE   PARTLP
99 FINI     RTS
100 *
101 * MULTIPLY ROUTINE
102 *
103 MULT16  LDA   #0
104         STA   PRODL
105         STA   PRODH
106         LDX   #17
107         CLC
108 MULT    ROR   PRODH
109         ROR   PRODL
110         ROR   MPRH
111         ROR   MPRL
112         BCC   CTDOWN
113         CLC
114         LDA   MPDL
115         ADC   PRODL
116         STA   PRODL
117         LDA   MPDH
118         ADC   PRODH
119         STA   PRODH
120 CTDOWN  DEX
121         BNE   MULT
122         RTS
123 *
124 * PLOT ROUTINE
125 *
126 * ROW=VPSN/8
127 *
128 PLOT     LDA   VPSN
129         LSR   A
130         LSR   A
131         LSR   A
132         STA   ROW
133 *
134 * CHAR=HPSN/8
135 *
136         LDA   HPSN
137         STA   TEMPA
138         LDA   HPSN+1
139         STA   TEMPA+1
140         LDX   #3
141 DLOOP    LSR   TEMPA+1
142         ROR   TEMPA
143         DEX
144         BNE   DLOOP
145         LDA   TEMPA
146         STA   CHAR
147 *
148 * LINE=VPSN AND 7
149 *
150         LDA   VPSN
151         AND   #7
152         STA   LINE
153 *
154 * BIT=7-(HPSN AND 7)
155 *
156         LDA   HPSN
157         AND   #7
158         STA   BITT
159         SEC
160         LDA   #7
161         SBC   BITT
162         STA   BITT
163 *
164 * CALCULATE BYTE
165 *
```

```
166 * MULTIPLY ROW * HMAX
167 *
168           LDA   ROW
169           STA   MPRL
170           LDA   #0
171           STA   MPRH
172           LDA   #<HMAX
173           STA   MPDL
174           LDA   #>HMAX
175           STA   MPDH
176           JSR   MULT16
177           LDA   MPRL
178           STA   TEMPA
179           LDA   MPRL+1
180           STA   TEMPA+1
181 *
182 * ADD PRODUCT TO BASE
183 *
184           CLC
185           LDA   #<BASE
186           ADC   TEMPA
187           STA   TEMPA
188           LDA   #>BASE
189           ADC   TEMPA+1
190           STA   TEMPA+1
191 *
192 * MULTIPLY 8 * CHAR
193 *
194           LDA   #8
195           STA   MPRL
196           LDA   #0
197           STA   MPRH
198           LDA   CHAR
199           STA   MPDL
200           LDA   #0
201           STA   MPDH
202           JSR   MULT16
203           LDA   MPRL
204           STA   TMPB
205           LDA   MPRH
206           STA   TMPB+1
207 *
208 * ADD LINE
209 *
210           CLC
211           LDA   TMPB
212           ADC   LINE
213           STA   TMPB
214           LDA   TMPB+1
215           ADC   #0
216           STA   TMPB+1
217 *
218 * TEMPA + TMPB = BYTE
219 *
220           CLC
221           LDA   TEMPA
222           ADC   TMPB
223           STA   TMPB
224           LDA   TEMPA+1
225           ADC   TMPB+1
226           STA   TMPB+1
227 *
228 * BYTE=(BYTE)OR2^BIT
229 *
230           LDX   BITT
231           INX
232           LDA   #0
233           SEC
234 SQUARE    ROL
235           DEX
236           BNE   SQUARE
237           LDY   #0
238           ORA   (TMPB),Y
239           STA   (TMPB),Y
240           RTS
241 *
242 * GET CHCODE'S ADDRESS
243 *
244 GETADR    LDA   #0
245           STA   CHCODE+1
246           LDA   CHCODE
247           CLC
248           ASL   A
249           ROL   CHCODE+1
250           ASL   A
251           ROL   CHCODE+1
252           ASL   A
253           ROL   CHCODE+1
254           STA   CHCODE
255 *
256           CLC
257           LDA   CHCODE
258           ADC   #<NEWADR
259           STA   BPTR
260           LDA   CHCODE+1
261           ADC   #>NEWADR
262           STA   BPTR+1
263           RTS
264 *
265 * DRAW A CHARACTER
266 *
267 DRAWCH    LDA   LTTR
268           STA   CHCODE
269           JSR   GETADR
270 *
271 * A NESTED LOOP:
272 *
273 * (X IS OUTSIDE LOOP)
274 *
275           LDX   #8
276 *
277 * COUNT 2 VERT LINES
278 *
279 SETLIN    LDA   #2
280           STA   COUNT
281 *
282 DRAWLN    LDY   #0
283           LDA   (BPTR),Y
284           STA   ONEBYT
285 *
286 * THE INSIDE LOOP:
287 *
288 * (Y IS ZERO AT START)
289 *
290 RSHIFT    LDA   ONEBYT
291           ASL   A
292           STA   ONEBYT
293           BCS   SHOW
294 *
295           INC   HPSN
296           BNE   ITSOK
297           INC   HPSN+1
298 ITSOK     JMP   NOSHOW
299 *
300 * DISPLAY BIT
301 *
302 * SAVE X, Y REGISTERS
303 *
304 SHOW      TXA
305           PHA
306           TYA
307           PHA
308 *
309           JSR   PLOT
310 *
311 * NOW DO IT AGAIN
312 *
313           INC   HPSN
314           BNE   NOINC
315           INC   HPSN+1
316 *
317 NOINC     JSR   PLOT
318 *
319 * RETRIEVE X, Y REGS
320 *
321           PLA
322           TAY
323           PLA
324           TAX
325 *
326 NOSHOW    INC   HPSN
327           BNE   LEAP
328           INC   HPSN+1
329 *
330 LEAP      INY
331           CPY   #8
332           BCC   RSHIFT
333 *
334           INC   VPSN
335 *
336           LDA   HPTR
337           STA   HPSN
338           LDA   HPTR+1
339           STA   HPSN+1
```

```
340 *
341 * 2 LINES DONE YET?
342 *
343           DEC   COUNT
344           BNE   DRAWLN
345 *
346           INC   BPTR
347           BNE   OKMSB
348           INC   BPTR+1
349 OKMSB     DEX
350           BNE   SETLIN
351           RTS
352 *
353 * COPY CHR SET INTO RAM
354 *
355 * PUT DATA IN NEW BLOCK
356 *
357 COPYCHRS LDA    #<CHRBAS
358           STA   MVSRCE
359           LDA   #>CHRBAS
360           STA   MVSRCE+1
361 *
362           LDA   #<NEWADR
363           STA   DEST
364           LDA   #>NEWADR
365           STA   DEST+1
366 *
367           LDA   #<TABLEN
368           STA   LENPTR
369           LDA   #>TABLEN
370           STA   LENPTR+1
371 *
372           LDY   #0
373           LDX   LENPTR+1
374           BEQ   MVPART
375 MVPAGE    JSR   GETDATA
376           INY
377           BNE   MVPAGE
378           INC   MVSRCE+1
379           INC   DEST+1
380           DEX
381           BNE   MVPAGE
382 MVPART    LDX   LENPTR
383           BEQ   MVEXIT
384 MVLAST    JSR   GETDATA
385           INY
386           DEX
387           BNE   MVLAST
388 MVEXIT    RTS
389 *
390 * MOVEDATA
391 *
392 GETDATA   PHA
393           TXA
394           PHA
395           LDA   #MVSRCE
396           LDX   #14
397           JSR   INDFET
398           STA   (DEST),Y
399           PLA
400           TAX
401           PLA
402           RTS
403 *
404 *
405 * MAIN ROUTINE
406 *
407 START     JSR   COPYCHRS
408 *
409           LDA   #$20
410           STA   $D8
411 *
412 * USE VIDEO BANK 1
413 *
414           LDA   #0
415           STA   $FF00
416           LDA   CI2PRA
417           AND   #$FC
418           ORA   #$02
419           STA   CI2PRA
420           STA   $FF01
421 *
422 * PUT SCR MAP AT $6000
423 * PUT CLR MAP AT $5C00
424 *
425           LDA   #$78
426           STA   SVMCSB
427 *
428 * CLEAR BIT MAP
429 *
430           LDA   #0
431           STA   FILVAL
432           LDA   #<BASE
433           STA   TPTR
434           LDA   #>BASE
435           STA   TPTR+1
436           LDA   #<SCRLEN
437           STA   TABSIZ
438           LDA   #>SCRLEN
439           STA   TABSIZ+1
440           JSR   BLKFIL
441 *
442 * SET COLORS
443 *
444           LDA   #COLOR
445           STA   FILVAL
446           LDA   #<COLMAP
447           STA   TPTR
448           LDA   #>COLMAP
449           STA   TPTR+1
450           LDA   #<MAPLEN
451           STA   TABSIZ
452           LDA   #>MAPLEN
453           STA   TABSIZ+1
454           JSR   BLKFIL
455           LDA   #13
456           STA   BORDER
457 *
458 * POSITION MESSAGE
459 *
460           LDA   #8
461           STA   HPSN
462           STA   HPTR
463           LDA   #0
464           STA   HPSN+1
465           STA   HPTR+1
466           LDA   #VMID
467           STA   VPSN
468           STA   VPTR
469 *
470 * PRINT LINE
471 *
472           LDX   #0
473 DISP      LDA   TEXT,X
474           CMP   #0
475           BEQ   DONE
476           STA   LTTR
477           TXA
478           PHA
479           JSR   DRAWCH
480           PLA
481           TAX
482 *
483 * ADVANCE CURSOR
484 *
485           CLC
486           LDA   HPTR
487           ADC   #16
488           STA   HPTR
489           STA   HPSN
490           LDA   HPTR+1
491           ADC   #0
492           STA   HPTR+1
493           STA   HPSN+1
494           LDA   VPTR
495           STA   VPSN
496 *
497 * PRINT NEXT LETTER
498 *
499           INX
500           JMP   DISP
501 *
502 DONE      NOP
503 *
504 * DISPLAY SPRITE #0
505 *
506 * DEFINE SPRITE
507 *
508 * CLEAR SPRITE MAP
509 *
510           LDA   #$00
511           STA   FILVAL
512           LDA   #<SP0ADR
513           STA   TPTR
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 117 and 118 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
514             LDA   #>SP0ADR
515             STA   TPTR+1
516             LDA   #64
517             STA   TABSIZ
518             LDA   #0
519             STA   TABSIZ+1
520             JSR   BLKFIL
521 *
522 * COPY HEART CHARACTER
523 *
524             LDA   #<SP0ADR
525             STA   TEMPA
526             LDA   #>SP0ADR
527             STA   TEMPA+1
528             LDA   #83
529             STA   CHCODE
530             JSR   GETADR
531             LDY   #0
532             LDX   #8
533 *
534 DEFSP0      LDA   (BPTR),Y
535             STA   (TEMPA),Y
536 *
537             INC   BPTR
538             INC   TEMPA
539             INC   TEMPA
540             INC   TEMPA
541 *
542             DEX
543             BNE   DEFSP0
544 *
545 * SET SPRITE POINTER
546 *
547             LDA   #$38
548             STA   SPRPTR
549 *
550             LDA   #0
551             STA   $FF00
552 *
553 * EXPAND SPRITE
554 *
555             LDA   #1
556             STA   XXPAND
557             STA   YXPAND
558 *
559 * TURN ON SPRITE #0
560 *
561             LDA   #1
562             STA   SPENA
563 *
564 * MAKE SPRITE RED
565 *
566             LDA   #10
567             STA   SP0COL
568 *
569             STA   $FF01
570 *
571 * POSITION SPRITE
572 *
573             LDA   #54
574             STA   SP0X
575             LDA   #0
576             STA   MSIGX
577             LDA   #34
578             STA   SP0Y
579 *
580 * MOVE SPRITE DOWN
581 *
582 DROP        INC   SP0Y
583 *
584 * DELAY LOOP
585 *
586             LDX   #$FF
587 XLOOP       LDY   #$10
588 YLOOP       DEY
589             BNE   YLOOP
590             DEX
591             BNE   XLOOP
592 *
593             LDA   SP0Y
594             CMP   #142
595             BNE   DROP
596 *
597 INF         JMP   INF
```

FILE MANIPULATOR
FROM PAGE 20

PLANETARY MOONS DATABASE

```
·1 REM                                          JD
·2 REM  --- PLANETARY MOONS DATABASE ---        GI
·3 REM              FOR C-128                   IL
·4 REM          RUPERT REPORT #36               OM
·5 REM                                          JD
·10 FOR N=1 TO 24 : SP$=SP$+" " : NEXT          AN
·20 DATA MERCURY,0,VENUS,0,EARTH,1,MARS,2       OF
·30 DATA JUPITER,16,SATURN,17,URANUS,5          GL
·40 DATA NEPTUNE,2,PLUTO,1                      BB
·50 DOPEN#1,"PLANETS",L25                       AO
·60 FOR NR=1 TO 9                               PO
·70 READ P$,M$                                  JM
·80 WR$=SP$                                     EB
·90 MID$(WR$,1)=P$                              IG
·100 MID$(WR$,15)=M$                            JO
·110 RECORD#1,NR,1                              IK
·120 PRINT#1,WR$                                LM
·130 PRINT DS$,WR$                              FI
·140 NEXT                                       IA
·150 DCLOSE                                     AM
```

FILE MANIPULATOR

```
·1 REM                                          JD
·2 REM       "FILE MANIPULATOR"                 GN
·3 REM  --- RELATIVE FILE MODEL ---             JP
·4 REM            FOR C-128                     IL
·5 REM         RUPERT REPORT #36                OM
·6 REM                                          JD
·10 TRAP 1000                                   JM
·20 SP$=" " : FOR K=1 TO 6 : SP$=SP$+SP$
 : NEXT  : REM  64 SPACES                       KO
·30 :REM === FILE DEFINITION ===========        KE
·40 F$="PLANETS"          : REM  FILENAME       IL
·50 NR=60                 : REM 60 RECORDS      PI
·60 : REM  2 FIELDS, LENGTHS 14 & 10            OP
·70 FLD(1)=14 : FLD(2)=10                       GA
·80 : REM  POINTERS TO START OF FIELDS          NG
·90 PT(1)=1 : PT(2)=15                          LD
·100 :REM === INITIALIZE FILE ==========        JF
·110 :REM   25 BYTES PER RECORD                 EP
·120 :REM    MAX # RECORDS = NR                 IC
·130 DOPEN#1,(F$),L25                           EA
·140 RECORD#1,NR                                DK
·150 :REM IF LAST RECORD EMPTY, WRITE IT        EB
·160 IF VAL(DS$)=50 THEN PRINT#1,"*END*"        KG
·170 DCLOSE                                     AM
```

```
·180 :REM ------------------------------ PO
·190 :REM === MAIN LOOP ================ CE
·200 INPUT"READ, WRITE, OR QUIT (R/W/Q)";
 U$                                      BN
·210 ON INSTR("RWQ",U$) GOTO 260,400,230 MO
·220 GOTO 200                            BO
·230 DCLOSE : END                        ML
·240 REM ------------------------------ DD
·250 REM ==== READ ROUTINE ============= ED
·260 DOPEN#1,(F$)                        EP
·270 INPUT">READ WHICH RECORD (0=STOP)";N HC
·280 IF N=0 THEN DCLOSE : GOTO 200       OC
·290 IF N<1 OR N>NR THEN 270             HM
·300 GOSUB 320  :REM  READ & DISPLAY     DL
·310 GOTO 270                            CJ
·320 :REM  'READ & DISPLAY' SUBROUTINE   BD
·330 RECORD#1,N                          OF
·340 INPUT#1,RD$                         JJ
·350 PRINT"FULL RECORD ="RD$             BF
·360 FOR F=1 TO 2                        II
·370 PRINT"FIELD #"F"="MID$(RD$,PT(F),FLD
 (F)) : NEXT : PRINT                     AI
·380 RETURN                              IM
·390 REM ===== WRITE ROUTINE =========== GD
·400 DOPEN#1,(F$)                        EP
·410 INPUT">WRITE WHICH RECORD(0=STOP)";N GE
·420 IF N=0 THEN DCLOSE : GOTO 200       OC
·430 IF N<1 OR N>NR THEN 410             GE
·440 :REM  READ & DISPLAY ROUTINE        DP
·450 GOSUB 320                           CJ
·460 INPUT">WRITE WHICH FIELD (0,1,2)";F DI
·470 IF F<1 OR F>2 THEN 410              MF
·480 PRINT"ENTER UP TO"FLD(F)"CHARACTERS PB
·490 PRINT"  "; : FOR K=1 TO FLD(F) : PRI
 NT"-"; : NEXT : PRINT                   HP
·500 U$="" : INPUT U$       :REM GET DATA OF
·510 U$=U$+SP$              :REM  PAD IT  FP
·520 U$=LEFT$(U$,FLD(F))    :REM TRUNCATE PF
·530 :REM  GET THE CURRENT RECORD        CN
·540 WR$=RD$      :REM  FROM READ ROUTINE PH
·550 WR$=WR$+SP$          :REM  PAD IT   IK
·560 MID$(WR$,PT(F))=U$   :REM  UPDATE IT IG
·570 WR$=LEFT$(WR$,24)    :REM TRUNCATE  GL
·580 RECORD#1,N,1       :REM SET POINTER FL
·590 PRINT#1,WR$        :REM WRITE RECORD BN
·600 GOTO 410                            DP
·999 :REM ============================== KJ
·1000 PRINT DS$,ERR$(ER),"LINE "EL       JA
·1010 RESUME NEXT                        HA
```

THE ARTIST
FROM PAGE 34

BOOTER

```
·5 REM *===* THE ARTIST (V2.0) BOOTER *==
 =*                                      DI
·10 COLOR0,1:COLOR4,1:PRINT"[CLEAR][DOWN]
 [WHITE]ONE MOMENT PLEASE.               JL
·20 PRINT"[DOWN][BLACK]RUN"CHR$(34)"THE A
 RTIST                                   FO
·30 PRINT"[HOME]                         MI
·40 POKE842,13:POKE208,1:POKE46,108:POKE2
 7648,.:NEW                              HH
```

SPRITE DATA

```
>00E00 00 00 00 00 00 00 00 00
>00E08 00 00 00 00 00 30 00 00
>00E10 30 00 00 30 00 00 78 00
>00E18 00 CC 00 0F 87 C0 00 CC
>00E20 00 00 78 00 00 30 00 00
>00E28 30 00 00 30 00 00 00 00
>00E30 00 00 00 00 00 00 00 00
>00E38 00 00 00 00 00 00 00 00
>00E40 00 00 00 00 00 00 00 00
>00E48 00 00 00 00 00 00 00 00
>00E50 00 00 00 00 00 00 00 00
>00E58 00 00 00 00 30 00 00 00
>00E60 00 00 00 00 00 00 00 00
>00E68 00 00 00 00 00 00 00 00
>00E70 00 00 00 00 00 00 00 00
>00E78 00 00 00 00 00 00 00 00
>00E80 00 00 00 60 00 00 60 00
>00E88 00 10 00 00 08 00 00 04
>00E90 00 00 00 00 00 7B 66 00
>00E98 63 66 00 7B 66 00 63 66
>00EA0 00 63 77 00 00 00 00 00
>00EA8 00 00 00 00 00 00 00 00
>00EB0 00 00 00 00 00 00 00 00
>00EB8 00 00 00 00 00 00 00 00
```

ML

```
>00B00 A0 00 B1 FA 91 FC C8 D0
>00B08 F9 E6 FB E6 FD C6 FE A6
>00B10 FE D0 EF 60 00 00 00 00
```

THE ARTIST

```
·2 TRAP306:GOSUB280:GOTO52               PJ
·4 :                                     DI
·6 :                                     DI
·8 :       THE ARTIST (V2.0)             FI
·10 :                                    DI
·12 :                                    DI
·14 :          DENOY  DEBOER             GL
·16 :    70 B SOUTH MAGNOLIA DR.         KD
·18 :   SATELLITE BEACH, FL 32937        EL
·20 :         (305) 773-9343             IC
·22 :                                    DI
·24 :                                    DI
```

```
·26 :                                                          DI
·28 REM *===* DRAWING MODE *===*                               BO
·32 J=JOY(2):X=X-(J>1ANDJ<5)+(J>5ANDJ<9):
 Y=Y-(J>3ANDJ<7)+(J=8ORJ=1ORJ=2):IFJ>127T
 HENPRINT"[CNTRL G]";:P=-P:GOSUB322                            DC
·34 SPRITE8,1,RND(1)*16+1:MOVSPR8,X*2+Q+1
 ,Y+U:IFP=-1THENDRAWDC,X,Y                                     DO
·36 GOSUB44:IFA$<>""THEN52:ELSE32                              IL
·38 REM *===* READ JOYSTICK *===*                              GI
·40 J=JOY(2):IFJ=.THENS=.:RETURN:ELSES=CS
 :A=(J-1)*45:RETURN                                            JH
·42 REM *===* CHECK TO SEE IF USER WANTS
 TO SAVE OR RESTORE SCREEN FROM BUFFER *=
 ==*                                                           FO
·44 GETA$:IFA$="S"THENPRINT"[CNTRL G]";:G
 OTO230:REM BUFFER SAVE SCREEN                                 EN
·46 IFA$="R"THENPRINT"[CNTRL G]";:GOTO240
 :REM RESTORE SCREEN                                           CL
·48 RETURN                                                     IM
·50 REM *===* MAIN MENU *===*                                  CA
·52 GRAPHIC0:COLOR0,1:OC=.:SPRITE1,1,2,1:
 SPRITE2,1,2,1:GOSUB286:COLOR1,C1+1:COLOR
 2,C2+1:COLOR3,C3+1:IFC4>.ANDC4<17THENCOL
 OR4,C4:ELSEC4=1:COLOR4,C4                                     DM
·54 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT"
 [CNTRL G]";:GOTO62:REM BUTTON PUSHED                          GM
·56 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GET
 CS:IFCS>.THENOS=CS:ELSECS=OS                                  CL
·58 GOTO54                                                     PP
·60 REM *===* CHECK TO SEE WHAT SELECTION
  THE USER MADE *===*                                          BO
·62 X1=RSPPOS(8,.):Y1=RSPPOS(8,1):X=(X1-Q
 )/8:Y=(Y1-U)/8                                                JL
·64 IFY>.ANDY<4THEN78                                          KJ
·66 IFY>7ANDY<11THEN84                                         OL
·68 IFY>11ANDY<15THEN92                                        EI
·70 IFY>15ANDY<19THEN100                                       HM
·72 IFY>20ANDY<23THEN120                                       GJ
·74 GOTO52                                                     PB
·76 REM *===* THE VARIOUS OPTIONS *===*                       LI
·78 IFX>.ANDX<7THEN174:REM LOAD                                NJ
·80 IFX>7ANDX<14THEN166:REM SAVE                               AJ
·82 IFX>14ANDX<21THENGOSUB290:J=.:DO WHIL
 E J<127:J=JOY(2):LOOP:PRINT"[CNTRL G]";:
 GOTO52:REM VIEW                                               IN
·84 IFX>.ANDX<6THENWIDTH2:SPRITE8,,,,1:Q=
 3:GOTO54:REM LARGE  BRUSH                                     IH
·86 IFX>6ANDX<12THENWIDTH1:SPRITE8,,,,0:Q
 =13:GOTO54:REM SMALL BRUSH                                    DC
·88 IFX>12ANDX<16THENSPRSAV1,8:GOTO54:REM
  CROSSHAIR CURSOR                                             FG
·90 IFX>16ANDX<20THENSPRSAV2,8:GOTO54:REM
  SOLID CURSOR                                                 AD
·92 GOSUB322:IFX>.ANDX<5THENGOSUB290:GOTO
 212:REM RAYS                                                  OC
·94 IFX>5ANDX<12THENGOSUB290:GOTO136:REM
 CIRCLE                                                        JA
·96 IFX>12ANDX<16THENGOSUB290:GOTO156:REM
   BOX                                                         AE
·98 IFX>16ANDX<20THEN224:REM NEW                               NO
·100 IFX>.ANDX<5THENX=85:Y=90:GOSUB290:P=
 1:GOTO32:REM DRAW                                             ED
·102 IFX>5ANDX<10THEND1=.:GOSUB290:GOTO18
 8:REM LINE                                                    EJ
·104 IFX>10ANDX<15THENGOSUB290:SPRSAV3,8:
 GOSUB322:GOTO130:REM FILL                                     HJ
·106 IFX>15ANDX<20THEN198:REM TEXT                             IE
·108 REM *===* DETERMINE THE PALETTE TO B
 E USED *===*                                                  MD
·110 IFX>22ANDX<27THENDC=0:IFOC=1THENC0=C
 -1:GOTO52                                                     JL
·112 IFX>26ANDX<31THENDC=1:IFOC=1THENC1=C
 -1:GOTO52                                                     IM
·114 IFX>30ANDX<35THENDC=2:IFOC=1THENC2=C
 -1:GOTO52                                                     IB
·116 IFX>34ANDX<39THENDC=3:IFOC=1THENC3=C
 -1:GOTO52                                                     NJ
·118 REM *===* DETERMINE COLOR TO BE PUT
 INTO PALETTE *===*                                            OA
·120 OC=1:IFX>3ANDX<6THENC=2:GOTO54:REM W
 HITE                                                          NL
·122 IFX>5ANDX<8THENC=1:GOTO54:REM BLACK                       BF
·124 C=INT(X/2)-1:IFC>16THENOC=.:REM CALC
 ULATE COLOR                                                   PL
·126 GOTO 54                                                   PP
·128 REM *===* FILL *===*                                      NP
·130 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
 "[CNTRL G]";:X=RSPPOS(8,.)-24:Y=RSPPOS(8
 ,1)-50:PAINTDC,X/2,Y:GOTO130                                  MH
·132 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
 SUB44:IFA$=""THEN130:ELSESPRSAV1,8:GOTO5
 2                                                             DF
·134 REM *===* CIRCLE *===*                                    GF
·136 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
 "[CNTRL G]";:X=RSPPOS(8,.)-Q:Y=RSPPOS(8,
 1)-U:DRAWDC,X/2,Y:CX=RDOT(.):CY=RDOT(1):
 GOSUB322:GOTO140                                              DI
·138 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
 SUB44:IFA$=""THEN136:ELSE52                                   OO
·140 GOSUB40:IFA>.ANDA<>180THENS=.:REM RE
 AD JOYSTICK                                                   PC
·142 IFJ>127THENMOVSPR8,A#.:PRINT"[CNTRL
 G]";:Y1=RSPPOS(8,1)-U:Y2=ABS(Y1-Y):DRAWD
 C,X/2,Y1:MOVSPR8,CX*2+Q,CY+U:GOSUB322:GO
 TO146                                                         DJ
·144 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
 SUB44:IFA$=""THEN140:ELSE52                                   EJ
·146 GOSUB40:IFA<>90ANDA<>270THENS=.:REM
 READ JOYSTICK                                                 EH
·148 IFJ>127THENMOVSPR8,A#.:PRINT"[CNTRL
 G]";:X1=RSPPOS(8,.):IFX1=CX*2+QANDRSPPOS
 (8,1)=CY+UTHENCIRCLEDC,X/2,Y,Y2*SC,Y2,,,
 ,1:GOTO136                                                    KF
·150 IFJ>127THENX2=ABS((X1-Q)-X):CIRCLEDC
 ,X/2,Y,X2/2,Y2,,,,1:GOTO136                                   PA
·152 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
```

```
SUB44:IFA$=""THEN146:ELSE52                                    PP
·154 REM *===* BOX *===*                                       HO
·156 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
"[CNTRL G]";:X=RSPPOS(8,.)-Q:Y=RSPPOS(8,
1)-U:DRAWDC,X/2,Y:GOSUB322:GOTO160                             IG
·158 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
SUB44:IFA$=""THEN156:ELSE52                                    JI
·160 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
"[CNTRL G]";:Y1=RSPPOS(8,1)-U:X1=RSPPOS(
8,.)-Q:BOXDC,X/2,Y,X1/2,Y1:GOSUB322:GOTO
156                                                            KM
·162 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
SUB44:IFA$=""THEN160:ELSE52                                    JL
·164 REM *===* SAVE *===*                                      ML
·166 N$="":CHAR,0,24,"[WHITE]NAME?":GOSUB
294                                                            CA
·168 GOSUB276:IFN$=""THEN52:ELSEIFLEN(N$)
>12THENE$="NAME MUST BE 12 CHARACTERS OR
 LESS":GOSUB288:GOTO52                                         BO
·170 POKE8168,C0:POKE8169,C1:POKE8170,C2:
POKE8171,C3:BSAVE(N$)+".PIC",B0,P7168 TO
 P16192:GOSUB182:BSAVE(N$)+".COL",B15,P5
5296 TO P56395:GOSUB184:GOTO52                                 KF
·172 REM *===* LOAD *===*                                      KA
·174 N$="":CHAR,0,24,"[WHITE]NAME?":GOSUB
294                                                            CA
·176 GOSUB276:IFN$=""THEN52:ELSEIFLEN(N$)
>12THENE$="NAME MUST BE 12 CHARACTERS OR
 LESS":GOSUB288:GOTO52                                         BO
·178 FAST:GRAPHICM:GOSUB182:BLOAD(N$)+".C
OL",B15,P55296:GOSUB184                                        NJ
·180 BLOAD(N$)+".PIC",B0,P7168:C0=PEEK(81
68):C1=PEEK(8169):C2=PEEK(8170):C3=PEEK(
8171):GOTO52                                                   IL
·182 BANK15:POKE(DEC("D01A")),PEEK(DEC("D
01A"))AND254:POKE1,PEEK(1)AND254:RETURN                        MJ
·184 POKE1,PEEK(1)OR1:POKE(DEC("D01A")),P
EEK(DEC("D01A"))OR1:SLOW:RETURN                                KH
·186 REM *===* LINE *===*                                      MC
·188 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
"[CNTRL G]";:X=RSPPOS(8,.)-Q:Y=RSPPOS(8,
1)-U:DRAWDC,X/2,Y:GOSUB322:GOTO192                             IP
·190 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
SUB44:IFA$=""THEN188:ELSE52                                    PN
·192 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
"[CNTRL G]";:X1=RSPPOS(8,.)-Q:Y1=RSPPOS(
8,1)-U:DRAWDC,X/2,Y TO X1/2,Y1:GOSUB322:
S=.:GOTO190                                                    LN
·194 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
SUB44:IFA$=""THEN192:ELSE52                                    OA
·196 REM *===* TEXT *===*                                      PB
·198 N$="":CHAR,0,24,"[WHITE]TEXT:":GOSUB
294                                                            MD
·200 GOSUB276:IFN$=""THEN52                                    IH
·202 PRINT"[CNTRL G]";:CHAR,12,24,"1=[RVS
ON]RVS[RVSOFF][3" "]2=NORMAL":TM=DC:GETK
EYA$:IFA$="2"THENTM=.                                          MB
·204 GOSUB276:GOSUB290                                         JA
·206 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
"[CNTRL G]";:X1=RSPPOS(8,.):Y1=RSPPOS(8,
1):X=(X1-Q)/8:Y=(Y1-U)/8:CHARDC,X,Y,N$,T
M:GOTO206                                                      FE
·208 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
SUB44:IFA$=""THEN206:ELSE52                                    NE
·210 REM *===* RAYS *===*                                      MN
·212 GOSUB40:IFJ>127THENMOVSPR8,A#.:PRINT
"[CNTRL G]";:X=RSPPOS(8,.)-Q:Y=RSPPOS(8,
1)-U:DRAWDC,X/2,Y:GOSUB322:GOTO216                             II
·214 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
SUB44:IFA$=""THEN212:ELSE52                                    EJ
·216 GOSUB40:IFJ>128THENX1=RSPPOS(8,.)-Q:
Y1=RSPPOS(8,1)-U:DRAWDC,X/2,Y TO X1/2,Y1                       AB
·218 IFJ=128THENGOSUB322:GOSUB40:IFJ=128T
HEN212                                                         BE
·220 MOVSPR8,A#S:SPRITE8,1,RND(1)*16+1:GO
SUB44:IFA$=""THEN216:ELSE52                                    KN
·222 REM *===* NEW *===*                                       LH
·224 CHAR,0,24,"[WHITE]ARE YOU SURE? (Y/N
)":GETKEYA$:IFA$="Y"THENGRAPHICM,1:GRAPH
IC0                                                            DI
·226 GOSUB276:GOTO52                                           FN
·228 REM *===* SAVE SCREEN TO BUFFER *===
*                                                              AF
·230 FAST:R0=C0:R1=C1:R2=C2:R3=C3                              ON
·232 POKE250,.:POKE251,28:POKE252,.:POKE2
53,64:POKE254,36:BANK0:SYS2816:REM SAVE
IMAGE                                                          KM
·234 GOSUB182:FORI=.TOD:C(I)=PEEK(W+I):NE
XT:GOSUB184:REM SAVE COLOR                                     JH
·236 PRINT"[CNTRL G]";:A$="":RETURN                            JN
·238 REM *===* RESTORE SCREEN FROM BUFFER
 *===*                                                         JD
·240 FAST:POKE250,.:POKE251,64:POKE252,.:
POKE253,28:POKE254,36:BANK0:SYS2816:REM
RESTORE IMAGE                                                  OB
·242 C0=R0:C1=R1:C2=R2:C3=R3                                   PJ
·244 GOSUB182:FORI=.TOD:POKEW+I,C(I):NEXT
:GOSUB184:REM RESTORE COLOR                                    ME
·246 COLOR0,C0+1:PRINT"[CNTRL G]";:A$="":
RETURN                                                         EE
·248 REM *===* PRINT MENU SCREEN *===*                         BP
·250 PRINT"[CLEAR][WHITE][RVSON][s M][38"
"][s N] [RVSOFF][6" "][RVSON] [RVSOFF][
6" "][RVSON] [RVSOFF][6" "][RVSON] [RVSO
FF] [RVSON] [RVSOFF][15" "][RVSON] ";                          JO
·252 PRINT"[RVSON] [RVSOFF] [YELLOW]LOAD[
WHITE] [RVSON] [RVSOFF] [YELLOW]SAVE[WHI
TE] [RVSON] [RVSOFF] [YELLOW]VIEW[WHITE]
 [RVSON] [RVSOFF] [RVSON] "SPC(15)"  [RV
SOFF][6" "][RVSON] [RVSOFF][6" "][RVSON]
 [RVSOFF][6" "][RVSON] [RVSOFF] [RVSON]
"SPC(15)" ";                                                   FJ
·254 PRINT"[RVSON][22" "][RVSOFF] [RVSON]
 [RVSOFF][15" "][RVSON]  "TAB(23)" "SPC(
15)" ";                                                        LH
·256 PRINT"[RVSON][21" "][RVSOFF]  [RVSON
```

```
] "SPC(15)"  [RVSOFF][5" "][RVSON] [RVSO
FF][5" "][RVSON] [RVSOFF][3" "][RVSON] [
RVSOFF][3" "][RVSON] [RVSOFF]  [RVSON] "
SPC(15)" ";                                   PG
·258 PRINT"[RVSON] [RVSOFF]LARGE[RVSON] [
RVSOFF]SMALL[RVSON] [RVSOFF][3" "][RVSON
] [RVSOFF][3" "][RVSON] [RVSOFF]  [RVSON
] "SPC(15)"  [RVSOFF][5" "][RVSON] [RVSO
FF][5" "][RVSON] [RVSOFF][3" "][RVSON] [
RVSOFF][3" "][RVSON] [RVSOFF]  [RVSON] "
SPC(15)" ";                                   OK
·260 PRINT"[RVSON][21" "][RVSOFF]  [RVSON
] "SPC(15)"  [RVSOFF][4" "][RVSON] [RVSO
FF][6" "][RVSON] [RVSOFF][3" "][RVSON] [
RVSOFF][3" "][RVSON] [RVSOFF]  [RVSON] "
SPC(15)" ";                                   GA
·262 PRINT"[RVSON] [RVSOFF]RAYS[RVSON] [R
VSOFF]CIRCLE[RVSON] [RVSOFF]BOX[RVSON] [
RVSOFF]NEW[RVSON] [RVSOFF]  [RVSON] "SPC
(15)"  [RVSOFF][4" "][RVSON] [RVSOFF][6"
 "][RVSON] [RVSOFF][3" "][RVSON] [RVSOFF
][3" "][RVSON] [RVSOFF]  [RVSON][17" "]"
;                                             PO
·264 PRINT"[RVSON][21" "]"SPC(18)"  [RVSO
FF][4" "][RVSON] [RVSOFF][4" "][RVSON] [
RVSOFF][4" "][RVSON] [RVSOFF][4" "][RVSO
N] "SPC(18)" ";                               DD
·266 PRINT"[RVSON] [RVSOFF]DRAW[RVSON] [R
VSOFF]LINE[RVSON] [RVSOFF]FILL[RVSON] [R
VSOFF]TEXT[RVSON] "SPC(18)"  [RVSOFF][4"
 "][RVSON] [RVSOFF][4" "][RVSON] [RVSOFF
][4" "][RVSON] [RVSOFF][4" "][RVSON] "SP
C(18)" ";                                     GH
·268 PRINT"[RVSON][41" "]"SPC(38)" ";         JL
·270 PRINT"[RVSON] [RVSOFF][3" "][RVSON][
WHITE] [BLACK] [RED] [CYAN] [PURPLE]
  [GREEN] [BLUE] [YELLOW] [c 1] [c 2
] [c 3] [c 4] [c 5] [c 6] [c 7] [c
 8] [RVSOFF][WHITE][3" "][RVSON] ";           AD
·272 PRINT"[RVSON] [RVSOFF][3" "][RVSON][
WHITE] [BLACK] [RED] [CYAN] [PURPLE]
  [GREEN] [BLUE] [YELLOW] [c 1] [c 2
] [c 3] [c 4] [c 5] [c 6] [c 7] [c
 8] [RVSOFF][WHITE][3" "][RVSON] ";           AD
·274 PRINT"[RVSON] "SPC(38)" [s N][38" "]
[s M]":RETURN                                 IN
·276 CHAR,0,24,"[39" "]":RETURN:REM 39 SP
ACES                                          NG
·278 REM *===* INITIALIZATION *===*           LN
·280 GOSUB318:FAST:D=999:DIMC(D):W=55296:
GRAPHIC3:GRAPHIC0,1:COLOR0,1:GOSUB250:OS
=5:SC=.646:C4=1                               LD
·282 MOVSPR1,129,108:MOVSPR2,161,108:X=17
1:Y=148:C=1:Q=13:U=41:M=3:P$="[RVSON][4"
 "][DOWN][4"[LEFT]"][4" "][DOWN][4"[LEFT
]"][4" "][RVSOFF]":PC=241:C0=.:C1=1:C2=2
:C3=6:DC=C1:SPRSAV1,8                         II
·284 MOVSPR8,162,141:SLOW:RETURN              ON
·286 POKEPC,C0:CHAR,23,15,P$:POKEPC,C1:CH
AR,27,15,P$:POKEPC,C2:CHAR,31,15,P$:POKE
PC,C3:CHAR,35,15,P$:OC=.:RETURN               PO
·288 FORI=.TO2:PRINT"[WHITE][CNTRL G]";:C
HAR,0,24,E$:PRINT"[CNTRL G]";:FORT=.TO35
0:NEXT:GOSUB276:FORT=.TO10:NEXT:NEXT:RET
URN                                           MP
·290 GRAPHICM:COLOR0,C0+1:SPRITE1,.:SPRIT
E2,.:SPRITE8,.:RETURN                         LJ
·292 REM *===* INPUT ROUTINE FOR TEXT, LO
AD, AND SAVE *===*                            NC
·294 GETKEYA$:IFA$="*"ORA$="[CLEAR]"ORA$=
"[HOME]"ORA$="[DOWN]"ORA$="[UP]"ORA$="[R
IGHT]"ORA$="[LEFT]"THEN294                    BE
·295 IFPOS(X)=39THENPRINT"[LEFT][CNTRL G]
";:GOTO294                                    CM
·296 IFA$=CHR$(13)THENRETURN                  DI
·298 IFA$=CHR$(20)ANDN$=""THEN294             LG
·300 IFA$=CHR$(20)THENL1=LEN(N$):N$=LEFT$
(N$,L1-1):PRINTA$;:GOTO294                    KF
·302 PRINTA$;:N$=N$+A$:GOTO294                HO
·304 REM *===* ERROR CORRECTION *===*         GB
·306 IFDS=62THENGOSUB184                      CE
·308 GRAPHIC0:SLOW:GOSUB276:SPRSAV1,8:IFD
S>1THENE$=DS$:GOSUB288:RESUME52:REM DISK
 ERROR                                        HJ
·310 IFX1<.THENX1=.:RESUME                    DA
·312 IFY1<.THENY1=.:RESUME                    BO
·314 RESUME52:REM GO BACK TO MENU             HP
·316 REM *===* LOAD SPRITES AND MACHINE L
ANGUAGE ROUTINE IF NECESSARY *===*            EN
·318 IFPEEK(3584)=.ANDPEEK(3600)=48THENRE
TURN:REM ROUTINES HAVE BEEN LOADED            BI
·320 BLOAD"ART SPR",B0,P3584:BLOAD"ML",B0
,P2816:RETURN                                 HN
·321 REM *===* DELAY LOOP FOR FIRE BUTTON
 PRESSES *===*                                EL
·322 FORT=.TO150:NEXT:RETURN                  IK
```

MINOTAUR MAZE
FROM PAGE 52

Starting address in hex: C000
Ending address in hex: C957
SYS to start: 49152

***Flankspeed* required for entry! See page 119.**

```
C000: 78 A5 01 29 FB 85 01 A9 74
C008: 00 85 FB 85 FD A9 D0 85 0D
C010: FC A9 38 85 FE A2 08 A0 BE
C018: 00 B1 FB 91 FD 88 D0 F9 A8
C020: E6 FC E6 FE CA D0 F0 A5 1C
C028: 01 09 04 85 01 58 20 B2 E7
C030: C5 20 06 C6 AD 18 D0 09 82
C038: 0E 8D 18 D0 A9 FF 8D 0F 03
C040: D4 A9 80 8D 12 D4 A9 00 5D
C048: 8D D9 02 8D DA 02 8D 20 C9
C050: D0 A2 40 A9 FF 9D 40 03 8E
```

```
C058: CA 10 FA A9 07 8D 27 D0 64
C060: A9 01 8D 17 D0 8D 1D D0 FB
C068: 8D 1B D0 A9 0D 8D F8 07 26
C070: A9 00 8D D5 02 8D D7 02 E6
C078: 8D D8 02 A9 03 8D DB 02 F8
C080: A9 04 8D D6 02 A9 93 20 F1
C088: D2 FF A9 00 20 27 C7 20 34
C090: 66 C6 A2 00 A0 00 18 20 39
C098: F0 FF A9 F2 A0 C7 20 1E CC
C0A0: AB A2 09 A0 0E 18 20 F0 CF
C0A8: FF A9 40 A0 C8 20 1E AB E5
C0B0: A2 18 A0 00 18 20 F0 FF 35
C0B8: A9 1B A0 C8 20 1E AB A9 7A
C0C0: 0E 8D B8 02 AE D5 02 E8 86
C0C8: 8A 0A 0A 8D CA 02 20 2A 0C
C0D0: C5 AE A8 02 AC A9 02 A9 F1
C0D8: 24 18 20 41 C7 CE CA 02 D9
C0E0: D0 EC 20 CA C4 20 4E C4 81
C0E8: 20 C1 C1 20 D8 C2 AD AE A4
C0F0: 02 C9 01 D0 3F AD AF 02 2D
C0F8: C9 02 D0 38 AD CB 02 D0 1A
C100: 33 EE D5 02 A9 00 8D CA FB
C108: 02 AD B1 02 0A 2E CA 02 70
C110: 0A 2E CA 02 18 6D B1 02 4E
C118: 90 03 EE CA 02 0A 2E CA 6A
C120: 02 18 6D D7 02 8D D7 02 E8
C128: AD CA 02 6D D8 02 8D D8 51
C130: 02 4C 85 C0 AD AE 02 CD F0
C138: BA 02 D0 0B AD AF 02 CD FD
C140: BB 02 D0 03 4C 57 C1 AD E4
C148: AE 02 CD C8 02 D0 43 AD 53
C150: AF 02 CD C9 02 D0 3B A9 51
C158: 00 8D 12 D4 CE D6 02 D0 45
C160: 2B A2 08 A0 27 18 20 F0 27
C168: FF A9 4B A0 C6 20 1E AB AE
C170: 20 01 C4 A2 20 8E 12 04 BD
C178: 20 D8 C2 A9 FD 85 A2 A5 A9
C180: A2 D0 FC A5 C5 C9 04 D0 FA
C188: EF 4C 70 C0 20 01 C4 4C 28
C190: 85 C0 A9 80 8D 12 D4 A2 18
C198: 20 A0 FF 88 D0 FD CA D0 4C
C1A0: F8 AD D6 02 A2 20 38 C9 E4
C1A8: 04 B0 03 8E 18 04 38 C9 0D
C1B0: 03 B0 03 8E 16 04 38 C9 12
C1B8: 02 B0 03 8E 14 04 4C E8 4A
C1C0: C0 AD AE 02 8D B6 02 AD D3
C1C8: AF 02 8D B7 02 AD 00 DC 4C
C1D0: C9 7E D0 15 AE AE 02 AC 0B
C1D8: AF 02 88 38 20 41 C7 C9 3E
C1E0: BA F0 03 CE AF 02 4C 2F 8B
C1E8: C2 C9 7D D0 14 AE AE 02 37
C1F0: AC AF 02 C8 38 20 41 C7 79
C1F8: C9 BA F0 EA EE AF 02 D0 CA
C200: E5 C9 7B D0 14 AE AE 02 6F
C208: CA AC AF 02 38 20 41 C7 92
C210: C9 BA F0 D2 CE AE 02 D0 A8
C218: CD C9 77 D0 12 AE AE 02 69
C220: E8 AC AF 02 38 20 41 C7 C8
C228: C9 BA F0 03 EE AE 02 AE EE
C230: AE 02 AC AF 02 38 20 41 D8
C238: C7 C9 24 D0 1D EE B1 02 7E
C240: CE CB 02 20 2B C4 A9 32 C8
C248: 18 6D D7 02 8D D7 02 A9 B8
C250: 00 6D D8 02 8D D8 02 20 21
C258: 4E C4 A9 00 AE CC 02 F0 83
C260: 02 A9 0F 8D B8 02 A9 23 30
C268: AE AE 02 AC AF 02 18 20 5E
C270: 41 C7 AE B6 02 EC AE 02 7E
C278: D0 0B AC B7 02 CC AF 02 39
C280: D0 03 4C AA C2 A9 20 AE 86
C288: B6 02 AC B7 02 18 20 41 21
C290: C7 A9 0A 8D 0F D4 A9 08 2F
C298: 8D 0E D4 A9 F0 8D 14 D4 1A
C2A0: A9 00 8D 13 D4 A9 81 8D 78
C2A8: 12 D4 AD AE 02 0A 0A 0A 0C
C2B0: 18 69 05 8D 00 D0 AD AE F1
C2B8: 02 38 C9 20 90 08 A9 01 20
C2C0: 8D 10 D0 4C CB C2 A9 00 B3
C2C8: 8D 10 D0 AD AF 02 0A 0A AA
C2D0: 0A 18 69 23 8D 01 D0 60 3F
C2D8: CE CE 02 30 01 60 AD DB 93
C2E0: 02 8D CE 02 AD BA 02 8D 39
C2E8: C0 02 AD BB 02 8D C1 02 68
C2F0: AE B9 02 CA 10 02 A2 03 DD
C2F8: 8E C2 02 AD C2 02 0A AA 73
C300: AD BA 02 18 7D EA C7 8D 40
C308: A8 02 E8 AD BB 02 18 7D 9C
C310: EA C7 8D A9 02 AE A8 02 55
C318: AC A9 02 38 20 41 C7 C9 9B
C320: BA D0 12 EE C2 02 AD C2 E1
C328: 02 C9 04 D0 CE A9 00 8D CE
C330: C2 02 4C FB C2 C9 24 D0 BE
C338: 03 CE CB 02 AD C2 02 8D D7
C340: B9 02 8E BA 02 8C BB 02 91
C348: A9 09 8D B8 02 A9 25 18 2A
C350: 20 41 C7 AD C0 02 CD BA 72
C358: 02 D0 0B AD C1 02 CD BB 31
C360: 02 D0 03 4C 72 C3 A9 20 82
C368: AE C0 02 AC C1 02 18 20 82
C370: 41 C7 AD C8 02 8D C3 02 45
C378: AD C9 02 8D C4 02 AE C7 BC
C380: 02 CA 10 02 A2 03 8E C5 59
C388: 02 AD C5 02 0A AA AD C8 2B
C390: 02 18 7D EA C7 8D A8 02 13
C398: E8 AD C9 02 18 7D EA C7 43
C3A0: 8D A9 02 AE A8 02 AC A9 89
C3A8: 02 38 20 41 C7 C9 BA D0 61
C3B0: 12 EE C5 02 AD C5 02 C9 B8
C3B8: 04 D0 CE A9 00 8D C5 02 5B
C3C0: 4C 89 C3 C9 24 D0 03 CE EA
C3C8: CB 02 AD C5 02 8D C7 02 63
C3D0: 8E C8 02 8C C9 02 A9 0B 37
C3D8: 8D B8 02 A9 25 18 20 41 69
C3E0: C7 AD C3 02 CD C8 02 D0 85
C3E8: 0B AD C4 02 CD C9 02 D0 D2
C3F0: 03 4C 00 C4 A9 20 AE C3 41
```

```
C3F8: 02 AC C4 02 18 20 41 C7 AF
C400: 60 A9 FF 8D 08 D4 A9 0A 28
C408: 8D 07 D4 A9 F0 8D 0D D4 7B
C410: A9 21 8D 0B D4 A2 0F A9 A3
C418: FE 85 A2 A5 A2 D0 FC CA 20
C420: 8E 08 D4 D0 F2 A9 00 8D 86
C428: 0B D4 60 A9 6E 8D 08 D4 EA
C430: A9 09 8D 0C D4 8D 0D D4 C0
C438: 8D 07 D4 A9 11 8D 0B D4 C9
C440: A9 FD 85 A2 A5 A2 D0 FC 26
C448: A9 00 8D 0B D4 60 AD D8 46
C450: 02 38 CD DA 02 F0 04 90 BA
C458: 19 B0 0B AD D7 02 38 CD BA
C460: D9 02 F0 02 90 0C AD D7 51
C468: 02 8D D9 02 AD D8 02 8D E9
C470: DA 02 A2 00 A0 05 18 20 CD
C478: F0 FF A9 90 20 D2 FF A9 40
C480: 92 20 D2 FF AD DA 02 AE 3F
C488: D9 02 20 CD BD A2 00 A0 53
C490: 22 18 20 F0 FF AD D8 02 64
C498: AE D7 02 20 CD BD A2 18 87
C4A0: A0 0D 18 20 F0 FF A9 9B BC
C4A8: 20 D2 FF A9 00 AE D5 02 CB
C4B0: E8 20 CD BD A2 18 A0 21 C1
C4B8: 18 20 F0 FF A9 9A 20 D2 19
C4C0: FF A9 00 AE B1 02 20 CD BA
C4C8: BD 60 A2 14 A9 00 9D A7 8C
C4D0: 02 9D C0 02 CA 10 F7 A9 AF
C4D8: 01 8D AE 02 A9 02 8D AF 01
C4E0: 02 A9 25 8D BA 02 A9 16 BB
C4E8: 8D BB 02 A9 25 8D C8 02 5B
C4F0: A9 02 8D C9 02 AE D5 02 7C
C4F8: E8 8A 0A 0A 8D CB 02 AE 8A
C500: D5 02 38 E0 08 90 0D AD 44
C508: DB 02 F0 03 CE DB 02 A2 29
C510: 00 8E D5 02 BD CF C8 8D 5A
C518: CC 02 D0 08 A9 01 8D 15 0D
C520: D0 4C 29 C5 A9 00 8D 15 78
C528: D0 60 AD 1B D4 38 C9 01 F9
C530: 90 F8 38 C9 25 B0 F3 8D 13
C538: A8 02 AD 1B D4 38 C9 01 83
C540: 90 F8 38 C9 17 B0 F3 8D 15
C548: A9 02 AE A8 02 AC A9 02 A5
C550: 38 20 41 C7 C9 20 D0 D2 3F
C558: 60 A2 18 A9 00 9D 00 D4 8F
C560: CA 10 FA A9 F0 8D 06 D4 39
C568: A9 11 8D 04 D4 A9 0F 8D CF
C570: 18 D4 60 AE AC 02 BD E7 C0
C578: C7 8D AD 02 8A 0A AA BD 7A
C580: E8 C7 85 02 E8 BD E8 C7 10
C588: 85 03 CE AA 02 D0 22 A9 29
C590: 18 8D AA 02 AC AB 02 38 75
C598: CC AD 02 90 05 A0 00 8C D7
C5A0: AB 02 B1 02 8D 01 D4 C8 2E
C5A8: B1 02 8D 00 D4 C8 8C AB BF
C5B0: 02 60 A9 D3 8D 14 03 A9 DE
C5B8: C5 8D 15 03 A9 28 8D 12 95
C5C0: D0 AD 11 D0 29 7F 8D 11 68
C5C8: D0 A9 81 8D 1A D0 20 59 B6
C5D0: C5 58 60 AD 19 D0 8D 19 8D
C5D8: D0 29 01 D0 07 AD 0D DC 43
C5E0: 58 4C 31 EA AD 12 D0 38 6A
C5E8: C9 3A B0 0A A9 0F 8D 21 0F
C5F0: D0 A9 3A 4C FD C5 A9 00 5F
C5F8: 8D 21 D0 A9 28 8D 12 D0 BA
C600: 20 73 C5 4C BC FE A9 D7 E2
C608: 85 FB A9 C8 85 FC A0 00 1F
C610: B1 FB F0 36 A2 00 18 0A A9
C618: 48 8A 2A AA 68 0A 48 8A 05
C620: 2A AA 68 0A 48 8A 2A AA 0F
C628: 68 85 FD C6 FD 8A 18 69 E4
C630: 38 85 FE A0 08 B1 FB 91 D4
C638: FD 88 D0 F9 A5 FB 18 69 AC
C640: 09 85 FB 90 02 E6 FC 4C 8D
C648: 0E C6 60 12 9B 47 11 9D 21
C650: 41 11 9D 4D 11 9D 45 11 92
C658: 11 11 9D 4F 11 9D 56 11 7D
C660: 9D 45 11 9D 52 00 A9 51 3F
C668: 85 FB A9 28 85 FD A9 04 EC
C670: 85 FC 85 FE A2 00 A0 00 BA
C678: A9 BA 91 FD C8 C0 27 D0 ED
C680: F9 18 A5 FD 69 28 85 FD 4B
C688: 90 02 E6 FE E8 E0 17 D0 B2
C690: E5 A0 0E A2 09 18 20 F0 F9
C698: FF A9 40 A0 C8 20 1E AB D5
C6A0: A0 00 A9 04 91 FB A9 FF 26
C6A8: 8D 0F D4 A9 80 8D 12 D4 B8
C6B0: AD 1B D4 29 03 8D CA 02 D4
C6B8: AA 0A A8 18 B9 39 C7 65 4E
C6C0: FB 85 26 B9 3A C7 65 FC 86
C6C8: 85 27 18 B9 39 C7 65 26 D3
C6D0: 85 FD B9 3A C7 65 27 85 22
C6D8: FE A0 00 B1 FD C9 BA D0 7D
C6E0: 12 8A 91 FD A9 20 91 26 8E
C6E8: A5 FD 85 FB A5 FE 85 FC 35
C6F0: 4C A6 C6 E8 8A 29 03 CD 18
C6F8: CA 02 D0 BC B1 FB AA A9 55
C700: 20 91 FB E0 04 F0 1A 8A 28
C708: 0A A8 A2 02 38 A5 FB F9 33
C710: 39 C7 85 FB A5 FC F9 3A 69
C718: C7 85 FC CA D0 EE 4C A6 DF
C720: C6 AE D5 02 BD CF C8 A0 64
C728: 00 99 00 D8 99 00 D9 99 A7
C730: 00 DA 99 00 DB C8 D0 F1 0C
C738: 60 01 00 D8 FF FF FF 28 9A
C740: 00 8D A7 02 8E A8 02 8C 3D
C748: A9 02 48 8A 08 48 98 A2 52
C750: 00 0A 48 8A 2A AA 68 0A 74
C758: 48 8A 2A AA 68 0A 48 8A 45
C760: 2A AA 68 0A 48 8A 2A AA 4F
C768: 68 0A 48 8A 2A AA 68 85 70
C770: FB 86 FC 98 A2 00 0A 48 7D
C778: 8A 2A AA 68 0A 48 8A 2A 47
C780: AA 68 0A 48 8A 2A AA 68 AD
C788: 18 65 FB 85 FB 8A 65 FC 70
C790: 69 04 85 FC 68 A8 28 90 4A
```

```
C798: 06 68 B1 FB 4C BE C7 68 EF
C7A0: 91 FB A5 FC 38 E9 04 85 7C
C7A8: FC A5 FB E9 00 85 FB A5 58
C7B0: FC 18 69 D8 85 FC AD B8 F0
C7B8: 02 91 FB AD A7 02 AE A8 F6
C7C0: 02 AC A9 02 60 07 77 07 01
C7C8: 77 05 47 04 B4 04 70 07 C0
C7D0: 77 05 47 04 B4 04 70 07 C8
C7D8: 77 05 47 04 B4 05 98 04 F6
C7E0: 30 04 70 04 70 07 77 22 9A
C7E8: C5 C7 00 FF 01 00 00 01 78
C7F0: FF 00 90 12 20 48 49 20 65
C7F8: 92 20 20 20 20 20 20 20 6C
C800: 20 12 20 4D 45 4E 20 92 E5
C808: 20 23 20 23 20 23 20 23 15
C810: 20 12 20 53 43 4F 52 45 DF
C818: 20 92 00 9B 20 20 4D 41 35
C820: 5A 45 20 4E 4F 2E 3E 20 0A
C828: 20 20 20 20 20 20 20 20 29
C830: 20 20 20 20 20 20 9A 12 9D
C838: 20 47 45 4D 53 20 92 00 38
C840: 9B 12 3A 3A 3A 3A 3A 3A 4B
C848: 3A 3A 3A 3A 3A 3A 3A 92 72
C850: 11 9D 9D 9D 9D 9D 9D 9D B0
C858: 9D 9D 9D 9D 9D 9D 12 3A 56
C860: 92 A0 4D 49 4E 4F 54 41 5D
C868: 55 52 A0 1F 23 9B 12 3A DA
C870: 92 11 9D 9D 9D 9D 9D 9D C5
C878: 9D 9D 9D 9D 9D 9D 9D 12 D9
C880: 3A 92 A0 A0 A0 A0 A0 A0 11
C888: A0 A0 A0 A0 A0 12 3A 92 8A
C890: 11 9D 9D 9D 9D 9D 9D 9D F0
C898: 9D 9D 9D 9D 9D 9D 12 3A 96
C8A0: 92 A0 24 A0 4D 41 5A 45 C6
C8A8: A0 24 A0 95 25 9B 12 3A B0
C8B0: 92 11 9D 9D 9D 9D 9D 9D 06
C8B8: 9D 9D 9D 9D 9D 9D 9D 12 1A
C8C0: 3A 3A 3A 3A 3A 3A 3A 3A 92
C8C8: 3A 3A 3A 3A 3A 92 00 02 80
C8D0: 06 03 0C 04 08 0A 00 01 FC
C8D8: 10 28 44 82 FE 82 82 82 5E
C8E0: 05 FE 80 80 F8 80 80 80 60
C8E8: FE 09 10 10 10 10 10 10 51
C8F0: 10 10 0D 82 C6 AA 92 82 27
C8F8: 82 82 82 0E 82 C2 A2 92 09
C900: 8A 86 82 82 0F FE 82 82 29
C908: 82 82 82 82 FE 12 FE 82 A4
C910: 82 FE 90 88 84 82 14 FE C4
C918: 10 10 10 10 10 10 10 15 9D
C920: 82 82 82 82 82 82 82 FE B0
C928: 1A FE 02 04 08 10 20 40 BF
C930: FE 23 18 18 3C 5A 99 24 D6
C938: 42 C3 24 00 00 3C 42 AD 8E
C940: 42 24 18 25 99 7E 3C 7E B6
C948: FF 99 24 66 BA 9F 00 00 C6
C950: F9 F9 00 00 9F 00 7E 00 62
```

MICRO CITY
FROM PAGE 76

```
·5 POKE56,48:POKE52,48                                          JC
·10 POKE53276,255:POKE53277,255:POKE53271
 ,255:POKE53285,8                                               IE
·90 IFPEEK(2039)=68THEN5100                                     DO
·100 GOSUB800:POKE53280,6:PRINT"[HOME][DO
 WN][DOWN][11" "]ONE MOMENT PLEASE"                             BD
·105 FORT=12288TO12927:READT2:POKET,T2:NE
 XT                                                             FH
·110 GOSUB850:FORT=192TO200:FORT2=0TO60ST
 EP3:FORT3=0TO2:T4=PEEK(T*64+T2+T3)                             FP
·120 T5=(T4AND3)*64+(T4AND12)*4+(T4AND48)
 /4+(T4AND192)/64                                               LG
·130 POKE(T+16)*64+T2+2-T3,T5:NEXT:NEXT:N
 EXT                                                            CJ
·200 GOSUB870:POKE56334,PEEK(56334)AND254
 :POKE1,PEEK(1)AND251                                           KC
·205 FORT=0TO511:POKE14336+T,PEEK(53248+T
 ):NEXT                                                         NF
·210 FORT=0TO7:READT2:FORT3=0TO7:POKE(35+
 T)*8+14336+T3,PEEK(53248+T2*8+T3)                              GI
·215 NEXT:NEXT                                                  EF
·220 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
 4)OR1                                                          IE
·230 POKE2039,68:GOTO5100                                       JM
·800 POKE53281,1:POKE53280,2:POKE53272,21
 :POKE53265,27:POKE53269,0                                      PA
·810 PRINT"[CLEAR][RED][7"[DOWN]"][c M][s
  M]  [s N][c G][s P][c Y] [s N][c Y][s M
 ] [s O][s M] [s N][c Y][s M][5" "][s N][
 c Y][s M] [s P][c Y] [c Y][s P][c Y][c Y
 ] [s M][s N]"                                                  IN
·820 PRINT"[c M] [s M][s N] [c G][c M]  [
 c G][3" "][s L][s N] [c G] [c M][5" "][c
  G][3" "][c M][3" "][c M][3" "][c M]"                          DA
·830 PRINT"[c M][4" "][c G][s @][c P] [s
 M][c P][s N] [c G][s M] [s M][c P][s N][
 5" "][s M][c P][s N] [s @][c P]  [c M][3
 " "][c M]"                                                     BM
·840 RETURN                                                     IM
·850 PRINT"[HOME][11"[DOWN]"][BLUE][12" "
 ]CAN YOU SAVE IT?"                                             JD
·860 RETURN                                                     IM
·870 PRINT"[HOME][c 4][16"[DOWN]"][BLACK]
 [15" "]WRITTEN BY"                                             IH
·880 PRINT"[DOWN][10" "]DARRYL DION HAWKI
 NS"                                                            OI
·890 RETURN                                                     IM
·1000 REM ** PLAYER CONTROL **                                  FP
·1010 JY=PEEK(56320):ONMOGOTO1110,1160                          HB
```

```
·1020 IF(JYAND16)=0THEN1410                                    OB
·1030 ON15-(JYAND15)GOTO1310,1360,,1260,1
 260,1260,,1250,1250,1250                                      DO
·1040 POKE2040,192:DK=0:SD=0:GOTO1290                          AI
·1100 REM ** MOVE RIGHT **                                     CI
·1110 X=X+SP:IFX>MX-8THENMO=2:AN=AN+1:GOT
 O1160                                                         EO
·1120 HB=HBORINT(X/256):POKE53248,XAND255
 :POKE53264,HB                                                 KH
·1130 AN=(AN+1)AND3:POKE2040,192+AN:GOTO1
 200                                                           KE
·1150 REM ** MOVE LEFT **                                      MH
·1160 X=X-SP:IFX<24THENX=24:MO=0:GOTO1290                      ND
·1170 HB=HBAND254ORINT(X/256):POKE53248,X
 AND255:POKE53264,HB                                           FL
·1180 AN=(AN-1)AND3:POKE2040,192+AN:GOTO1
 200                                                           GL
·1200 ON15-(JYAND15)GOTO1310,1360                              BF
·1210 GOTO1290                                                 GD
·1250 MO=1:GOTO1110                                            AH
·1260 MO=2:GOTO1160                                            BN
·1290 IFH1>0THENH1=H1-1                                        BE
·1295 GOTO3030                                                 FF
·1300 REM ** HIGH PUNCH **                                     AF
·1310 IFH1THEN1290                                             IG
·1315 POKE2040,196:H1=2:AN=0:IFX<MX-32ORM
 D=1THEN1290                                                   HN
·1320 GOSUB4100:P2=P2-2:GOTO1290                               NL
·1350 REM ** LOW PUNCH **                                      MD
·1360 IFH1THEN1290                                             IG
·1365 POKE2040,197:H1=4:AN=0:IFX<MX-32THE
 N1290                                                         CG
·1370 IFMA=0THENIFMD=0THENGOSUB4100:GOTO1
 290                                                           DN
·1380 GOSUB4100:P2=P2-3:H2=H2+INT(RND(1)*
 2):GOTO1290                                                   JD
·1400 REM ** DUCK **                                           GI
·1410 IF(JYAND2)=0ANDQD<8THENPOKE2040,201
 :DK=1:SD=1:QD=QD+1:P1=P1+1:GOTO1290                           DC
·1420 DK=1:POKE2040,198:GOTO1290                               GB
·2000 REM ** COMPUTER CONTROL **                               NO
·2010 MD=0:ONM2GOTO2110                                        DG
·2020 IFMX>X+MS*4THEN2250                                      MG
·2030 ONSKGOTO2560,2450,2500                                   FN
·2040 POKE2041,208:MD=0:GOTO2290                               FE
·2100 REM ** MOVE LEFT **                                      MH
·2110 MX=MX-MS:IFMX<X+8THENM2=0:MX=MX+MS:
 GOTO2290                                                      PJ
·2120 HB=HBAND253OR2*INT(MX/256):POKE5325
 0,MXAND255:POKE53264,HB                                       OB
·2130 MA=(MA+1)AND3:POKE2041,208+MA:GOTO2
 290                                                           DO
·2250 M2=1:GOTO2110                                            OD
·2290 IFH2>0THENH2=H2-1                                        BN
·2295 GOTO3050                                                 FL
·2300 REM ** HIGH PUNCH **                                     AF
·2310 IFH2THEN2290                                             IG
·2315 POKE2041,212:H2=2:MA=0:IFMX>X+32ORD
 K=1THEN2290                                                   PF
·2320 GOSUB4100:P1=P1-2:H1=H1+INT(RND(1)*
 2):GOTO2290                                                   KL
·2350 REM ** LOW PUNCH **                                      MD
·2360 IFH2THEN2290                                             IG
·2365 POKE2041,213:H2=4:MA=0:IFMX>X+32ORS
 D=1THEN2290                                                   CI
·2370 IFAN=0THENIFDK=0THENGOSUB4100:GOTO2
 290                                                           GH
·2380 GOSUB4100:P1=P1-3:H1=H1+INT(RND(1)*
 3):GOTO2290                                                   KJ
·2400 REM ** DUCK **                                           GI
·2410 MD=1:POKE2041,214:GOTO2290                               CE
·2450 IFMX>X+4THEN2480                                         JD
·2460 IFINT(RND(1)*3)=0THEN2410                                EN
·2470 R=INT(RND(1)*2)+1:ONRGOTO2310,2360                       AF
·2480 R=INT(RND(1)*7)+1:ONRGOTO2250,2250,
 2410,2310,2360                                                DL
·2490 GOTO2040                                                 FL
·2500 IFMX>X+36THEN2540                                        AO
·2510 IFINT(RND(1)*6)=0THEN2410                                BG
·2520 IFDK=0THEN2310                                           HB
·2530 GOTO2360                                                 FM
·2540 R=INT(RND(1)*6)+1:ONRGOTO2250,2410,
 2410                                                          AO
·2550 GOTO2040                                                 FL
·2560 IFMX>X+4THENR=INT(RND(1)*5)+1:ONRGO
 TO2310,2360,2410,2040,2040                                    NL
·2570 R=INT(RND(1)*8)+1:ONRGOTO2250,2250,
 2250,2410,2310,2360                                           KN
·2580 GOTO2040                                                 FL
·3000 REM ** MAIN LOOP **                                      JI
·3010 FORQ=0TO4                                                JK
·3020 GOTO1010                                                 FB
·3030 IFP2<1THEN4010                                           HL
·3040 GOTO2010                                                 FG
·3050 IFP1<1THEN4000                                           HL
·3060 IFMO=0THENIFDK=0THENPOKE2040,192                         JA
·3070 IFM2=0THENIFMD=0THENPOKE2041,208                         GM
·3080 IFAN=0THENMO=0                                           CD
·3090 IFMA=0THENM2=0                                           BH
·3100 FORD=0TO100:NEXT:NEXT:GOSUB3550                          EK
·3110 GOTO3010                                                 EP
·3300 REM ** DISPLAY **                                        FE
·3310 POKE53265,PEEK(53265)AND239:POKE204
 0,192:POKE2041,208                                            JN
·3320 X=60:MX=280:HB=2:H1=0:H2=0:QD=0                          FC
·3330 POKE53248,X:POKE53249,160:POKE53250
 ,MXAND255:POKE53251,160:POKE53264,HB                          FJ
·3340 GOSUB7000                                                FA
·3350 POKE53269,3:POKE53265,PEEK(53265)AN
 D239OR16:GOTO3010                                             NA
·3500 ONSCGOTO3610,3620,3630,3640,3650,36
 70,3680,3690                                                  HM
·3510 POKE53280,6:POKE53281,1:POKE53272,2
 1:POKE53265,27:POKE53269,0                                    CE
·3520 PRINT"[CLEAR][BLUE][7"[DOWN]"][6" "
 ][s M][s N] [s N][c Y][s M] [c G] [c M][
```

```
4" "][c M][4" "][c G][s P][c Y][c M][s M
] [c G]"                                    NE
·3530 PRINT"[6" "][c M] [c G] [c M] [c G
] [c M][4" "][c M] [s N][s M] [c G][c M]
 [c M] [s M] [c G]"                         AI
·3540 PRINT"[6" "][c M] [s M][c P][s N]
[s M][c P][s N][4" "][c M][s N] [s M][c
 G][s @][c P][c M] [s M][c G]"              CP
·3545 FORT=0TO3000:NEXT:GOTO5100            HJ
·3550 PRINT"[HOME][23"[DOWN]"]";SC,RIGHT$
(" "+STR$(P1),2),,                          DA
·3555 PRINTRIGHT$(" "+STR$(P2),2)           PM
·3560 RETURN                                IM
·3610 SK=1:MS=8:P1=20:P2=25:GOTO3300        PB
·3620 SK=1:MS=12:P1=20:P2=30:GOTO3300       JM
·3630 SK=1:MS=12:P1=20:P2=35:GOTO3300       LN
·3640 SK=2:MS=8:P1=20:P2=25:GOTO3300        FO
·3650 SK=2:MS=12:P1=20:P2=25:GOTO3300       PJ
·3660 SK=2:MS=12:P1=20:P2=30:GOTO3300       JH
·3670 SK=3:MS=8:P1=20:P2=20:GOTO3300        MO
·3680 SK=3:MS=12:P1=20:P2=25:GOTO3300       CE
·3690 SK=3:MS=12:P1=20:P2=30:GOTO3300       CO
·4000 POKE2040,200:FORT=0TO2000:NEXT:GOTO
5100                                        OC
·4010 POKE2041,216:FORT=0TO2000:NEXT:SC=S
C+1:GOTO3500                                MB
·4100 POKE54277,0:POKE54278,240:POKE54273
,100:POKE54296,15:POKE54276,129             JN
·4110 POKE54276,128:RETURN                  DB
·5000 REM                                   JD
·5010 JY=PEEK(56320):SC=-((JYAND3)=3)-3*(
(JYAND1)=0)-6*((JYAND2)=0)                  JM
·5020 SP=8:EN=1:GOTO3500                    NP
·5100 GOSUB800:PRINT"[HOME][c 4][DOWN][DO
WN][11" "]PRESS FIRE BUTTON"                HA
·5110 FORT=0TO200:IF(PEEK(56320)AND16)=0T
HEN5010                                     CD
·5120 NEXT:GOSUB850:FORT=0TO300:IF(PEEK(5
6320)AND16)=0THEN5010:NEXT                  AP
·5130 NEXT:GOSUB870:FORT=0TO300:IF(PEEK(5
6320)AND16)=0THEN5010                       JM
·5140 NEXT:PRINT"[CLEAR]":FORT=0TO750:NEX
T:GOTO5100                                  HD
·7000 POKE53280,0:POKE53281,11:POKE53282,
15                                          OP
·7005 POKE53265,PEEK(53265)OR64:POKE53272
,PEEK(53272)OR14                            LJ
·7010 PRINT"[CLEAR][5" "][RED]&#[22" "][B
LUE]&[6"'"][c T][3"[SS]"]";                 DM
·7015 PRINT"[RED][4"'"]*'[BLUE][22"'"]*[6
"'"][c T][4"[SS]"]";                        JF
·7020 PRINT"[c 2][3"'"][RED]'*'[BLUE][9"'
"][c 2][4"'"][BLUE][4"'"][WHITE][4"'"][B
LUE]'*[5"'"][c T][5"[SS]"]";                OB
·7025 PRINT"[c 2][3"'"][RED]'*'[BLUE][9"'
"][c 2][4"'"][BLUE][4"'"][WHITE][4"'"][B
LUE]'*[4"'"][c T][5"[SS]"][c 5][c +]";      FO
·7030 PRINT"[c 2][3"'"][RED]'*'[BLUE][9"'
"][c 2][4"'"][BLUE][4"'"][WHITE][4"'"][B
LUE]'*[3"'"][c T][5"[SS]"][c 5][c +]#";     JG
·7035 PRINT"[c 2]''([RED]'*'[BLUE][9"'"][
c 2][3"'"]([BLUE][9"'"]*''[c T][5"[SS]"]
[c 5][c +]# ";                              FF
·7040 PRINT"[c 2][3"'"][RED]'*'[BLUE][9"'
"][c 2][4"'"][BLUE][9"'"]*'[c T][5"[SS]"
][c 5][c +]#  ";                            KB
·7045 PRINT"[c 2][3"'"][RED]'*[c T][BLUE]
[9"'"][c 2][4"'"][BLUE][9"'"]*[c T][5"[S
S]"][c 5][c +]#[3" "]";                     PP
·7050 PRINT"[34"[SS]"][c 5][c +]#[4" "]";   CD
·7055 PRINT"[33"[SS]"][c 5][c +]#[5" "]";   OG
·7060 PRINT"[32"[SS]"][c 5][c +]#[6" "]";   IP
·7065 PRINT"[c 5][31"'"]*#[7" "]";          CJ
·7070 RETURN                                IM
·8192 DATA 0,0,0,0,0,0,0,0,0,0,252,0,3,21
2,0,3                                       LK
·8208 DATA 84,0,3,84,0,0,80,0,6,169,16,22
,169,16,22,169                              GO
·8224 DATA 80,5,88,64,3,252,0,2,168,0,10,
170,0,10,10,128                             DF
·8240 DATA 40,2,160,40,0,160,160,2,128,22
4,2,128,60,3,240,0                          NJ
·8256 DATA 0,0,0,0,0,0,0,0,0,0,252,0,3,21
2,0,3                                       LK
·8272 DATA 84,0,3,84,0,0,80,0,6,169,16,22
,169,16,22,169                              GO
·8288 DATA 80,5,168,64,1,124,0,2,168,0,2,
184,0,0,174,0                               LI
·8304 DATA 0,46,0,0,174,0,3,184,0,0,248,0
,0,63,0,0                                   GJ
·8320 DATA 0,0,0,0,0,0,0,0,0,0,252,0,3,21
2,0,3                                       LK
·8336 DATA 84,0,3,84,0,0,80,0,6,169,16,22
,169,16,22,169                              GO
·8352 DATA 80,5,88,64,3,252,0,2,184,0,2,1
72,0,0,168,0                                JF
·8368 DATA 2,234,0,10,138,0,42,2,128,56,2
,128,15,3,240,0                             BD
·8384 DATA 0,0,0,0,0,0,0,0,0,0,252,0,3,21
2,0,3                                       LK
·8400 DATA 84,0,3,84,0,0,80,0,6,169,16,22
,169,16,22,169                              GO
·8416 DATA 80,5,168,64,1,124,0,2,168,0,2,
186,0,2,186,0                               ON
·8432 DATA 0,174,0,15,172,0,15,160,0,12,1
60,0,0,252,0,0                              ID
·8448 DATA 0,0,0,0,0,0,0,0,0,0,252,0,3,21
2,0,3                                       LK
·8464 DATA 84,1,3,84,21,0,81,84,0,149,64,
2,148,0,6,168                               AA
·8480 DATA 0,6,168,0,3,252,0,2,168,0,10,1
70,0,10,10,128                              EJ
·8496 DATA 42,2,160,40,0,160,168,2,128,22
4,2,128,60,3,240,0                          JL
·8512 DATA 0,0,0,0,0,0,0,0,0,0,252,0,3,21
2,0,3                                       LK
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 117 and 118 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·8528 DATA 84,0,3,84,0,0,80,0,0,148,0,2,1
49,80,6,169                                BF
·8544 DATA 85,6,168,5,3,252,0,2,168,0,10,
170,0,10,10,128                            CD
·8560 DATA 42,2,160,40,0,160,168,2,128,22
4,2,128,60,3,240,0                         JL
·8576 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0                                          FG
·8592 DATA 0,0,0,0,0,0,0,0,0,0,252,0,3,212,
0,3,84                                     FH
·8608 DATA 0,3,84,0,0,80,0,6,169,16,6,89,
16,1,108,64                                LH
·8624 DATA 3,250,128,2,170,160,2,128,160,
234,128,160,250,0,252,0                    MG
·8640 DATA 0,0,0,0,8,0,2,170,0,0,8,0,2,0,
0,170                                      OG
·8656 DATA 128,0,2,0,32,0,10,168,0,0,32,0
,0,0,0,0                                   AP
·8672 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0                                          FG
·8688 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0                                          FG
·8704 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0                                          FG
·8720 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0                                          FG
·8736 DATA 0,63,0,0,245,0,0,213,0,0,213,0
,0,20,0,0                                  DA
·8752 DATA 168,10,0,154,42,128,155,170,12
8,30,162,128,26,131,240,0                  KJ
·8768 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0                                          FG
·8784 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0                                          FG
·8800 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,23
4,60                                       NF
·8816 DATA 10,233,95,10,229,87,2,182,23,2
34,133,0,250,1,80,0                        FN
·9000 DATA105,95,223,233,160,209,229,234   ED
```

SPEEDWAY
FROM PAGE 78

Starting address in hex: C000
Ending address in hex: C8B0
SYS to start: 49152

***Flankspeed* required for entry! See page 119.**

```
C000: A0 00 B9 B1 C5 99 00 3C A7
C008: B9 B1 C6 99 00 3D B9 B1 7C
C010: C7 99 00 3E C8 D0 EB 8C C1
C018: 17 D0 8C 25 D0 8C 20 D0 FF
C020: A9 03 8D 1C D0 A9 0E 8D 8C
C028: 26 D0 A0 18 B9 9E C4 99 8E
C030: 00 D4 88 10 F7 78 A9 7F 37
C038: 8D 0D DC A9 01 8D 1A D0 D2
C040: 8D 12 D0 A9 1B 8D 11 D0 E4
C048: A9 07 8D 14 03 A9 C4 8D 99
C050: 15 03 58 A9 00 8D 15 D0 DD
C058: 8D B7 C4 A9 C8 A0 C4 20 5A
C060: 1E AB A2 08 A0 0C 18 20 B9
C068: F0 FF A2 33 A9 3B A0 C5 7A
C070: 20 FD C2 84 A8 A2 04 A0 C5
C078: 1D A9 01 99 B8 D9 8A 09 FF
C080: 30 99 B8 05 38 98 E9 07 C9
C088: A8 CA D0 ED A0 03 98 09 FF
C090: F8 99 FA 07 A9 05 99 29 96
C098: D0 98 0A AA A9 96 9D 05 99
C0A0: D0 B9 67 C4 9D 04 D0 88 52
C0A8: 10 E4 A9 3C 8D 1D D0 8D 8C
C0B0: 15 D0 A2 10 A0 08 18 20 2A
C0B8: F0 FF A2 35 A9 4B A0 C5 DC
C0C0: 20 FD C2 84 04 A2 12 A0 7F
C0C8: 02 18 20 F0 FF A2 34 A9 74
C0D0: 8E A0 C5 20 FD C2 84 05 30
C0D8: B9 90 C4 8D B7 C4 B9 93 3F
C0E0: C4 85 03 A9 93 20 D2 FF 5E
C0E8: A5 04 09 F8 85 FB A9 00 BF
C0F0: 85 FC 85 FD 85 39 A9 04 63
C0F8: 85 FE A9 D8 85 3A A0 06 66
C100: 06 FB 26 FC 88 D0 F9 A2 1B
C108: 00 B1 FB 85 02 06 02 90 D5
C110: 15 98 48 8A 0A A8 A9 A0 8D
C118: 91 FD C8 91 FD A5 03 91 3A
C120: 39 88 91 39 68 A8 E8 E0 87
C128: 14 90 16 18 A5 FD 69 28 30
C130: 85 FD 85 39 A5 FE 69 00 80
C138: 85 FE 18 69 D4 85 3A A2 75
C140: 00 8A 29 07 D0 C7 C8 C0 1D
C148: 3F 90 BE A6 A8 BD 70 C4 19
C150: 8D 15 D0 A9 F1 9D F8 07 FC
C158: 9D C4 C4 A9 07 9D C0 C4 53
C160: A9 FF 9D BE C4 A9 00 9D 72
C168: C2 C4 9D BA C4 A4 05 B9 70
C170: 8D C4 9D C6 C4 A9 AB 9D DE
C178: BC C4 BD 7B C4 9D 27 D0 8D
C180: 8D 86 02 86 02 BC 7D C4 1E
C188: 88 A2 16 18 20 F0 FF A9 9C
C190: 62 A0 C5 20 1E AB A6 02 EB
C198: CA 10 B8 A4 04 B9 83 C4 D6
C1A0: 8D B8 C4 38 E9 10 8D B9 25
C1A8: C4 A2 16 A0 13 18 20 F0 03
C1B0: FF A9 6F A0 C5 20 1E AB 1A
C1B8: A0 05 20 29 C3 A2 20 8E BC
C1C0: 12 D4 E8 8E 12 D4 CE D5 AA
C1C8: 07 88 D0 EE 98 A0 02 99 EC
C1D0: 08 DC 88 10 FA AD 1E D0 E5
C1D8: AD 1F D0 20 CE C2 E6 A7 B6
C1E0: A6 A8 A5 A7 29 0F D0 16 9C
C1E8: BD 00 DC 29 0C 4A 4A A8 F5
C1F0: 18 BD F8 07 79 6B C4 29 99
```

```
C1F8: 07 09 F0 9D F8 07 BD F8 4E
C200: 07 DD C4 C4 F0 18 BC C6 FA
C208: C4 D0 0B 9D C4 C4 A4 05 79
C210: B9 8D C4 9D C6 C4 DE C6 EA
C218: C4 A0 01 4C 28 C2 BD 00 73
C220: DC 29 10 4A 4A 4A 4A A8 08
C228: 18 BD BE C4 79 77 C4 85 BC
C230: 02 BD C0 C4 79 79 C4 30 5D
C238: 0C C9 08 B0 1B 9D C0 C4 05
C240: A5 02 9D BE C4 BC C0 C4 4B
C248: A5 A7 39 6F C4 D0 09 BD 9A
C250: C4 C4 29 07 A8 20 93 C2 29
C258: BC C0 C4 B9 96 C4 BC 8B F7
C260: C4 99 00 D4 A9 41 99 04 1C
C268: D4 20 8B C3 CA 30 03 4C F6
C270: E2 C1 AD 1F D0 29 03 F0 CF
C278: 03 20 73 C3 AD 1E D0 29 98
C280: 03 F0 03 20 33 C3 A2 04 35
C288: A0 00 C8 D0 FD CA D0 FA 57
C290: 4C DB C1 18 BD B8 C4 79 47
C298: 4F C4 C9 31 90 2F C9 C8 F9
C2A0: B0 2B 9D B8 C4 18 BD BC 2A
C2A8: C4 79 57 C4 85 02 BD BA 03
C2B0: C4 79 5F C4 A8 D0 07 A5 39
C2B8: 02 C9 18 B0 09 60 A5 02 5E
C2C0: C9 41 B0 09 A0 01 9D BC 81
C2C8: C4 98 9D BA C4 60 AD 0A 5B
C2D0: DC 29 0F A8 09 30 8D D2 28
C2D8: 07 AD 09 DC AA 29 F0 4A 82
C2E0: 4A 4A 4A 09 30 8D D4 07 62
C2E8: 8A 29 0F 09 30 8D D5 07 4F
C2F0: 98 F0 09 A5 A8 D0 05 68 10
C2F8: 68 4C DC C3 60 86 02 20 57
C300: 1E AB A2 00 86 C6 E8 8E 31
C308: 86 02 A5 A2 29 10 85 C7 5F
C310: A9 7D A0 C5 20 1E AB 20 A7
C318: E4 FF C9 31 90 EC C5 02 3D
C320: B0 E8 20 D2 FF 38 E9 31 FF
C328: A8 18 A5 A2 69 28 C5 A2 2B
C330: D0 FC 60 A0 00 38 AD BC A1
C338: C4 ED BD C4 85 02 AD BA 5D
C340: C4 ED BB C4 05 02 90 02 0D
C348: A0 02 AD B8 C4 CD B9 C4 62
C350: 90 01 C8 B9 87 C4 48 A8 A1
C358: A2 00 20 93 C2 18 68 69 5B
C360: 04 29 07 A8 E8 20 93 C2 9C
C368: A2 80 8E 12 D4 E8 8E 12 8A
C370: D4 A9 03 85 02 A2 00 46 62
C378: 02 90 0A A9 05 DD C0 C4 27
C380: 90 03 9D C0 C4 E8 E0 02 03
C388: 90 ED 60 A0 00 BD B8 C4 43
C390: 10 02 A0 02 BD BC C4 C9 4E
C398: A9 90 05 C9 AD B0 01 C8 C9
C3A0: BD C2 C4 19 7F C4 C9 0F 1C
C3A8: 90 2C C0 03 D0 28 BC 7D 5C
C3B0: C4 18 B9 C1 07 69 01 C9 44
C3B8: 3A 90 0B 18 B9 C0 07 69 91
C3C0: 01 99 C0 07 A9 30 99 C1 58
C3C8: 07 B9 C0 07 C9 32 90 04 E1
C3D0: A5 A8 D0 06 A9 00 9D C2 FF
C3D8: C4 60 68 68 A9 40 8D 04 4A
C3E0: D4 8D 0B D4 A9 00 85 C6 19
C3E8: A2 16 A0 0E 18 20 F0 FF 79
C3F0: A9 81 A0 C5 20 1E AB 20 8C
C3F8: 29 C3 EE 86 02 20 E4 FF 62
C400: C9 0D D0 E4 4C 53 C0 A9 96
C408: 01 8D 19 D0 A2 D6 AC B7 5E
C410: C4 AD 12 D0 10 26 AD BB 05
C418: C4 0A 0D BA C4 8D 10 D0 E1
C420: AD BC C4 8D 00 D0 AD BD 19
C428: C4 8D 02 D0 AD B8 C4 8D 06
C430: 01 D0 AD B9 C4 8D 03 D0 8F
C438: A2 01 A0 00 8E 12 D0 8C 7A
C440: 21 D0 AD 0D DC 29 01 F0 E4
C448: 03 4C 31 EA 4C BC FE FF BB
C450: 00 01 01 01 00 FF FF FF 53
C458: FF FF 00 01 01 01 00 FF 5B
C460: FF FF 00 00 00 00 00 47 A7
C468: 80 B9 F2 00 FF 01 00 00 96
C470: 01 03 07 0F 1F 3F 7F EB 54
C478: 0A FF 00 02 07 05 21 01 B2
C480: 02 04 08 BC AE B3 BF 00 6D
C488: 02 06 04 00 07 0B 1F 3D 03
C490: 0B 0C 0F 05 09 01 FF DB A1
C498: B7 93 6F 4B 27 03 51 01 1B
C4A0: 20 44 00 00 F8 51 01 20 70
C4A8: 44 00 00 F8 EF 0E 00 00 E3
C4B0: 00 09 00 00 38 03 1F 00 14
C4B8: 00 00 00 00 00 00 00 00 B8
C4C0: 00 00 00 00 00 00 00 00 C0
C4C8: 93 8E 08 0D 0D 0D 0D 20 47
C4D0: 20 20 9F D5 C3 C9 20 B2 E6
C4D8: C3 C9 20 B2 C3 AE 20 B2 7E
C4E0: C3 AE 20 B2 C3 C9 20 B2 86
C4E8: 20 20 B2 20 D5 C3 C9 20 7F
C4F0: B2 20 B2 0D 20 20 20 CA AE
C4F8: C3 C9 20 AB C3 CB 20 AB AD
C500: B3 20 20 AB B3 20 20 C2 56
C508: 20 C2 20 C2 D5 C9 C2 20 50
C510: AB C3 B3 20 CA B2 CB 0D A9
C518: 20 20 20 CA C3 CB 20 B1 A4
C520: 20 20 20 B1 C3 BD 20 B1 85
C528: C3 BD 20 B1 C3 CB 20 CA F5
C530: CB CA CB 20 B1 20 B1 20 56
C538: 20 B1 00 81 50 4C 41 59 C2
C540: 45 52 53 20 28 31 2F 32 06
C548: 29 20 00 9C 53 45 4C 45 58
C550: 43 54 20 41 20 54 52 41 51
C558: 43 4B 20 28 31 2D 34 29 EA
C560: 20 00 4C 41 50 53 11 11 D3
C568: 9D 9D 9D 05 30 30 00 9A 41
C570: 45 54 11 11 9D 9D 9D 05 0A
C578: 30 3A 30 35 00 20 9D 92 98
C580: 00 50 52 45 53 53 20 52 81
C588: 45 54 55 52 4E 00 9A 52 05
C590: 4F 41 44 20 43 4F 4E 44 AA
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 117 and 118 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
C598: 49 54 49 4F 4E 20 20 31 8E
C5A0: 3D 44 52 59 20 32 3D 57 B4
C5A8: 45 54 20 33 3D 49 43 59 B8
C5B0: 20 00 00 00 00 00 00 00 D0
C5B8: 00 00 54 10 00 69 54 00 DA
C5C0: 6A 55 00 67 D5 00 1D F4 CF
C5C8: 10 1F 69 54 57 A6 95 55 9E
C5D0: 9F A5 15 9F 94 04 6E 90 62
C5D8: 00 6A 40 01 59 00 01 54 33
C5E0: 00 00 54 00 00 10 00 00 45
C5E8: 00 00 00 00 00 00 00 00 E8
C5F0: 00 00 00 00 00 00 00 00 F0
C5F8: 00 00 00 00 00 00 00 00 F8
C600: 15 40 55 15 40 55 15 71 DB
C608: 55 05 D6 A4 1B FA 79 69 D6
C610: D9 F9 69 D9 F9 1B FA 79 B0
C618: 05 D6 A4 15 71 55 15 40 C9
C620: 55 15 40 55 00 00 00 00 20
C628: 00 00 00 00 00 00 00 00 28
C630: 00 00 00 00 00 00 00 00 30
C638: 00 00 00 10 00 00 54 00 9C
C640: 01 54 00 01 59 00 00 6A 5A
C648: 40 04 6E 90 15 9F 94 55 2A
C650: 9F A5 57 A6 95 1F 69 54 06
C658: 1D F4 10 67 D5 00 6A 55 77
C660: 00 69 54 00 54 10 00 00 82
C668: 00 00 00 00 00 00 00 00 68
C670: 00 00 00 00 00 00 00 00 70
C678: 00 00 00 54 00 14 54 50 85
C680: 15 A9 50 16 FE 50 16 FE 0A
C688: 50 16 76 50 16 9A 50 01 B7
C690: A9 00 01 99 00 0D DD C0 80
C698: 17 FF 50 15 DD 50 15 DD 36
C6A0: 50 15 A9 50 15 A9 50 00 0F
C6A8: 64 00 00 10 00 00 00 00 1D
C6B0: D9 00 00 00 00 00 00 00 8A
C6B8: 00 00 00 04 00 00 15 00 D1
C6C0: 00 15 40 00 65 40 01 A9 66
C6C8: 00 06 B9 10 16 F6 54 5A 54
C6D0: F6 55 56 9A D5 15 69 F4 57
C6D8: 04 1F 74 00 57 D9 00 55 F6
C6E0: A9 00 15 69 00 04 15 00 22
C6E8: 00 00 00 00 00 00 00 00 E8
C6F0: 00 00 00 00 00 00 00 00 F0
C6F8: 00 00 00 00 00 00 00 00 F8
C700: 55 01 54 55 01 54 55 4D F7
C708: 54 1A 97 50 6D AF E4 6F CF
C710: 67 69 6F 67 69 6D AF E4 23
C718: 1A 97 50 55 4D 54 55 01 67
C720: 54 55 01 54 00 00 00 00 1F
C728: 00 00 00 00 00 00 00 00 28
C730: 00 00 00 00 00 00 00 00 30
C738: 00 00 00 04 15 00 15 69 CF
C740: 00 55 A9 00 57 D9 04 1F 93
C748: 74 15 69 F4 56 9A D5 5A 51
C750: F6 55 16 F6 54 06 B9 10 CD
C758: 01 A9 00 00 65 40 00 15 BD
C760: 40 00 15 00 00 04 00 00 B9
C768: 00 00 00 00 00 00 00 00 68
C770: 00 00 00 00 00 10 00 00 80
C778: 64 00 15 A9 50 15 A9 50 FA
C780: 15 DD 50 15 DD 50 17 FF 1E
C788: 50 0D DD C0 01 99 00 01 20
C790: A9 00 16 9A 50 16 76 50 18
C798: 16 FE 50 16 FE 50 15 A9 22
C7A0: 50 14 54 50 00 54 00 00 FD
C7A8: 00 00 00 00 00 00 00 00 A8
C7B0: D9 FF FF F0 E0 00 70 C0 8D
C7B8: 00 30 C0 00 30 80 00 10 6A
C7C0: 80 00 10 80 F0 10 81 F8 4D
C7C8: 10 83 FC 10 83 FC 10 83 7D
C7D0: FC 10 83 FC 10 83 FC 10 FE
C7D8: 81 F8 10 80 F0 10 80 00 65
C7E0: 10 80 00 10 C0 00 30 C0 33
C7E8: 00 30 E0 00 70 FF FF F0 5B
C7F0: 12 FF FF F0 E1 F8 70 C0 FF
C7F8: F0 30 C0 00 30 80 00 10 9B
C800: 80 00 10 80 00 10 84 02 A7
C808: 10 87 0E 10 87 FE 10 87 DB
C810: FE 10 87 FE 10 87 0E 10 5B
C818: 84 02 10 80 00 10 80 00 BF
C820: 10 80 00 10 C0 00 30 C0 72
C828: F0 30 E1 F8 70 FF FF F0 85
C830: 12 FF FF F0 E0 F0 70 C0 36
C838: 60 30 C0 60 30 80 00 10 AA
C840: 80 00 10 80 00 10 80 00 E1
C848: 10 83 0C 10 87 9E 10 87 B5
C850: FE 10 87 FE 10 83 FC 10 86
C858: 80 00 10 80 00 10 80 00 F9
C860: 10 80 00 10 C0 00 30 C0 B2
C868: 00 30 E0 00 70 FF FF F0 DA
C870: 12 FF FF F0 C0 00 30 80 E4
C878: 00 10 80 00 10 80 00 10 A9
C880: 87 FE 10 81 F8 10 80 F0 13
C888: 10 C0 60 30 F0 60 F0 F8 25
C890: 61 F0 F0 60 F0 C0 60 30 76
C898: 80 F0 10 81 F8 10 87 FE 2B
C8A0: 10 80 00 10 80 00 10 80 52
C8A8: 00 10 C0 00 30 FF FF F0 9A
C8B0: 12 C2
```

THE EDITOR
FROM PAGE 56

```
·10 REM *** THE EDITOR *** BUCK CHILDRESS
  ***                                      HO
·20 REM *** P.O. BOX 13575 SALEM, OR 9730
 9 ***                                     DC
```

```
·30 PRINTCHR$(147)"LOADING AND CHECKING D
 ATA LINE:":J=49152:L=100:C=11                KI
·40 PRINTCHR$(19)TAB(31)L:PRINT:FORB=0TOC
 :READA:IFA<0ORA>255THEN60                    KG
·50 POKEJ+B,A:X=X+A:NEXTB:READA:IFA=XTHEN
 70                                           BP
·60 PRINT"ERROR IN DATA LINE:"L:END           HL
·70 X=0:J=J+12:L=L+10:IFL<471THEN40           AI
·80 PRINT"THE DATA IS OK AND LOADED[3"."]
 ":PRINT                                      GI
·90 PRINT"SYS 49152 TO ACTIVATE[3"."]":EN
 D                                            EG
·100 DATA120,169,22,162,192,141,20,3,142,
 21,3,169,1164                                HH
·110 DATA15,162,3,141,24,212,134,252,88,9
 6,173,190,1490                               AO
·120 DATA193,174,141,2,142,190,193,224,4,
 240,64,224,1791                              JJ
·130 DATA5,208,11,205,190,193,240,52,32,1
 77,193,76,1582                               KK
·140 DATA81,192,166,253,164,203,132,253,1
 92,64,240,36,1976                            ND
·150 DATA228,203,208,17,173,139,2,197,251
 ,240,25,133,1816                             BF
·160 DATA251,198,252,16,19,169,3,133,252,
 169,0,162,1624                               FN
·170 DATA23,157,0,212,202,16,250,169,65,1
 41,4,212,1451                                DB
·180 DATA76,49,234,166,203,228,254,240,24
 7,134,254,224,2309                           EE
·190 DATA64,240,241,165,207,141,189,193,1
 64,211,177,209,2201                          KD
·200 DATA141,187,193,140,188,193,165,209,
 133,65,165,210,1989                          GL
·210 DATA133,66,169,32,224,47,208,8,145,2
 09,136,16,1393                               ON
·220 DATA251,76,151,193,224,44,208,12,145
 ,209,200,196,1909                            OD
·230 DATA213,144,249,240,247,76,151,193,2
 24,4,208,19,1968                             FJ
·240 DATA196,213,176,165,162,9,202,240,5,
 200,196,213,1977                             EE
·250 DATA144,248,132,211,76,151,193,224,5
 ,208,15,192,1799                             GC
·260 DATA0,240,69,162,9,202,240,238,136,2
 40,235,76,1847                               EB
·270 DATA197,192,224,3,208,5,160,0,76,182
 ,192,224,1663                                DN
·280 DATA6,208,30,164,213,177,209,201,32,
 208,19,136,1603                              JA
·290 DATA48,15,177,209,201,32,240,247,201
 ,160,208,5,1743                              JJ
·300 DATA173,189,193,208,238,200,76,182,1
 92,224,1,208,2084                            PH
·310 DATA20,166,214,160,6,232,224,24,144,
 5,240,25,1460                                NB
·320 DATA76,81,192,136,208,243,76,137,193
 ,224,7,208,1781                              NO
·330 DATA15,166,214,240,239,160,6,202,240
 ,107,136,208,1933                            HO
·340 DATA250,76,137,193,224,33,208,88,166
 ,214,224,24,1837                             DE
·350 DATA176,218,232,224,25,176,213,181,2
 17,16,247,142,2067                           EB
·360 DATA192,193,181,217,41,128,157,194,1
 93,232,224,24,1976                           LL
·370 DATA144,244,240,242,32,101,233,174,1
 92,193,181,217,2193                          DG
·380 DATA41,15,73,128,149,217,172,192,193
 ,200,185,217,1782                            BA
·390 DATA0,41,15,24,125,194,193,153,217,0
 ,232,200,1394                                DE
·400 DATA224,24,144,238,240,236,165,242,4
 1,15,73,128,1770                             IP
·410 DATA133,242,160,0,132,211,32,124,232
 ,76,151,193,1686                             EM
·420 DATA224,28,240,3,76,81,192,162,24,16
 5,211,201,1607                               CB
·430 DATA40,144,3,56,233,40,168,24,32,240
 ,255,32,1267                                 DL
·440 DATA177,193,132,198,132,207,173,187,
 193,174,189,193,2148                         PA
·450 DATA240,3,56,233,128,172,188,193,145
 ,65,76,81,1580                               CO
·460 DATA192,160,0,132,199,132,212,132,21
 6,96,0,255,1726                              HD
·470 DATA255,1,2,3,4,5,6,0,0,0,0,0,276        NP
```

LAZY SOURCE CODE FROM PAGE 88

Starting address in hex: C000
Ending address in hex: C7CF
SYS to start: 49152

***Flankspeed* required for entry! See page 119.**

```
C000: 78 A9 3F 8D 14 03 A9 C0 70
C008: 8D 15 03 A9 21 8D 02 03 0B
C010: A9 C0 8D 03 03 58 60 C9 90
C018: 30 D0 03 20 B3 C7 4C 31 35
C020: EA A9 CE 85 5F A9 C3 85 5B
C028: 60 A9 C6 85 5A A9 C6 85 CE
C030: 5B A9 F8 85 58 A9 06 85 41
C038: 59 20 BF A3 4C 83 A4 A5 2F
C040: C5 C9 40 D0 06 8D EA C2 22
C048: 4C AF C0 CD EA C2 F0 5F D0
C050: 8D EA C2 A2 03 DD EB C2 BD
C058: F0 05 CA 10 F8 30 50 8A 2D
C060: AE 8D 02 E0 01 D0 10 C9 2B
C068: 03 F0 3B C9 02 F0 34 C9 52
C070: 01 F0 2A C9 00 F0 29 C9 3A
C078: 03 F0 0C C9 02 F0 1B C9 1A
C080: 01 F0 14 C9 00 F0 0D A9 F7
C088: 7B 8D 23 C1 A9 11 8D 1C DA
C090: C1 4C 02 C1 4C 88 C1 4C 45
```

```
C098: 95 C1 4C 9C C1 4C B7 C1 60
C0A0: 4C A3 C1 4C CB C1 4C D8 51
C0A8: C1 20 9F FF 4C 02 C1 4C 86
C0B0: 17 C0 A2 03 B9 00 C7 20 CF
C0B8: E4 C0 C8 CA D0 F6 A9 20 83
C0C0: 8D 77 02 A9 14 8D 78 02 8D
C0C8: 20 F2 C0 4C 31 EA C9 22 F0
C0D0: F0 06 20 EF C0 4C 31 EA 01
C0D8: 4C C7 C6 28 50 78 A0 C8 0E
C0E0: F0 18 40 68 8D 79 02 EE 8A
C0E8: E5 C0 EE EE C0 60 02 20 B0
C0F0: E4 C0 A9 79 8D E5 C0 AD 9B
C0F8: EE C0 85 C6 A9 02 8D EE 1D
C100: C0 60 A5 C5 C9 40 D0 06 6D
C108: 8D EA C2 4C 2B C1 CD EA 35
C110: C2 F0 18 8D EA C2 A2 12 CB
C118: DD F8 C2 F0 11 CA 10 F8 87
C120: 30 5F BC 7B C3 4C B2 C0 6B
C128: 4C 31 EA 4C A9 C0 E0 05 2D
C130: 10 03 4C 22 C1 E0 10 F0 55
C138: 08 A9 20 8D 77 02 4C 6D CA
C140: C1 A9 83 A0 C3 20 1E AB 7D
C148: A9 00 85 C6 A9 31 8D 14 BA
C150: 03 A9 EA 8D 15 03 A9 83 BA
C158: 8D 02 03 A9 A4 8D 03 03 CC
C160: A9 57 8D 24 03 A9 F1 8D 3F
C168: 25 03 4C 31 EA A9 14 8D 44
C170: 78 02 4C 78 C1 EE EE C0 10
C178: BD A3 C7 4C CE C0 4C 31 FA
C180: EA A9 00 85 C6 4C 9F C2 10
C188: A9 48 8D 23 C1 A9 05 8D 29
C190: 1C C1 4C 02 C1 A9 59 8D 0F
C198: 23 C1 10 F1 A9 6A 8D 23 44
C1A0: C1 10 EA AD 21 D0 18 69 7E
C1A8: 01 C9 10 F0 06 8D 21 D0 F9
C1B0: 4C 31 EA A9 00 90 F6 AD F7
C1B8: 20 D0 18 69 01 C9 10 F0 F6
C1C0: 06 8D 20 D0 4C 31 EA A9 57
C1C8: 00 90 F6 A9 1C 20 FC C1 F4
C1D0: 8D 3F C3 A2 FF 6C 00 03 73
C1D8: A9 08 20 FC C1 8D 43 C3 FD
C1E0: 8D 3D C3 A5 15 8D 42 C3 BD
C1E8: 8D 3C C3 A9 1F 8D 24 03 F3
C1F0: A9 C2 8D 25 03 A9 00 8D 4A
C1F8: 45 C3 10 D7 A0 C3 20 1E 8C
C200: AB 20 60 A5 86 7A 84 7B D2
C208: 20 73 00 AA F0 F3 A2 FF CD
C210: 86 3A 90 01 18 20 6B A9 AF
C218: A9 00 85 C6 A5 14 60 20 48
C220: 57 F1 08 85 FB C9 0D F0 BA
C228: 04 A5 FB 28 60 E6 C8 A5 AB
C230: C8 CD 45 C3 F0 04 30 02 F6
C238: 10 0C A9 57 8D 24 03 A9 B3
C240: F1 8D 25 03 B0 E3 A9 0D 33
C248: 20 D2 FF A5 D6 8D 3B C3 44
C250: CE 3B C3 86 FC 84 FD AD D1
C258: 42 C3 AE 43 C3 85 62 86 82
C260: 63 8E 3D C3 8D 3C C3 A2 83
C268: 90 38 20 49 BC 20 DD BD 13
C270: A2 00 BD 01 01 F0 07 9D 68
C278: 77 02 E8 4C 72 C2 E8 86 CB
C280: C6 8E 45 C3 AD 43 C3 18 AB
C288: 6D 3F C3 8D 43 C3 90 03 21
C290: EE 42 C3 A6 FC A4 FD AD 79
C298: 3B C3 85 D6 4C 29 C2 AD D9
C2A0: AD C3 C9 99 D0 06 20 BF 2C
C2A8: C2 4C AF C2 20 D4 C2 A2 84
C2B0: 00 BD AD C3 E8 9D D0 06 3D
C2B8: E0 20 D0 F5 4C 31 EA A2 8B
C2C0: 00 BD AD C3 29 7F 9D AD E3
C2C8: C3 A9 1E 9D F8 06 E8 E0 BA
C2D0: 20 D0 EE 60 A2 00 BD AD 1F
C2D8: C3 18 69 80 9D AD C3 A9 57
C2E0: 20 9D F8 06 E8 E0 20 D0 58
C2E8: ED 60 3E 04 05 06 03 23 AA
C2F0: 24 22 22 22 22 22 22 22 04
C2F8: 0A 1C 14 12 0E 15 1A 1D 9F
C300: 21 22 25 2A 24 27 26 29 2D
C308: 3E 57 48 41 54 27 53 20 16
C310: 59 4F 55 52 20 4C 49 4E 64
C318: 45 23 3A 00 0D 4A 55 4D B4
C320: 50 20 42 45 54 57 45 45 4E
C328: 4E 20 4C 49 4E 45 53 20 33
C330: 28 31 2D 32 35 35 29 3A B6
C338: 20 00 20 17 00 55 09 0C F9
C340: 04 40 00 61 0D 03 02 F0 E8
C348: 00 03 06 09 0C 0F 12 15 9C
C350: 18 1B 1E 21 24 27 2A 2D 65
C358: 30 33 36 39 3C 3F 42 45 2E
C360: 48 4B 4E 51 54 57 5A 5D F6
C368: 60 63 66 69 6C 6F 72 75 BF
C370: 78 7B 7E 81 84 87 8A 8D 88
C378: 90 93 96 99 9C 9F A2 A5 51
C380: 14 20 14 12 4F 4B 21 21 B7
C388: 20 49 27 4D 20 44 45 41 51
C390: 44 21 53 59 53 20 34 39 83
C398: 31 35 32 20 54 4F 20 52 67
C3A0: 45 56 49 56 45 20 4D 45 D3
C3A8: 21 21 00 20 20 19 0F 15 68
C3B0: 20 08 09 14 20 14 08 05 37
C3B8: 20 17 12 0F 0E 07 20 0B 51
C3C0: 05 19 2D 14 12 19 20 01 6C
C3C8: 07 01 09 0E 21 2D 20 20 76
C3D0: 20 86 B1 20 20 20 20 20 C9
C3D8: 20 20 20 86 B3 20 20 20 D3
C3E0: 20 20 20 20 86 B5 20 20 DD
C3E8: 20 20 20 20 86 B7 20 20 E7
C3F0: 20 20 20 20 20 20 81 2D 60
C3F8: 20 01 04 03 2D 2D 2D 2D D5
C400: 2D 2D 2D 03 0D 10 2D 2D 02
C408: 81 2D 2D 2D 0F 12 01 2D 60
C410: 2D 2D 2D 2D 14 01 19 2D 20
C418: 2D 2D 2D 2D 2D 81 82 2D 2B
C420: 20 01 0E 04 2D 2D 2D 2D 08
C428: 2D 2D 2D 03 10 18 2D 2D 35
C430: 82 2D 2D 2D 10 08 01 2D 80
```

```
C438: 2D 2D 2D 2D 14 13 18 2D 59
C440: 2D 2D 2D 2D 2D 82 83 2D 55
C448: 20 01 13 0C 2D 2D 2D 2D 3D
C450: 2D 2D 2D 03 10 19 2D 2D 5E
C458: 83 2D 2D 2D 10 08 10 2D B8
C460: 2D 2D 2D 2D 14 18 01 2D 6F
C468: 2D 2D 2D 2D 2D 83 84 2D 7F
C470: 20 02 03 03 2D 2D 2D 2D 4D
C478: 2D 2D 2D 04 05 03 2D 2D 66
C480: 84 2D 2D 2D 10 0C 01 2D D6
C488: 2D 2D 2D 2D 14 18 13 2D A9
C490: 2D 2D 2D 2D 2D 84 85 2D A9
C498: 20 02 03 13 2D 2D 2D 2D 85
C4A0: 2D 2D 2D 04 05 18 2D 2D A3
C4A8: 85 2D 2D 2D 10 0C 10 2D 0F
C4B0: 2D 2D 2D 2D 14 19 01 2D C0
C4B8: 2D 2D 2D 2D 2D 85 86 2D D3
C4C0: 20 02 05 11 2D 2D 2D 2D AD
C4C8: 2D 2D 2D 04 05 19 2D 2D CC
C4D0: 86 2D 2D 2D 12 0F 0C 2D 39
C4D8: 2D 2D 2D 2D 23 20 2D 2D 2B
C4E0: 2D 2D 2D 2D 2D 86 87 2D FD
C4E8: 20 02 09 14 2D 2D 2D 2D DC
C4F0: 2D 2D 2D 05 0F 12 2D 2D F8
C4F8: 87 2D 2D 2D 12 0F 12 2D 68
C500: 2D 2D 2D 2D 24 20 2D 2D 53
C508: 2D 2D 2D 2D 2D 87 88 2D 27
C510: 20 02 0D 09 2D 2D 2D 2D FC
C518: 2D 2D 2D 09 0E 03 2D 2D 14
C520: 88 2D 2D 2D 12 14 09 2D 8C
C528: 13 05 0E 04 20 06 05 05 82
C530: 04 02 01 03 0B 88 89 2D 84
C538: 20 02 0E 05 2D 2D 2D 2D 22
C540: 2D 2D 2D 09 0E 18 2D 2D 51
C548: 89 2D 2D 2D 12 14 13 2D BF
C550: 20 20 14 0F 20 14 08 09 F8
C558: 13 20 20 20 20 89 8A 2D 2D
C560: 20 02 10 0C 2D 2D 2D 2D 53
C568: 2D 2D 2D 09 0E 19 2D 2D 7A
C570: 8A 2D 2D 2D 13 02 03 2D C7
C578: 20 20 01 04 04 12 05 13 EB
C580: 13 3A 20 20 20 8A 8B 2D 71
C588: 20 02 12 0B 2D 2D 2D 2D 7C
C590: 2D 2D 2D 0A 0D 10 2D 2D 99
C598: 8B 2D 2D 2D 13 05 03 2D F3
C5A0: 20 0D 2E 20 02 05 0E 0E 3F
C5A8: 05 14 14 20 20 8B 8C 2D 5B
C5B0: 20 02 16 03 2D 2D 2D 2D A0
C5B8: 2D 2D 2D 0A 13 12 2D 2D C9
C5C0: 8C 2D 2D 2D 13 05 04 2D 1E
C5C8: 03 2F 0F 20 07 01 12 07 4B
C5D0: 0F 19 0C 05 13 8C 8D 2D 64
C5D8: 20 02 16 13 2D 2D 2D 2D D8
C5E0: 2D 2D 2D 0C 04 01 2D 2D D3
C5E8: 8D 2D 2D 2D 13 05 09 2D 4C
C5F0: 20 20 20 20 13 14 15 04 B1
C5F8: 09 0F 20 20 20 8D 8E 2D BA
C600: 20 03 0C 03 2D 2D 2D 2D E6
C608: 2D 2D 2D 0C 04 18 2D 2D 12
C610: 8E 2D 2D 2D 13 14 01 2D 7B
C618: 32 32 31 20 32 31 13 14 58
C620: 20 13 14 2E 20 8E 8F 2D 01
C628: 20 03 0C 04 2D 2D 2D 2D 10
C630: 2D 2D 2D 0C 04 19 2D 2D 3B
C638: 8F 2D 2D 2D 13 14 18 2D BB
C640: 02 12 0F 0F 0B 0C 19 0E B0
C648: 2C 0E 2E 19 2E 8F 90 2D 45
C650: 20 03 0C 09 2D 2D 2D 2D 3D
C658: 2D 2D 2D 0C 13 12 2D 2D 6B
C660: 90 2D 2D 2D 13 14 19 2D E5
C668: 20 20 31 31 32 33 32 20 C2
C670: 20 20 20 20 20 90 91 2D 60
C678: 20 03 0C 16 2D 2D 2D 2D 72
C680: 2D 2D 2D 0E 0F 10 2D 2D 8F
C688: 91 2D 2D 2D 14 01 18 2D FB
C690: 2D 2D 2D 2D 84 89 85 A1 7A
C698: A1 2D 2D 2D 2D 91 20 20 C0
C6A0: 20 86 B1 20 20 20 20 20 99
C6A8: 20 20 20 86 B3 20 20 20 A3
C6B0: 20 20 20 20 86 B5 20 20 AD
C6B8: 20 20 20 20 86 B7 20 20 B7
C6C0: 20 20 20 20 20 20 FF BD 3F
C6C8: D4 C0 8D D9 C6 E0 0D 30 AA
C6D0: 03 EE DA C6 A0 00 A2 0D B4
C6D8: B9 68 C5 C9 20 F0 15 C9 7A
C6E0: 00 F0 0E C9 21 30 07 C9 CB
C6E8: 40 10 03 4C F1 C6 18 69 C2
C6F0: 40 20 E4 C0 C8 CA D0 E0 3C
C6F8: A9 C5 8D DA C6 4C C8 C0 6D
C700: 41 44 43 41 4E 44 41 53 31
C708: 4C 42 43 43 42 43 53 42 38
C710: 45 51 42 49 54 42 4D 49 5F
C718: 42 4E 45 42 50 4C 42 52 61
C720: 4B 42 56 43 42 56 53 43 76
C728: 4C 43 43 4C 44 43 4C 49 64
C730: 43 4C 56 43 4D 50 43 50 8A
C738: 58 43 50 59 44 45 43 44 8E
C740: 45 58 44 45 59 45 4F 52 A7
C748: 49 4E 43 49 4E 58 49 4E AA
C750: 59 4A 4D 50 4A 53 52 4C CD
C758: 44 41 4C 44 58 4C 44 59 B0
C760: 4C 53 52 4E 4F 50 4F 52 E1
C768: 41 50 48 41 50 48 50 50 BC
C770: 4C 41 50 4C 50 52 4F 4C D8
C778: 52 4F 52 52 54 49 52 54 03
C780: 53 53 42 43 53 45 43 53 DB
C788: 45 44 53 45 49 53 54 41 DC
C790: 53 54 58 53 54 59 54 41 27
C798: 58 54 41 59 54 53 58 54 34
C7A0: 58 41 54 58 53 54 59 41 29
C7A8: 23 24 22 22 22 22 22 22 BC
C7B0: 22 22 22 A9 00 85 5F A9 4F
C7B8: 04 85 60 A9 F8 85 5A A9 CE
C7C0: 06 85 5B A9 C6 85 58 A9 9F
C7C8: C6 85 59 20 BF A3 60 00 52
```

REBELS AND LORDS
FROM PAGE 96

```
·10 POKE52,128:POKE56,128:CLR:GOTO1260                    CP
·20 PRINTD$B$;:PRINTD$DD$B$;:PRINTD$DD$DD
 $B$;:PRINTD$DD$DD$DD$B$;D$;:RETURN                       ND
·30 GOSUB50:GOTO70                                        CH
·40 POKEP1,F1:POKEG1,W1:POKEG1,W1-W:RETUR
 N                                                        BO
·50 POKEP1,F2:POKEG1,W3:POKEG1,W3-W:RETUR
 N                                                        GL
·60 POKEP1,F3:POKEG1,W1:POKEG1,W1-W:RETUR
 N                                                        OA
·70 GOSUB90:IFJF=OANDJD=OTHEN70                           LO
·80 RETURN                                                IM
·90 JD=O:JF=T1-(PEEK(T2)ANDT1):IFJF=OTHEN
 110                                                      KP
·100 JD=JFANDT5:JF=JFANDT6:RETURN                         AE
·110 GETA$:IFA$=""THENRETURN                              HI
·120 JF=M:FORJD=WTO5:IFA$=MID$(O$,JD,W)TH
 ENJF=BI(JD-W)                                            OF
·130 NEXT:JF=JF-(JF<O):GOTO100                            HB
·140 POKEZY+II*SN,OY+8*R:POKEZX+II*SN,FNL
 (OX+T6*C):J=BI(SN)                                       LB
·150 POKET9,PEEK(T9)ORJ:IFFNH(OX+T6*C)>OT
 HENPOKET4,PEEK(T4)ORJ:RETURN                             BF
·160 J=T7-J:POKET4,PEEK(T4)ANDJ:RETURN                    II
·170 IFA=OTHENA=W:NA=W:RETURN                             EN
·180 NA=O:FORI=ATOWSTEPM:IFA(I,O)<OTHENNA
 =I                                                       OJ
·190 NEXT:IFNA>OTHENRETURN                                AB
·200 IFA<199THENA=A+W:NA=A:RETURN                         JN
·210 GOSUB20:PRINT"[RED] TOO MANY ARMIES-
 HIT F7/FB";:GOSUB40:GOSUB70                              IO
·220 RETURN                                               IM
·230 PRINT"[CLEAR]";:FORR=OTO18:PRINTLEFT
 $(D$,R+3);                                               BA
·240 FORC=OTO18:K=M(R,C):IFK<3THENPRINTM$
 (K);:GOTO260                                             LJ
·250 K=F(K-T3,O):PRINTCL$(K);FT$;                         KD
·260 NEXT:NEXT:IFA<WTHEN310                               KA
·270 FORI=WTOA:IFA(I,O)<OTHEN300                          PG
·280 R=A(I,II):C=A(I,T3):K=M(R,C):IFK=WOR
 K>IITHEN300                                              PD
·290 K=A(I,O):PRINTLEFT$(D$,R+T3);LEFT$(R
 $,II*C+II);CL$(K);AR$;                                   HC
·300 NEXT                                                 IA
·310 RETURN                                               IM
·320 GOSUB90:IFJD=OTHEN380                                KA
·330 IFJDANDWTHENR=(R-W)-(R<W):GOTO370                    KP
·340 IFJDANDIITHENR=(R+W)+(R>17):GOTO370                  IM
·350 IFJDAND4THENC=(C-W)-(C<W):GOTO370                    IP
·360 C=(C+W)+(C>17)                                       NP
·370 GOSUB140                                             CN
·380 RETURN                                               IM
·390 K=O:AA(O,O)=M:AA(O,W)=O:IFM(R,C)<T3T
 HEN410                                                   DI
·400 J=M(R,C)-T3:AA(O,O)=J:AA(O,W)=F(J,W)                 GK
·410 IFA<WTHEN470                                         CD
·420 FORJ=WTOA:IF(A(J,O)=M)ORA(J,O)<>OW(R
 ,C)THEN460                                               CH
·430 IF(R<>A(J,II))OR(C<>A(J,T3))THEN460                  GK
·440 IFA(J,4)<OTHENAA(O,O)=J:AA(O,W)=A(J,
 W):GOTO460                                               FP
·450 K=K+W+(K>9):AA(K,O)=J:AA(K,W)=A(J,W)                 PF
·460 NEXT                                                 IA
·470 MA=AA(O,W):RETURN                                    MK
·480 R=10:C=10:SN=O:GOSUB140:PRINTBB$H$"
 "PN$(PP);" HIT FB/F7, WEEK"WK;:GOSUB30                   MA
·490 IFPEEK(T0)AND6THENGOSUB20:POKET9,O:R
 ETURN                                                    HF
·500 PRINTBB$H$" "PN$(PP)" LOOKING (COMO
 TO END)";:GOSUB770                                       LP
·510 GOSUB320:IFJD=OANDJF=OAND((PEEK(T0)A
 ND6)=O)THEN510                                           KL
·520 IFPEEK(T0)AND6THENGOSUB20:POKET9,O:R
 ETURN                                                    HF
·530 IFJD<>OTHEN500                                       BA
·540 IFMA=OTHEN570                                        GN
·550 GOSUB1210:IFMS=OTHEN490                              PC
·560 GOSUB1020:GOTO490                                    LI
·570 PRINTBB$H$" NO TROOPS AVAILABLE-HIT
 FB/F7";:GOSUB40:GOSUB70:GOTO490                          IP
·580 QA=O:UA=M:FORI=OTONF-W:IFF(I,W)>QATH
 ENQA=F(I,W):UA=I                                         AB
·590 NEXT:UD=M:QD=O:IFA<WTHEN620                          DI
·600 FORI=WTOA:IFA(I,O)=MTHEN620                          PI
·610 IFA(I,W)>QDTHENUD=I:QD=A(I,W)                        EF
·620 NEXT                                                 IA
·630 IFUA=MORQD=>QATHEN670                                NL
·640 IFF(UA,O)<>OTHEN670                                  HB
·650 GOSUB170:A(NA,O)=O:A(NA,W)=INT(.34*Q
 A):F(UA,W)=F(UA,W)-A(NA,W)                               GB
·660 A(NA,II)=F(UA,II):A(NA,T3)=F(UA,T3):
 A(NA,4)=M                                                NG
·670 IFA=OTHENRETURN                                      AI
·680 FORI=WTOA:IFA(I,O)<>OORA(I,4)<>MTHEN
 760                                                      NB
·690 UA=A(I,II):UD=A(I,T3):R=M:C=O                        JD
·700 FORJ=OTONF-W:IFR>MORF(J,O)>OORRND(W)
 >.6THEN730                                               PP
·710 IFABS(F(J,II)-UA)>T3THEN730                          HD
·720 IFABS(F(J,T3)-UD)<=T3THENR=F(J,II):C
 =F(J,T3)                                                 EH
·730 NEXT:IFR>MTHEN750                                    PI
·740 R=INT(19*RND(2)):C=INT(19*RND(W)):IF
 (R=UA)AND(C=UD)THEN740                                   JB
·750 A(I,4)=R:A(I,FI)=C                                   FK
·760 NEXT:RETURN                                          EJ
```

```
·770 GOSUB20:MA=0:L=M(R,C):IFL<T3THENONL+WGOTO830,890,940 AI
·780 IFOW(R,C)<>PPTHENK=OW(R,C):PRINTPN$(OW(R,C))"'S CASTLE";:GOTO950 LI
·790 PRINTCL$(PP);"[3" "]YOUR CASTLE";:GOSUB390 PM
·800 PRINTD$DD$"TROOPS:";MA;" LEVY:"F(L-T3,4);:IFMA>OTHENPRINTS$; BB
·810 IFK>OTHENGOSUB960 LJ
·820 GOTO950 CO
·830 IFOW(R,C)=MTHENPRINT"[YELLOW]GRASS";:GOTO950 KO
·840 IFOW(R,C)<>PPTHENK=OW(R,C):PRINTPN$(K)"'S TROOPS";:GOTO950 JN
·850 PRINTCL$(PP);"[3" "]YOUR TROOPS";:GOSUB390:PRINTD$DD$"IDLE TROOPS:";MA; EO
·860 IFMA>OTHENPRINTS$; PN
·870 IFK>OTHENGOSUB960 LJ
·880 GOTO950 CO
·890 IFOW(R,C)<>PPTHENPRINT"[c 2]WOODS";:GOTO950 NB
·900 PRINT"[c 2][3" "]YOUR TROOPS, HIDDEN";:GOSUB390:PRINTD$DD$"IDLE TROOPS:";MA; NA
·910 IFMA>OTHENPRINTS$; PN
·920 IFK>OTHENGOSUB960 LJ
·930 GOTO950 CO
·940 PRINT"[c 4]ROCKS"; EF
·950 RETURN IM
·960 PRINTD$DD$DD$;:IFK<4THENFORI=WTOK:GOSUB1000:NEXT:RETURN LJ
·970 FORI=WTOT3:GOSUB1000:NEXT:PRINTD$DD$DD$DD$; OE
·980 IFK<7THENFORI=4TOK:GOSUB1000:NEXT:RETURN EM
·990 FORI=4TO6:GOSUB1000:NEXT:RETURN LF
·1000 PRINTRIGHT$("[5" "]"+STR$(AA(I,O)),FI)+AR$+MID$(STR$(AA(I,W))+"[4" "]",II,4); LF
·1010 RETURN IM
·1020 GOSUB170:IFNA=OTHENRETURN LE
·1030 A(NA,O)=PP:A(NA,W)=MS:A(NA,II)=RS:A(NA,T3)=CS:A(NA,4)=RD:A(NA,FI)=CD GP
·1040 GOSUB20:PRINT"ARMY "NA" OF"MS" SENT"; GP
·1050 PRINTBB$H$" HIT FB/F7 TO CONTINUE";:GOSUB30 JC
·1060 GOSUB20:PRINTBB$;:IFM(RS,CS)>IITHEN1090 GI
·1070 NK=AA(O,O):A(NK,W)=A(NK,W)-MS:IFA(NK,W)<WTHENA(NK,O)=M NG
·1080 RETURN IM
·1090 NK=AA(O,O):F(NK,W)=F(NK,W)-MS:IFF(NK,W)<OTHENF(NK,W)=O CK
·1100 RETURN IM
·1110 PRINTBB$H$"JS/CRSR SIZE, O CANCEL, FB/F7 END";:GOSUB20:PRINT"TROOPS 1-";MA; EM
·1120 MS=O:GOTO1200 CJ
·1130 GOSUB90:IF(JF=O)AND(JD=O)THEN1130 BP
·1140 IFJF>OTHENRETURN FF
·1150 IFJDANDWTHENMS=MS+10:GOTO1190 II
·1160 IFJDANDIITHENMS=MS-10:GOTO1190 GN
·1170 IFJDAND4THENMS=MS-W:GOTO1190 DE
·1180 IFJDAND8THENMS=MS+W ME
·1190 MS=-MS*(MS=>O):IFMS>MATHENMS=MA KG
·1200 PRINTD$DD$B$D$DD$" SENDING"MS;:GOTO1130 MM
·1210 GOSUB1110:IFMS=OTHENRETURN BL
·1220 RS=R:CS=C:SN=W:GOSUB140:GOSUB20 NE
·1230 PRINTBB$H$"POSITION X WITH JS/CRSR, THEN F7/FB"; EM
·1240 GOSUB320:IFJF=OTHEN1240 MN
·1250 RD=R:CD=C:SN=O:POKET9,PEEK(T9)ANDW:C=CS:R=RS:GOSUB140:RETURN IP
·1260 GOSUB2560:GOSUB2190 OK
·1270 FORI=OTONP:DP(I)=O:NEXT HE
·1280 IFNP>WTHEN1300 KA
·1290 PP=W:GOSUB480:GOSUB580:GOSUB1390:GOTO1370 JC
·1300 FORI=WTONP:PP(I)=I:NEXT:FORI=WTONP:FORJ=WTONP-W:IFRND(O)>.5THEN1320 LM
·1310 QA=PP(J):PP(J)=PP(J+W):PP(J+W)=QA GO
·1320 NEXT:NEXT:PP(O)=W MK
·1330 PP=PP(PP(O)):IFDP(PP)>OTHEN1350 PM
·1340 GOSUB480 CM
·1350 PP(O)=PP(O)+1:IFPP(O)<=NPTHEN1330 FP
·1360 GOSUB580:GOSUB1390 MI
·1370 GOSUB2730:IFNP-DP(O)>OTHEN1280 OH
·1380 GOSUB2990:STOP DA
·1390 PRINTH$B$H$" HIT FB/F7 TO SEE ARMIES MARCH";:GOSUB60:GOSUB70 PJ
·1400 FORTQ=WTOFI:GOSUB1410:NEXT:WK=WK+W:RETURN CA
·1410 IFA=OTHENRETURN ON
·1420 NS=INT(W+A*RND(-TI)):NA=NS HJ
·1430 PRINTBB$H$" [BLACK]MOVING ARMIES[3"."]";:NA=W-(NA<A)*NA PN
·1440 IFA(NA,O)<OORA(NA,4)=MTHEN1960 BE
·1450 RS=A(NA,II):CS=A(NA,T3):RD=A(NA,4):CD=A(NA,FI):PP=A(NA,O) IG
·1460 IF(RS=RD)OR(CS=CD)THENRD=RS+SGN(RD-RS):CD=CS+SGN(CD-CS):GOTO1490 EG
·1470 IFRND(2)<.5THENRD=RS+SGN(RD-RS):CD=CS:GOTO1490 OL
·1480 RD=RS:CD=CS+SGN(CD-CS) FA
·1490 IFM(RD,CD)=2THENGOSUB2050:GOTO1910 FC
·1500 A(NA,II)=M:IFM(RS,CS)>WTHEN1530 HP
·1510 R=RS:C=CS:GOSUB390:IFMA>OORK>OTHEN1530 OH
·1520 OW(R,C)=M:IFM(R,C)=OTHENPRINTLEFT$(D$,R+T3)LEFT$(R$,II*C+II);BL$; FC
·1530 QA=A(NA,W):UA=QA:DP=OW(RD,CD):IFDP=MORDP=PPTHEN1890 LH
·1540 R=RD:C=CD:GOSUB390:IFMA>OORK>OTHEN1600 NB
·1550 IFM(R,C)<2THEN1890 EC
·1560 GOSUB20:PRINTPN$(PP)" TAKES "PN$(DP
```

```
FF )"'S";                                                    AF
II ·1570 PRINTD$DD$" [BLACK]WITHOUT A FIGHT!
GN ";                                                        CL
DE ·1580 SN=II:GOSUB1400:PRINTBB$H$" HIT FB/F
ME 7 TO CONTINUE";:GOSUB600                                  JB
KG ·1590 GOSUB700:PRINTBB$;:POKET9,0:GOSUB200:
   GOTO1890                                                  JG
MM ·1600 QD=MA:GA=.2+.35*RND(W):FF=.6:IFK>OT
BL HENFORI=WTOK:QD=QD+AA(I,W):NEXT                           MI
NE ·1610 GD=.25+.3*RND(2)-.1*(M(R,C)>2):UD=Q
   D:GOSUB2110                                               DE
EM ·1620 QA=INT(QA-QD*GD*FF):IFQA<OTHENQA=0                  DF
MN ·1630 GOSUB2150:IFQA=OTHEN1780                            CJ
   ·1640 QD=INT(QD-QA*GA):IFQD<OTHENQD=0                     DE
IP ·1650 GOSUB2150:IFQD>OTHENFF=W:GOTO1620                   LD
OK ·1660 IFK=OTHEN1680                                       EB
HE ·1670 FORI=WTOK:A(AA(I,0),0)=M:NEXT:GOSUB
KA 390:GOTO1660                                              NM
   ·1680 K=AA(0,0):IFM(R,C)>2THEN1710                        NM
JC ·1690 IFMA>OTHENA(K,0)=M                                  AD
   ·1700 GOTO1720                                            FH
LM ·1710 F(K,0)=PP:F(K,W)=0                                  ED
GO ·1720 OW(R,C)=PP:PRINTLEFT$(D$,R+T3)LEFT$
MK (R$,II*C+II);                                             BH
PM ·1730 IFM(R,C)=WTHENPRINTWD$;:GOTO1760                    OI
CM ·1740 IFM(R,C)>IITHENPRINTCL$(PP)FT$;:GOT
FP 01760                                                     OF
MI ·1750 PRINTCL$(PP)AR$;                                    CA
OH ·1760 GOSUB20:PRINTPN$(PP);" IS VICTORIOU
DA S!";:GOSUB1980:A(NA,W)=QA                                 AA
   ·1770 GOSUB60:GOTO1890                                    IE
PJ ·1780 GOSUB20:PRINTPN$(DP);" REPELLS INVA
   DERS!";:GOSUB40:GOSUB1980                                 LF
CA ·1790 UD=UD-QD:IFUD<=OTHEN1850                            PO
ON ·1800 IFK=00RUD<=OTHEN1850                                GO
HJ ·1810 FORI=WTOK:IFUD<=OTHEN1840                           IG
   ·1820 IFAA(I,W)>UDTHENA(AA(I,0),W)=AA(I,W
PN )-UD:UD=0:GOTO1840                                        FG
BE ·1830 A(AA(I,0),0)=M:UD=UD-AA(I,W)                        DK
   ·1840 NEXT:GOSUB390:GOTO1800                              JC
IG ·1850 IFUD<=OTHEN1880                                     EB
   ·1860 IFM(R,C)>IITHENF(AA(0,0),W)=MA-UD:G
EG OTO1880                                                   NE
   ·1870 A(AA(0,0),W)=MA-UD                                  HM
OL ·1880 A(NA,0)=M:GOTO1960                                  DH
FA ·1890 A(NA,II)=RD:A(NA,T3)=CD:R=RD:C=CD:O
FC W(R,C)=A(NA,0):GOSUB2000                                  HG
HP ·1900 IFRD<>A(NA,4)ORCD<>A(NA,FI)THEN1960                 NC
   ·1910 R=RD:C=CD:GOSUB390:IFM(R,C)>IITHEN1
OH 940                                                       DL
   ·1920 IFAA(0,0)<WTHENA(NA,4)=M:GOTO1960                   GO
FC ·1930 K=AA(0,0):A(K,W)=A(K,W)+QA:GOTO1950                 IB
   ·1940 K=AA(0,0):F(K,W)=F(K,W)+QA                          KA
LH ·1950 A(NA,0)=M                                           KH
   ·1960 IFNA<>NSTHEN1430                                    JB
NB ·1970 RETURN                                              IM
EC ·1980 POKE53280,FI:PRINTBB$H$" HIT FB/F7
   TO CONTINUE";:GOSUB70                                     MP
   ·1990 POKET9,0:PRINTBB$;:GOSUB20                          JL
   ·2000 PRINTLEFT$(D$,R+T3)LEFT$(R$,II*C+II
   );                                                        IE
   ·2010 IFM(R,C)=WTHENPRINTWD$;:GOTO2040                    ME
   ·2020 IFM(R,C)>IITHENPRINTCL$(OW(R,C))FT$
   ;:GOTO2040                                                CB
   ·2030 PRINTCL$(OW(R,C))AR$;                               HN
   ·2040 RETURN                                              IM
   ·2050 R=RS:RD=R:C=CS:CD=C:IFA(NA,0)=OTHEN
   2100                                                      JN
   ·2060 PRINTBB$H$"HIT FB/F7 TO GO ON";                     LC
   ·2070 RD=R:CD=C:IFM(R,C)=OTHENSN=0:GOSUB1
   40                                                        LC
   ·2080 GOSUB20:PRINTPN$(A(NA,0));"'S ARMY
   "NA;                                                      AE
   ·2090 PRINTD$DD$" UNABLE TO PROCEED";:GOS
   UB40:GOSUB70:POKET9,0                                     EJ
   ·2100 A(NA,4)=RS:A(NA,FI)=CS:GOSUB20:PRIN
   TBB$;:RETURN                                              JN
   ·2110 CT=II:SN=II:GOSUB140:PP=A(NA,0):DP=
   OW(R,C):POKE53280,II                                      EE
   ·2120 PRINTLEFT$(D$,R+T3)LEFT$(R$,II*C+II
   )CL$(PP)"[UPARROW]";:GOSUB20                              LM
   ·2130 PRINT" "PN$(PP)" ATTACKS "PN$(DP);:
   GOSUB2160                                                 CN
   ·2140 PRINTBB$H$" HIT FB/F7 TO SEE OUTCOM
   E";:GOSUB30:PRINTBB$;:RETURN                              MI
   ·2150 POKEP2,F1:POKEG2,W2:POKEG2,W2-W                     KA
   ·2160 CT=-(CT+W)*(CT<2):POKE53289,CT:GP=[
   PI][UPARROW]6-LOG([PI][UPARROW]3):PRINT"
   [c 2]";                                                   JH
   ·2170 PRINTD$DD$RIGHT$("[7" "]"+STR$(QA),
   9)RIGHT$("[14" "]"+STR$(QD),14);                          FN
   ·2180 RETURN                                              IM
   ·2190 PRINT"[CLEAR][BLACK]"                               AA
   ·2200 A$="":INPUT"LORDS (1-4)";A$:NP=VAL(
   A$):IFNP<WORNP>4THEN2200                                  GD
   ·2210 FORI=WTONP:PRINT"LORD"I"'S NAME";:I
   NPUTA$:PN$(I)=PN$(I)+A$:NEXT                              KI
   ·2220 A$="":PRINT"CASTLES (";T3*NP;:INPUT
   "-35)";A$                                                 JL
   ·2230 NF=VAL(A$):IFNF<T3*NPORNF>35THEN222
   0                                                         NM
   ·2240 PRINT"LEVELS: 1-NO TERRAIN":PRINT"[
   8" "]2-HILLS, SOME TREES"                                 HL
   ·2250 PRINT"[8" "]3-HILLS AND FORESTS"                    NC
   ·2260 A$="":INPUT"LEVEL (1-3)";A$:GL=VAL(
   A$):IFGL<WORGL>T3THEN2260                                 BJ
   ·2270 GOSUB2970                                           GF
   ·2280 PRINT"[CLEAR][BLUE]PLACING [YELLOW]
   GRASS &";:FORI=0TO18:FORJ=0TO18:M(I,J)=0
   :OW(I,J)=M                                                IL
   ·2290 NEXT:PRINT"&";:NEXT:IFGL=WTHEN2360                  HO
   ·2300 PRINT:PRINT"[c 4]ROCKS %";:J=6:FORI
   =0TO18:IFRND(0)<.2THEN2330                                NL
   ·2310 IFRND(T3)>.2THENM(I,J)=II                           EA
   ·2320 IFRND(W)>.2THENM(I,J+W)=II                          ON
   ·2330 J=INT(J-.2+1.9*RND(2)):IFJ<4THENJ=4                 IH
```

```
•2340 IFJ>14THENJ=14                                                PH
•2350 PRINT"%";:NEXT                                                PK
•2360 IFGL=WTHEN2400                                                KM
•2370 PRINT:PRINT"[c 2]TREES #";:FORI=WTO
50                                                                  CB
•2380 R=INT(19*RND(T3)):C=INT(19*RND(W)):
IFM(R,C)<>OTHEN2380                                                 GA
•2390 M(R,C)=W:PRINT"#";:NEXT                                       KC
•2400 IFGL<T3THEN2420                                               ME
•2410 FORKK=WTOFI:I=INT(T3+12*RND(9)):J=I
NT(T3+12*RND(6)):GOSUB2520:NEXT                                     CO
•2420 PRINT:PRINT"[RED]CASTLES []";:FORI=
OTONF-W:KK=0:IFI<NPTHENKK=I+W                                       CE
•2430 R=INT(19*RND(0)):C=INT(19*RND(T3)):
IFM(R,C)>WTHEN2430                                                  HB
•2440 F(I,II)=R:F(I,T3)=C:F(I,K)=KK:M(R,C
)=I+T3:OW(R,C)=KK:PRINT"[]";:NEXT                                   CK
•2450 GOSUB230:PRINTD$;"OK (Y/N)?";:GOSUB
50                                                                  DC
•2460 GETA$:IFA$<>"Y"ANDA$<>"N"THEN2460                             PA
•2470 IFA$="N"THEN2280                                              LI
•2480 GOSUB20:FORI=OTONF-W:IFF(I,O)>OTHEN
2500                                                                KN
•2490 K=.09+1.3*RND(W):F(I,4)=W+INT(W/K):
F(I,W)=INT(T5*RND(W)+RND(II)/K):GOTO2510                            HJ
•2500 F(I,4)=7+NP:F(I,W)=INT(20+NF+15*RND
(W)+II*(ABS(9-F(I,II))+ABS(9-F(I,T3))))                             KD
•2510 NEXT:RETURN                                                   EJ
•2520 FORR=I-4TOI+4:FORC=J-4TOJ+4:IFR<OOR
R>18ORC<OORC>18THEN2550                                             ID
•2530 IFM(R,C)<>OTHEN2550                                           BH
•2540 K=8-ABS(I-R)-ABS(J-C):IFRND(II)<K/1
0THENM(R,C)=W                                                       GH
•2550 NEXT:PRINT"#";:NEXT:RETURN                                    IG
•2560 O=0:W=1:II=2:M=-1:I=0:J=0:R=0:C=0:Q
A=0:JF=0:JD=0:K=0:KK=0:RR=0:T1=31:FI=5                              IE
•2570 T2=56320:T3=3:T4=53264:T5=15:T6=16:
T7=255:T8=256:T9=53269:T0=653:WK=1                                  MD
•2580 W2=129:W1=21:W3=17:P1=54273:F1=50:F
2=100:F3=150:G1=54276:G2=54283                                      FF
•2590 ZX=53248:ZY=ZX+1:O$="[UP][DOWN][LEF
T][RIGHT][F7]":DEFFNH(K)=INT(K/T8):DEFFN
L(K)=K-T8*FNH(K)                                                    ID
•2600 DIMM(18,18),CL$(4),BI(7),M$(2),A(19
9,5),F(35,4),OW(18,18),PN$(4)                                       BG
•2610 FORI=OTO4:CL$(I)=MID$("[BLACK][RED]
[YELLOW][CYAN][WHITE]",I+W,W):PN$(I)=CL$
(I):NEXT:PN$(0)="[BLACK]REBEL"                                      GL
•2620 B$="[38" "]":DD$="[DOWN]":BB$="[HOM
E] "+B$                                                             HC
•2630 FT$="[]":AR$="[UPARROW][BACKARROW]"
:WD$="[c 2]##":RK$="[c 4]%%":BL$="[YELLO
W]&&":H$="[HOME]":M$(O)=BL$:M$(W)=WD$                               LL
•2640 PRINT"[CLEAR][8"[DOWN]"]"SPC(11)"RE
BELS[DOWN]":PRINTSPC(14)"AND":PRINTSPC(1
4)"[DOWN]LORDS"                                                     DH
•2650 M$(II)=RK$:FORI=OTO7:BI(I)=II[UPARR
OW]I:NEXT:OX=30:OY=57:GOSUB2890                                     CI
•2660 FORI=OTOII:POKE34808+I,I:POKE53287+
I,6:NEXT:POKE53285,6:POKE53275,0                                    LC
•2670 POKE53276,0:POKE53277,0:POKET9,0:PO
KE53271,0:POKE53280,FI:POKE53281,FI                                 JG
•2680 FORI=WTO199:A(I,O)=M:NEXT:A=O:D$="[
HOME][RIGHT][20"[DOWN]"]"                                           NI
•2690 R$="[LEFT][39"[RIGHT]"]":FORI=54272
TO54300:POKEI,0                                                     AP
•2700 S$=D$+DD$+LEFT$(R$,27)+"(FB/F7-SEND
)"                                                                  GB
•2710 NEXT:POKE54296,143:POKE54287,102:PO
KE54292,240:POKE54290,17:POKE54285,246                              JH
•2720 POKE54278,250:POKE54280,50:RETURN                             FJ
•2730 FORI=OTONF-W:F(I,W)=F(I,W)+F(I,4):N
EXT                                                                 BM
•2740 GOSUB20:PRINTBB$H$" HIT FB/F7 TO GO
 ON, CTRL TO QUIT";                                                 DJ
•2750 GOSUB90:IFPEEK(T0)AND6THEN2780                                MP
•2760 IFJF=OTHEN2750                                                KL
•2770 PRINTBB$;:RETURN                                              EH
•2780 PRINTBB$H$" CRSR/JS FOR PLAYER, FB/
F7 TO QUIT"D$" -1 FOR NONE, 0 FOR ALL";                             GB
•2790 MS=M                                                          KE
•2800 PRINTD$DD$" "MS" ";:GOSUB90:IFJF>
OTHEN2830                                                           NE
•2810 IFJD>OTHENMS=MS+W:IFMS>NPTHEN2790                             GM
•2820 GOTO2800                                                      FH
•2830 IFMS=MTHENGOSUB20:PRINTBB$;:RETURN                            EE
•2840 IFMS=OTHENDP(O)=NP:PRINTBB$;:GOSUB2
0:RETURN                                                            HK
•2850 IFA>OTHENFORI=WTOA:IFA(I,O)=MSTHENA
(I,O)=O                                                             KJ
•2860 NEXT:FORI=OTONF-W:IFF(I,O)=MSTHENF(
I,O)=O                                                              GF
•2870 NEXT:FORR=OTO18:FORC=OTO18:IFOW(R,C
)=MSTHENOW(R,C)=O                                                   MB
•2880 NEXT:NEXT:GOSUB230:DP(MS)=W:DP(O)=D
P(O)+W:GOTO2740                                                     IB
•2890 BA=40960:SA=ZX:POKE56334,PEEK(56334
)AND254:POKEW,PEEK(W)AND251                                         NB
•2900 FORI=OTO1023:POKEI+BA,PEEK(I+SA):NE
XT                                                                  IO
•2910 POKEW,PEEK(W)OR4:POKE56334,PEEK(563
34)ORW                                                              BK
•2920 BA=40960:READSA:IFSA<OTHEN2940                                PF
•2930 SA=BA+8*SA:FORI=SATOSA+7:READJ:POKE
I,J:NEXT:GOTO2920                                                   HL
•2940 BA=32768:READSA:IFSA<OTHEN2960                                NK
•2950 SA=BA+64*SA:FORI=SATOSA+63:READJ:PO
KEI,J:NEXT:GOTO2940                                                 MA
•2960 RETURN                                                        IM
•2970 POKE56578,PEEK(56578)ORT3:POKE56576
,(PEEK(56576)AND252)ORW                                             OD
•2980 POKE648,132:POKE53272,(PEEK(53272)A
ND240)OR8:RETURN                                                    ML
•2990 POKE56578,PEEK(56578)ORT3:POKE56576
,(PEEK(56576)AND252)ORT3                                            CO
•3000 POKE648,4:POKE53272,(PEEK(53272)AND
```

```
240)OR4:PRINT"[CLEAR]":RETURN                                    HF
•3010 DATA27,,102,102,63,60,60,60,,29,,10
2,102,252,60,60,60,0,30,240,128,237                              BO
•3020 DATA246,246,109,255,,31,,,182,219,2
19,182,254,,38,,5,2,80,32,10,4,0                                 DD
•3030 DATA35,216,115,22,124,24,24,124,,37
,34,119,239,239,126,253,239,255,-1                               BC
•3040 DATA0,255,255,240,255,255,240,240,,
240,240,,240,240,,240,240,,240                                   CH
•3050 DATA240,,240,240,,240,255,255,240,2
55,255,240,,,,,,,,,,,,,,,0                                      AC
•3060 DATA0,,,,,,,,,,,,,,,1,15,255,,3,2
52,,195,252,48,255,15,240,252,3,240                              MP
•3070 DATA252,3,240,255,15,240,195,252,48
,3,252,,15,255,,,,,,,,,,,,,,,0                                  CB
•3080 DATA0,,,,,,,,,,,,,,,2,227,12,112,
113,152,224,,,,248,1,240,,,,,0                                   KC
•3090 DATA248,1,240,,,,113,152,224,227,12
,112                                                             EO
•3100 DATA0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,-1                                                           GP
```

MOUSE IN THE HOUSE
FROM PAGE 55

```
•10 REM     ********************                                 OP
•20 REM     *MOUSE IN THE HOUSE*                                 JG
•30 REM     *    BY JC HILTY   *                                 KE
•40 REM     ********************                                 OP
•50 V=53248:Q=54272:CS=0:MS=0:LP=0:TM=100
0                                                                LJ
•60 GOSUB1200                                                    FA
•70 DIMS$(4),S(6):S$(0)="[HOME][DOWN][DOW
N]"                                                              PD
•80 FORX=1TO4:S$(X)=S$(X-1)+"[5"[DOWN]"]"
:NEXT                                                            PA
•90 FORX=0TO6:S(X)=X*5+2:NEXT:A$="[RVSON]
[WHITE][c *][sEP][DOWN][LEFT][LEFT][RVSO
FF][c 7][c U][c U]"                                              EN
•100 PRINT"[CLEAR]":POKE53280,0:POKE53281
,15                                                              CE
•110 FORX=56257TO56294:POKEX,2:NEXT:FORX=
1985TO2022:POKEX,160:NEXT                                        BO
•120 PRINTTAB(1)"[RVSON][RED][38" "]"                            II
•130 FORX=1TO22:PRINT"[RVSON]"TAB(1)" "TA
B(38)" ":NEXT                                                    LP
•140 PRINT"[HOME]"TAB(5)TM;TAB(15)"[BLUE]
MOUSE "MS;TAB(27)"[RED]CAT "CS                                   MD
•150 GOSUB800                                                    CI
•160 POKEV+39,8:POKEV+40,11:POKEV+41,6:PO
KE2040,197:POKE2041,192:POKE2042,196                             ML
•170 FORX=50435TO50438:POKEX,0:NEXT:POKE5
0688,6                                                           IF
•175 Y=INT(RND(9)*140)+80:POKEV+1,Y                              DB
•180 POKEV+0,255:POKEV+2,58:POKEV+3,76:PO
KEV+16,4:POKEV+4,46                                              FE
•190 POKEV+5,76:POKEV+21,7:POKE50432,6                           GC
•195 W1=PEEK(V+30):W2=PEEK(V+31)                                 OE
•199 REM    MAIN LOOP                                            PK
•200 JY=PEEK(56321)AND15                                         HM
•202 IFJY=7THENPOKE50435,1:POKE50436,0:PO
KE50688,2:POKE2041,192:GOTO210                                   GL
•204 IFJY=11THENPOKE50435,255:POKE50436,0
:POKE50688,2:POKE2041,194:GOTO210                                NM
•206 IFJY=13THENPOKE50435,0:POKE50436,1:P
OKE50688,2:GOTO210                                               PB
•208 IFJY=14THENPOKE50435,0:POKE50436,255
:POKE50688,2:GOTO210                                             PM
•210 JG=PEEK(56320)AND15                                         IL
•212 IFJG=7THENPOKE50437,1:POKE50438,0:PO
KE50688,4:POKE2042,195:GOTO220                                   AO
•214 IFJG=11THENPOKE50437,255:POKE50438,0
:POKE50688,4:POKE2042,196:GOTO220                                IE
•216 IFJG=13THENPOKE50437,0:POKE50438,1:P
OKE50688,4:GOTO220                                               CG
•218 IFJG=14THENPOKE50437,0:POKE50438,255
:POKE50688,4:GOTO220                                             DB
•220 TM=TM-1:PRINT"[HOME][RED]"TAB(6)"[4"
"]":PRINT"[HOME]"TAB(5)TM:IFTM=0THEN500                          JH
•225 LP=LP+1:IFLP=50THENPOKEV+39,5                               FI
•226 IFLP>70 THENGOSUB300                                        AL
•227 W1=PEEK(V+30):IFW1AND2THEN350                               GI
•228 W2=PEEK(V+31):IFW2>1THEN400                                 PM
•230 GOTO200                                                     BO
•300 R=INT(6*RND(5)+1):ONRGOSUB302,303,30
4,305,306,307                                                    FO
•301 Y=INT(RND(6)*140)+80:POKEV+1,Y:POKEV
+39,8:LP=0:RETURN                                                EO
•302 POKEV+0,55:RETURN                                           DN
•303 POKEV+0,95:RETURN                                           CB
•304 POKEV+0,135:RETURN                                          GI
•305 POKEV+0,175:RETURN                                          FE
•306 POKEV+0,215:RETURN                                          HB
•307 POKEV+0,255:RETURN                                          HN
•349 REM    MOUSE COLLISION                                      NP
•350 POKE50432,0:IFW1>3THEN370                                   GF
•352 POKEV+21,6:FORL=0TO24:POKEQ+L,0:NEXT
:POKEQ+24,143:POKEQ+5,16:POKEQ+19,16                             EH
•354 POKEQ+6,252:POKEQ+20,249                                    NI
•355 FORT=0TO10:POKE2041,193:POKEQ+4,21:P
OKEQ+18,17:POKEQ+1,68:POKEQ+15,42                                HL
•356 FORI=0TO100:NEXT:POKE2041,192:POKEQ+
4,20:POKEQ+18,16:FORI=0TO100:NEXT:NEXTT                          MH
•358 MS=MS+10:PRINT"[HOME][BLUE]"TAB(21)M
S:POKEV+21,0:LP=0:GOTO150                                        FG
•370 POKEV+21,5:FORL=0TO24:POKEQ+L,0:NEXT
:POKEQ+24,15:POKEQ+5,80:POKEQ+6,243                              JC
•372 POKEQ+3,4                                                   CP
•374 FORT=0TO8:POKE2042,195:POKEQ+4,65:FO
RI=20TO80STEP5:POKEQ+1,I:NEXT                                    AL
•376 POKE2042,196:POKEQ+4,64:FORI=0TO50:N
EXT:NEXTT                                                        JO
•380 CS=CS+10:PRINT"[HOME][RED]"TAB(31)CS
:POKEV+21,0:LP=0:GOTO150                                         MO
```

```
·399 REM    SCREEN COLLISIONS                    GK
·400 POKE50432,0                                 AG
·402 IFW2AND2THENK=40:GOSUB420:CS=CS+10:P
 RINT"[HOME][RED]"TAB(31)CS:POKEV+21,0:GO
 TO150                                           FE
·404 K=41:GOSUB420:MS=MS+10:PRINT"[HOME][
 BLUE]"TAB(21)MS:POKEV+21,0:GOTO150              PC
·420 FORL=0TO24:POKEQ+L,0:NEXT:POKEQ+24,1
 5:POKEQ+5,80:POKEQ+6,243:POKEQ+3,7              NL
·422 FORT=50TO17STEP-1:POKEQ+4,65:POKEV+K
 ,2:POKEQ+1,T:FORDR=1TO20:NEXTDR                 HD
·424 POKEV+K,1:FORDR=1TO10:NEXTDR:NEXTT:P
 OKEQ+4,64:LP=0:RETURN                           EG
·499 REM    GAME OVER                            OH
·500 POKE50432,0:PRINT"[CLEAR]":POKEV+21,
 0:POKE53281,0:POKE53280,0                       KF
·505 POKEV+23,6:POKEV+29,6:POKEV+2,80:POK
 EV+3,80:POKEV+16,0:POKEV+4,255                  KD
·506 POKEV+5,80:POKE2041,192:POKE2042,196
 :POKEV+21,6                                     OE
·510 PRINT"[6"[DOWN]"]"TAB(12)"[CYAN]G A
 M E  O V E R":PRINT:PRINT                       GG
·515 PRINTTAB(12)"[BLUE]MOUSE SCORE "MS:P
 RINT:PRINTTAB(12)"[RED]CAT SCORE "CS:PRI
 NT:PRINT                                        EN
·520 GOSUB900:PRINTTAB(12)"[YELLOW]PLAY A
 GAIN  Y OR N"                                   NK
·525 GET JK$:IFJK$<>""THEN525                    ND
·530 GET P$:IFP$=""THEN530                       HF
·535 IFP$="Y"THEN545                             IO
·540 END                                         IC
·545 TM=1000:MS=0:CS=0:LP=0:POKEV+23,0:PO
 KEV+29,0:POKEV+21,0:GOTO100                     LL
·799 REM    SET TRAPS                            AL
·800 PRINT"[HOME][DOWN]":FORX=1TO22:PRINT
 TAB(2)"[36" "]":NEXT                            BK
·810 L=INT(RND(9)*10)+15                         ED
·820 FORX=0TOL:H=INT(RND(5)*7):CO=INT(RND
 (9)*5)                                          KO
·830 PRINTS$(CO)TAB(S(H))A$:NEXT                 MG
·840 RETURN                                      IM
·899 REM    MUSIC                                BC
·900 RESTORE                                     IO
·905 FORX=0TO24:POKEQ+X,0:NEXT:POKEQ+24,1
 5                                               HP
·910 POKEQ+6,240:POKEQ+13,240:POKEQ+20,24
 0                                               DF
·920 READN1,N2,N3,N4,N5,N6,DR:IFN1=0THENR
 ETURN                                           OH
·925 POKEQ+1,N1:POKEQ+8,N3:POKEQ+15,N5:PO
 KEQ,N2:POKEQ+7,N4:POKEQ+14,N6                   BJ
·930 POKEQ+4,17:POKEQ+11,17:POKEQ+18,17:F
 ORZ=1TODR/4:NEXTZ                               CG
·935 POKEQ+4,16:POKEQ+11,16:POKEQ+18,16:F
 ORT=1TO20:NEXTT:GOTO920                         BO
·940 DATA 36,208,0,0,0,0,500,36,208,0,0,0
 ,0,500,30,245,0,0,0,0,500                       BA
·945 DATA 36,208,0,0,0,0,500,41,83,30,245
 ,24,146,500,36,208,0,0,0,0,500                  BJ
·950 DATA 30,245,24,146,18,104,1000              AJ
·955 DATA 30,245,0,0,0,0,500,27,148,0,0,0
 ,0,1000,46,99,32,204,27,148,500                 OE
·960 DATA 30,245,0,0,0,0,500,27,148,0,0,0
 ,0,1000                                         MG
·965 DATA 36,208,0,0,0,0,500,36,208,0,0,0
 ,0,500,36,208,0,0,0,0,500                       BP
·970 DATA 30,245,0,0,0,0,500,36,208,0,0,0
 ,0,500                                          BC
·975 DATA 41,83,30,245,24,146,500,36,208,
 0,0,0,0,500,30,245,24,146,18,104,1000           HN
·980 DATA 27,148,0,0,0,0,500,46,99,32,204
 ,27,148,500                                     GJ
·985 DATA 30,245,0,0,0,0,500,27,148,0,0,0
 ,0,500                                          LD
·990 DATA 24,146,0,0,0,0,1000,30,245,24,1
 46,18,204,1000,0,0,0,0,0,0,0                    KP
·1190 REM    TITLE SCREEN                        NG
·1200 POKE53281,0:POKE53280,0                    ED
·1205 PRINT"[CLEAR]"TAB(3)"[RVSON][c 3][3
 4" "]"                                          OO
·1210 PRINTTAB(3)"[RVSON] M O U S E  I N
  T H E  H O U S E "                             CL
·1215 PRINTTAB(3)"[RVSON][34" "]":PRINT          DD
·1220 PRINTTAB(16)"[RVSON][c 8]  [RVSOFF]
 [3" "][RVSON]  ":PRINTTAB(15)"[RVSON][4"
  "][RVSOFF] [RVSON][4" "]"                      KC
·1225 PRINTTAB(15)"[RVSON][4" "][RVSOFF]
 [RVSON][4" "]":   PRINTTAB(16)"[RVSON]
 [RVSOFF][3" "][RVSON]  ":PRINTTAB(17)"[R
 VSON][4" "]"                                    GL
·1230 PRINTTAB(16)"[RVSON][3" "][RVSOFF][
 WHITE][sEP][RVSON][c 8] [RVSOFF][WHITE][
 sEP]":PRINTTAB(16)"[RVSON][c 8][8" "][RE
 D] "                                            GC
·1235 PRINTTAB(10)"[RVSON][c 8] [RVSOFF][
 5" "][RVSON][8" "]":   PRINTTAB(10)"[RVS
 ON][3" "][RVSOFF]  [RVSON][4" "]"               DA
·1240 PRINTTAB(12)"[RVSON] [RVSOFF]  [RVS
 ON]  [c 4] [c 8] ":PRINTTAB(10)"[RVSON][
 3" "][RVSOFF]  [RVSON]  [c 4] [c 8] "           FC
·1245 PRINTTAB(10)"[RVSON] [RVSOFF][4" "]
 [RVSON] [c 4]  [c 8]  ":PRINTTAB(10)"[RV
 SON] [RVSOFF][3" "][RVSON][6" "]"               OJ
·1250 PRINTTAB(10)"[RVSON] [RVSOFF][3" "]
 [RVSON][6" "]":PRINTTAB(10)"[RVSON][7" "
 ][RVSOFF]  [RVSON]  "                           LB
·1255 PRINTTAB(15)"[RVSON][3" "][RVSOFF]
 [RVSON][3" "]":PRINT                            HM
·1260 PRINTTAB(1)"[CYAN]THERE ARE A LOT O
 F TRAPS AROUND HERE!!"                          AI
·1265 PRINTTAB(4)"BUT I SURE DO LOVE THAT
  CHEESE!!"                                      BJ
·1270 GOSUB900                                   DB
·1280 PRINTTAB(6)"[RVSON][YELLOW] PLEASE
 WAIT..READING DATA "                            OM
·1285 FORX=50880TO51116:READA:POKEX,A:NEX
```

```
T                                                              ND
·1286 FORX=50433TO50438:POKEX,0:NEXT:POKE
50432,0:SYS51104                                               KJ
·1287 FORX=12288TO12670:READA:POKEX,A:NEX
T                                                              ML
·1288 RETURN                                                   IM
·1290 DATA 169,255,45,0,198,240,16,169,0,
141,0,198,162,21,189,0,197,157,0,198                           OC
·1295 DATA 202,208,247,162,1,169,1,141,80
,197,173,80,197,45,0,197,240,3,76                              GF
·1300 DATA 243,198,232,232,14,80,197,208,
238,76,49,234,169,0,29,0,197,208,3,76                          OB
·1305 DATA 97,199,169,128,61,0,197,240,48
,254,0,198,208,40,222,255,207,76,144                           FO
·1310 DATA 199,80,197,45,16,208,208,12,17
3,16,208,13,80,197,141,16,208,76,43                            EA
·1315 DATA 199,173,16,208,77,80,197,141,1
6,208,189,0,197,157,0,198,76,97,199,222                        FC
·1320 DATA 0,198,208,40,254,255,207,208,2
9,173,80,197,45,16,208,208,12,173,16                           GC
·1325 DATA 208,13,80,197,141,16,208,76,91
,199,173,16,208,77,80,197,141,16,208                           LC
·1330 DATA 189,0,197,157,0,198,169,0,232,
29,0,197,208,3,76,140,199,169,128,61,0                         EA
·1335 DATA 197,240,11,254,0,198,208,20,22
2,255,207,76,134,199,222,0,198,208                             CM
·1340 DATA 9,254,255,207,189,0,197,157,0,
198,202,76,233,198,169,255,221,255                             IE
·1345 DATA 207,240,3,76,43,199,173,80,197
,76,17,199,120,169,192,141,20,3,169                            CK
·1350 DATA 198,141,21,3,88,96                                  AF
·1355 REM    SPRITE DATA                                       IB
·1360 DATA 0,0,0,0,0,0,0,0,0,0,198,0,1,23
9,0,1,239,0,0,198,0,0,120,0,0,236,0                            PH
·1365 DATA 32,255,128,56,255,0,9,224,0,57
,160,0,33,160,0,33,48,0,35,240,0                               MM
·1370 DATA 35,240,0,63,152,0,1,220,0,0,0,
0,0,0,0,0                                                      CG
·1375 DATA 0,0,0,0,0,0,0,0,0,0,198,0,1,23
9,0,1,239,0,0,198,0,0,120,0,0,236,0                            PH
·1380 DATA 32,255,128,224,254,0,129,236,0
,225,166,0,33,163,0,33,48,0,35,240,0                           DO
·1385 DATA 35,240,0,63,224,0,0,112,0,0,0,
0,0,0,0,0                                                      LD
·1390 DATA 0,0,0,0,0,0,0,0,0,0,99,0,0,247
,128,0,247,128,0,99,0,0,30,0,0,55,0                            GM
·1395 DATA 1,255,0,0,255,28,0,7,144,0,5,1
56,0,5,132,0,12,132,0,15,196,0,15,196                          PG
·1400 DATA 0,25,252,0,59,128,0,0,0,0,0,0,
0                                                              ED
·1405 DATA 0,0,0,0,0,0,7,248,0,4,8,0,6,8,
0,2,0,0,3,0,64,1,0,120,1,240,108                               PE
·1410 DATA 1,248,252,1,255,252,3,255,192,
3,255,192,3,255,192,7,255,192                                  DC
·1415 DATA 7,192,224,7,240,124,7,240,60,0
,0,0,0,0,0,0,0,0,0,0                                           BP
·1420 DATA 0,0,0,0,0,0,0,0,31,224,0,16,32,0
,16,96,0,0,64,2,0,192,30,0,128                                 JD
·1425 DATA 54,15,128,63,31,128,63,255,128
,3,255,192,3,255,192,3,255,192                                 CG
·1430 DATA 3,255,224,7,3,224,62,15,224,60
,15,224,0,0,0,0,0,0,0,0,0,0,0                                  KM
·1450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,7,255,192,6,255,224,7,254,112                            LO
·1455 DATA 7,254,112,7,59,240,7,63,240,7,
255,176,7,255,240,5,231,144                                    EI
·1460 DATA 7,231,144,3,253,240,3,255,240,
0,0,0,0,0,0,0,0,0,0                                            JK
$360 DATA 0,0,0,0,0,0,0,0,0,0,0,198,0,1,23
9,0,1,239,0,0,198,0,0,120,0,0,236,0                            PH
·1365 DATA 32,255,128,56,255,0,9,224,0,57
,160,0,33,160,0,33,48,0,35,240,0                               MM
·1370 DATA 35,240,0,63,152,0,1,220,0,0,0,
0,0,0,0,0                                                      CG
·1375 DATA 0,0,0,0,0,0,0,0,0,0,0,198,0,1,23
9,0,1,239,0,0,198,0,0,120,0,0,236,0                            PH
·1380 DATA 32,255,128,224,254,0,129,236,0
,225,166,0,33,163,0,33,48,0,35,240,0                           DO
·1385 DATA 35,240,0,63,224,0,0,112,0,0,0,
0,0,0,0,0                                                      LD
·1390 DATA 0,0,0,0,0,0,0,0,0,0,0,99,0,0,247
,128,0,247,128,0,99,0,0,30,0,0,55,0                            GM
·1395 DATA 1,255,0,0,255,28,0,7,144,0,5,1
56,0,5,132,0,12,132,0,15,196,0,15,196                          PG
·1400 DATA 0,25,252,0,59,128,0,0,0,0,0,0,
0                                                              ED
·1405 DATA 0,0,0,0,0,0,7,248,0,4,8,0,6,8,
0,2,0,0,3,0,64,1,0,120,1,240,108                               PE
·1410 DATA 1,248,252,1,255,252,3,255,192,
3,255,192,3,255,192,7,255,192                                  DC
·1415 DATA 7,192,224,7,240,124,7,240,60,0
,0,0,0,0,0,0,0,0,0,0                                           BP
·1420 DATA 0,0,0,0,0,0,0,0,31,224,0,16,32,0
,16,96,0,0,64,2,0,192,30,0,128                                 JD
·1425 DATA 54,15,128,63,31,128,63,255,128
,3,255,192,3,255,192,3,255,192                                 CG
·1430 DATA 3,255,224,7,3,224,62,15,224,60
,15,224,0,0,0,0,0,0,0,0,0,0,0                                  KM
·1450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,7,255,192,6,255,224,7,254,112                            LO
·1455 DATA 7,254,112,7,59,240,7,63,240,7,
255,176,7,255,240,5,231,144                                    EI
·1460 DATA 7,231,144,3,253,240,3,255,240,
0,0,0,0,0,0,0,0,0,0                                            JK
```

PROGRAMS NEEDED

Ahoy! is constantly in search of the finest utility, productivity, and game programs available. Send your best original work on disk, accompanied by a program printout, documentation, and a stamped, self-addressed envelope to: *Ahoy!* Program Submissions, Ion International Inc., 34 West 34th Street—Suite 407, New York, NY 10001.

Continued from page 14

Y-NOT? ($15.00), a 6' "Y" cable for the serial port, contains one male 6-pin plug and two female 6-pin jacks and can be used to operate two printers or to separate the disk drive and the printer to opposite sides of the computer.

The *80 Mono Cable* ($9.00) produces an 80-column mono display from the C-128 on any composite color or monochrome monitor.

All prices include shipping to US or Canada; add $3.00 for COD. MD residents add 5%.

Master Software, 301-922-2962 (see address list, page 14).

C-64 ANIMATION

Consisting of three integrated modules that provide access to the graphics, music, and animation capabilities of the 64 and 128, *Cyber Video* ($39.95) makes it possible to create an animation with hi-res color graphics, sprites, text, music, and sound effects.

The Graphics module provides color mixing, a sprite editor, and standard drawing features like zoom, copying, and outline filling. Pictures are compressed before they are saved so more can be stored on a disk.

The Music module has staffs for three voices, with its editor automatically aligning notes and inserting measure bars. Changes in tempo, volume, octave, time, key, and voice are noted on the score. The user also controls ADSR patterns and filter parameters.

The Animation module permits the loading of picture and song files created with the other two modules. Sprites can then be put into motion over desired paths at desired rates. Included is a demo disk ($5.00 separately—deductible if you buy the full program), a 96-page manual, and an unconditional money back guarantee.

Touchstone, 206-667-9290 (see address list, page 14).

MULTIFINDER

Featuring an editor specially designed to handle text data, the *Multifinder* database manager ($39.95) performs searches in disk drive RAM, eliminating slow data transfer through the serial bus. Field lengths are not defined, and can be of any length up to 248 characters. An interactive demo is available for $2.00.

Knowledgeware, 904-371-6523 (see address list, page 14).

AMIGA BIZ GRAPHICS

Impact ($199.95) allows text, graphics, and icons to be combined in desktop video production of slides for visual annual reports, data analysis, and the like. Its four modules include a Graph Builder (create bar, line, area, scattergram, and pie charts, and display them stacked, overlapped, horizontally or vertically, or in 3-D), Table Builder (text editor supporting multiple fonts and point sizes, and bold, italic, underlining, and shadow effects), Icon Builder (create symbols to connote values on charts, and company logos or initials), Slide Builder (final editing board for combining graphs, icons, tables, and other drawing tools for use in a slide show), and Show Builder (combine slides in any order and choose style of "wipe" from up, down, left, right, fade, spiral, random, and trickle). Slides can be switched automatically or by the mouse.

Aegis Development, Inc., 213-306-0735 (see address list, page 14).

OUTLINE PROCESSOR

The *Thoughtform* outline processor ($35.00) supports headlines nested 200 levels deep. Each headline can have a block of text attached to it, and can be collapsed or expanded to hide or reveal detail. All or part of the outline can be printed out, to any depth specified.

Thoughtform (see address list, page 14).

21st CENTURY WARRIOR:

Apache GUNSHIP™

THE ATTACK HELICOPTER SIMULATION

The Apache... Fierce and elusive, like its warrior namesake... Capable of defeating enemy tanks, infantry, and hostile aircraft on the modern electronic battlefield.

Gunship's revolutionary 3-D graphics enable you, the pilot, to fly into the world's hottest trouble spots... You'll use an unbelievable array of high tech information and weapon systems, including: lasers, video cameras, night viewers, radar warnings, jammers, computers, missiles, rockets, flares, and a 30mm cannon! Successful missions will be rewarded with medals and rank promotions.

Experience the danger and excitement of attack helicopter action... your latest adventure in the ever-growing line of MicroProse Simulation Software.

Challenge the enemy, the sky, and yourself with this extraordinary simulation!

GUNSHIP. *For Commodore 64/128 and Atari XL/XE computers at a suggested retail of $34.95; also for Apple II Family, IBM PC/PC Jr., Tandy 1000, Atari ST and Amiga at $39.95. Call or write for specific machine availability, and for MC/VISA orders if product not found locally.*

MICRO PROSE™
SIMULATION • SOFTWARE

120 Lakefront Drive • Hunt Valley, MD 21030 • (301) 771-1151

Reader Service No. 228

THE STANDARDS
by which the others are judged...

NEW

THE EXECUTIVE DISK #1

Two new 18 hole courses for use with LEADER BOARD. Trees, sand traps, rough, water hazards.

$19.95 C64/128 ONLY

MACH 5

Increase the power of your C64 and 1541 drive. Loads programs up to 500% faster. Adds easy-to-use DOS commands.

$34.95 C64

TOURNAMENT DISK #1

Our new 18 hole courses for use with LEADER BOARD.

$19.95 C64/128, AMIGA ATARI ST, ATARI(48K)

MACH 128

All the power of MACH 5 for the C128 and C64. Features 64 and 128 select switch, warp drive and more!

$49.95 C64 and C128

AMIGA SCREENS

LEADER BOARD
Pro Golf Simulator

$39.95 C64/128, AMIGA, ATARI ST, ATARI(48K)

"This game is simply amazing—there's no other word for it!"
COMPUTE'S GAZETTE

"...the next best thing to a sunny day on the fairways..."
AHOY MAGAZINE

"...the sports simulation of the year, if not the decade!"
ZZAP! 64 (U.K.)

NEW

10th FRAME
Pro Bowling Simulator

The look and 'feel' of real bowling. Up to eight players including league play.

$39.95 C64/C128 (Coming soon for ATARI ST and AMIGA)

NEW

ROBOTIC WORKSHOP

The ultimate high-tec construction set. Discover how easy it is to build hundreds of Robotic projects and control them with your home computer.

$149.95 C64/128 (Coming for Apple,Atari,IBM)

NEW

TRIPLE PACK

BEACHHEAD, BEACHHEAD II RAID OVER MOSCOW

Three best selling, award winning classics! All for one incredible low price.

$19.95 C64/128, APPLE2+/E/C, ATARI 48K

COMMODORE SCREENS

Attention Dealers!
If you wish to be included on our dealer list for new product information and special promotional material, please call 1-(800) 824-2549

ACCESS™
Software Incorporated
Reader Service No. 236

If you can't find our products at your local dealer, you can order direct.
For mail orders, enclose check or money order, plus $1.00 shipping and handling, and specify machine version desired. Order by phone on VISA, Master Charge or C.O.D. by calling (801) 298-9077.
ACCESS SOFTWARE, INC. 2561 South 1560 West
Woods Cross, UT 84087

DON'T MISS THE BOAT!

SUBSCRIBE TO AHOY!

☐ Send me twelve issues of AHOY! for $21.95 ($29.25 Canada & elsewhere). I understand that I will be billed later.

☐ Send me twenty-four issues of AHOY! for $41.75 ($54.25 Canada and elsewhere). I understand that I will be billed later.

NAME_____

ADDRESS_____

CITY_____

STATE_____ ZIP_____

DON'T MISS THE BOAT!

SUBSCRIBE TO THE AHOY! DISK MAGAZINE

☐ Send me 12 issues of the AHOY! Disk Magazine for $88.00 ($115.00 Canada and elsewhere). I understand that I will be billed later.

☐ Send me 24 issues of the AHOY! Disk Magazine for $165.00 ($214.00 Canada and elsewhere). I understand that I will be billed later.

NAME_____

ADDRESS_____

CITY_____

STATE_____ ZIP_____

December 1986 *Ahoy!* Void after December 11, 1986

READER SERVICE CARD

To request additional information on any product advertised or otherwise pictured in this issue of *Ahoy!* that is accompanied by a reader service number, circle the corresponding number below and mail this card. We will promptly forward your request to the designated companies.

101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	210
211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300

A. Please check whether you are...

1. ☐ male 2. ☐ female

B. What is your age?

1. ☐ under 18 2. ☐ 18-24 3. ☐ 25-34
4. ☐ 35-44 5. ☐ 45-54 6. ☐ 55-64 7. ☐ 65+

C. Education level completed

1. ☐ elementary 2. ☐ high school
3. ☐ junior college 4. ☐ college graduate
5. ☐ master's degree 6. ☐ PhD

D. From which of the following sources did you obtain your copy of *Ahoy!*?

1. ☐ newsstand 2. ☐ subscription (mail)
3. ☐ from a friend or family member
4. ☐ other_____

If not currently a subscriber, do you plan to become one?

1. ☐ yes 2. ☐ no

Name_____

Street_____

City_____ State_____ Zip_____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 65 MT. MORRIS, ILL.

POSTAGE WILL BE PAID BY ADDRESSEE

Ahoy!

P.O. BOX 341
MT. MORRIS, ILL. 61054

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 65 MT. MORRIS, ILL.

POSTAGE WILL BE PAID BY ADDRESSEE

Ahoy!

P.O. BOX 341
MT. MORRIS, ILL. 61054

PLACE
STAMP
HERE

AHOY! MAGAZINE
P.O. Box 2501
Clinton, Iowa 52735