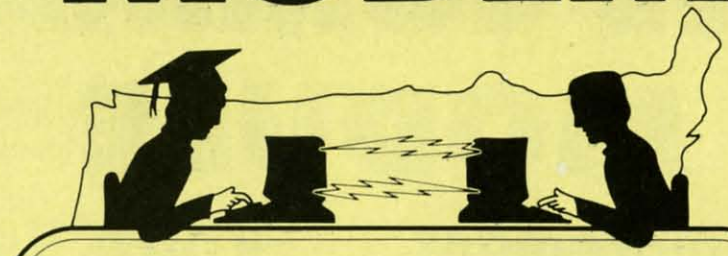
Commodore 64

MODEM

Commodore 64

Sale



SAVE

List $99.00

Sale $29.95

Coupon $24.95

Telecommunications

Lowest Price In the U.S.A.

List $99.00

Sale $29.95

Coupon $24.95

FOR CHILDREN·ADULTS·BUSINESS

Complete Auto Dial Telecommunications Package

"The only telecommunications package you will ever need."

(Exclusive Easy To Use Features)

FREE QUANTUM LINK DATABASE MEMBERSHIP

• Only Good Color Graphic Database Service in the U.S.A. (C-64)

Quantum Link Software Plus First Month FREE (See the Protecto Catalog On-Line) $9.95 value

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

Reach Out and Access Someone

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

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

- Auto Log-on
- 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

List $99.00 **Sale $29.95** *Coupon $24.95*

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

Viewtron Membership sold separately — $9.95.

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. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 300

Commodore 64

VOICE COMMAND

Coupon $34.95 *Coupon $34.95*

$39.95 MODULE $39.95

Keyboard Replacement Voice Recognition

The Voice Command Module is a speech recognition device that lets you give commands to your Commodore-64 with your voice instead of a keyboard. This unit converts the sound waves generated when you talk into digital data that is stored in the computer memory. When you speak to your computer, the words you speak are matched against the data stored in memory and the result is converted to an instruction for the computer to perform. This is perfect for programmers and first time users alike. Six programs are included to help you get acquainted with the world of speech recognition.

SOS — Speech Operating System — This is the general utility program which helps you to build a speech file made up of a set of words.

Card File Program — This is a data base much like an index card file which you can control with your voice. You can store recipes, addresses, phone numbers or any kind of information you need to have filed. Up to 100 files may be kept on a single disk.

Aeronaut Game — This game challenges you to land a hot air balloon on 5 different landing pads without crashing into anything or running out of fuel. The balloons altitude is controlled by your voice which adds or removes hot air from the balloon.

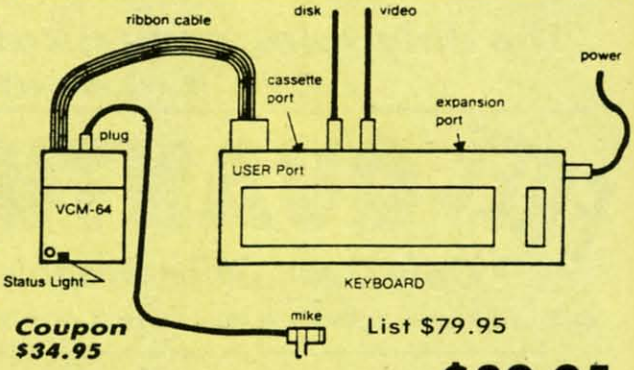
Word Mix Puzzle — Here you must match words much like Concentration. If you guess correctly you win. No hands on the keyboard are needed since the speech recognition unit does the keyboard work from your voice.

Speech Graphics — Demonstrates how the voice command module works. Here you can graphically see what your speech looks like on the screen.

Demo Program — This is a simpler version of the Aeronaut game that shows you how a simple program is made when listed.

PLUS: You get easy to use instructions for making your own programs in BASIC or machine language using the voice command module.

All Six Programs Included FREE



Coupon $34.95 List $79.95

Special Introductory Price $39.95

VOICE SYNTHESIZER

MAKE YOUR COMPUTER TALK List $89.00

VOICE SYNTHESIZER — You can program any words or sentences • Adjust volume and pitch • Receive Modem messages • Make adventure games that talk • Real sound action games • Make customized talkies • Plugs into cartridge port. Plus **FREE** Text to Speech Software. (See Below) List $89.00. **Sale $39.95.**

$39.95 SALE

FREE $19.95 Software w/ Synthesizer Purchase.

TALKING MODEM PROGRAM — This program allows all words sent to your modem to be spoken. Fantastic for modem games and receiving reports. List $24.95. **Sale $16.95.** (Disk/Tape)

TEXT TO SPEECH SOFTWARE — Allows you to simply type what you want to hear!! Also allows you to add sound & voice to SCOTT ADAMS & "ZORK" ADVENTURE GAMES. List $29.95. **Sale $19.95.** (Disk).

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. · No. C.O.D. to Canada, APO-FPO

Reader Service No. 300

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Commodore Software Sale

ORDER TODAY!

Name _____

Address _____

City _____ State _____ Zip _____

Visa/Mastercard No. _____

Exp. Date _____ Phone (_____) _____

(T) Tape, (C) Cartridge, (D) Disk.

GAMES

Accolade

5950 HARDBALL (D)	$29.95	$18.95
5952 LAW OF THE WEST (D)	29.95	18.95
5954 FIGHT NIGHT (D)	29.95	18.95
5956 PSI 5 TRADING CO. (D)	29.95	18.95
5958 THE DAM BUSTERS (D)	29.95	18.95

Activision

0757 RIVER RAID (D)	$39.95	$18.95
0761 PITFALL II — LOST CAVERNS (D)	39.95	18.95
0900 SPACE SHUTTLE (D)	37.95	18.95
0932 ON FIELD FOOTBALL (D)	39.95	18.95
0936 ON COURT TENNIS (D)	39.95	18.95
0940 GHOSTBUSTERS (D)	39.95	22.95
3580 GREAT AMERICAN RD. RACE (D)	29.95	20.95
3582 MASTER OF THE LAMPS (D)	29.95	20.95
3584 COUNTDOWN/SHUTDOWN (D)	29.95	20.95
3588 MINDSHADOW (D)	29.95	20.95
3590 STAR LEAGUE BASEBALL (D)	29.95	20.95
3592 ALCAZAR (D)	29.95	20.95
5196 LITTLE PEOPLE PROJECT (D)	34.95	24.95
5198 FAST TRACKS (D)	34.95	20.95

Broderbund

2900 MASK OF THE SUN (D)	$39.95	$23.95
2901 OPERATION WHIRLWIND (D)	39.95	22.95
2903 LODE RUNNER (D)	34.95	19.95
2904 THE CASTLES OF DR. CREEP (D)	29.95	18.95
2906 WHISTLERS BROTHER (D)	29.95	18.95
3039 STEALTH (D)	29.95	22.95
3041 RAID ON BUNGELING BAY (D)	29.95	18.50
2905 KARATEKA (D)	29.95	23.95
3038 CHAMPION LODE RUNNER (D)	34.95	26.95
5158 BANK STREET WRITER (D)	49.95	32.95
5330 BANK STREET SPELLER (D)	49.95	32.95
5332 BANK STREET FILER (D)	49.95	32.95
5334 BANK STREET MAILER (D)	49.95	32.95

Datasoft

3025 BRUCE LEE (D)	$34.95	$18.95
3026 PAC-MAN (D)	34.95	18.00
3027 MIGHTY CONAN (D)	34.95	18.00
3028 MR DO! (D)	34.95	18.95
3029 DIG DUG (D)	34.95	18.95
3032 POLE POSITION (D)	34.95	18.95
5216 ALTERNATE REALITY	39.95	25.95
5218 THE GOONIES (D)	29.95	18.95
5220 ZORRO (D)	29.95	18.95

Electronic Arts

3830 DR. J & LARRY BIRD (D)	$29.95	$23.95
3832 FINANCIAL COOKBOOK (D)	39.95	27.95
3834 MAIL ORDER MONSTERS (D)	34.95	22.95
3840 THE SEVEN CITIES OF GOLD (D)	29.95	23.95
3842 SKY FOX (D)	29.95	23.95
5176 CARRIERS AT WAR (D)	42.95	32.95
5178 REARCH FOR THE STARS II (D)	37.95	28.95
5180 HEART OF AFRICA (D)	29.95	23.95
5182 MOVIE MAKER (D)	29.95	23.95
5184 EUROPE ABLAZE (D)	42.95	34.95
5186 M.U.L.E. (D)	19.95	16.95
5188 MURDER ON ZINDERNEUF (D)	19.95	16.95
5190 MUSIC CONSTRUCTION SET (D)	19.95	16.95
5192 PINBALL CONSTRUCTION SET (D)	19.95	16.95
5194 RACING CONSTRUCTION SET (D)	29.95	22.95

Epyx

0337 WORLD'S GREAT FOOTBALL (D)	$39.95	$23.95
0338 WINTER GAMES (D)	39.95	20.95
0339 THE EIDOLON (D)	39.95	20.95
0340 KORONIS RIFT (D)	39.95	20.95
0360 JET COMBAT SIMULATION (D)	39.95	20.95
0364 SUMMER OLYMPIC GAMES (D)	39.95	18.95
0365 WORLD'S GREAT BASEBALL (D)	34.95	20.95
0382 SUMMER OLYMPIC GAMES II (D)	39.95	20.95
0750 PITSTOP II (D)	39.95	22.95
2046 IMPOSSIBLE MISSION (D)	34.95	16.95
2066 ROBOTS OF DAWN (D)	39.95	15.95
2070 BARBIE (D)	39.95	18.95
2074 G.I. JOE (D)	39.95	18.95
2085 BREAKDANCE (D)	39.95	16.95
2305 SCRABBLE (D)	39.95	24.95
3004 CHIPWITS (D)	34.95	15.95
3005 BALLBLAZER (D)	29.95	24.95
3006 RESCUE ON FRACTALUS! (D)	29.95	20.95

Strategic Simulations, Inc.

2995 RDF 1985 (D)	$34.95	$20.95
2996 COMBAT LEADER (D)	39.95	23.95
2997 GEOPOLITIQUE (D)	39.95	24.95
2998 BALTIC 1985 (D)	34.95	20.95
3008 RINGSIDE SEAT (D)	39.95	23.95
3009 THE COOSMIC BALANCE (D)	39.95	23.95
3010 IMPERIUM GALACTUM (D)	39.95	23.95
3011 CARTELS AND CUTTHROATS (D)	39.95	23.95
3012 RAILS WEST (D)	39.95	26.95
3013 TIGERS IN THE SNOW (D)	39.95	23.95
3014 PROFESSIONAL TOUR GOLF (D)	39.95	23.95
3015 50 MISSION CRUSH (D)	39.95	23.95
3016 PRESIDENT ELECT (D)	39.95	23.95
3017 BROADSIDES (D)	39.95	24.95
3018 COMPUTER QUARTERBACK (D)	39.95	24.95
3020 COMPUTER AMBUSH (D)	59.95	37.95
3021 COMPUTER BASEBALL (D)	39.95	23.95
3030 KNIGHTS IN THE DESERT (D)	39.95	23.95
3031 FIELD OF FIRE (D)	39.95	23.95

Suncom

3876 PARTY QUIZ (D)	$49.95	$14.95
3880 GENERAL EDITION (D)	39.95	14.95
3882 GENERAL EDITION III	39.95	14.95
3884 SPORTS EDITION (D)	39.95	14.95
3886 EDUCATION EDITION (D)	39.95	14.95
3888 BIBLE EDITION (D)	39.95	14.95

BUSINESS

Codewriter

0129 FILEWRITER (D)	$39.95	$29.95
0706 REPORTWRITER (D)	39.95	29.95
0707 ADVENTUREWRITER (D)	49.95	39.95
0708 ELF (D)	39.95	29.95
0709 DIALOG (D)	49.95	39.95
0784 MENUWRITER (D)	39.95	29.95
2550 SPEEDWRITER (D)	49.95	39.95
2551 HOME INTEGRATOR (D)	29.95	19.95

Softcync

5930 ACCOUNTANT, INC. (D)	$99.95	$64.95
5932 PERSONAL ACCOUNTANT (D)	34.95	26.95
5934 MODEL DIET (D)	29.95	23.95
5936 TRIO (D)	49.95	45.95
5938 KID PRO QUO (D)	29.95	23.95
5940 DESK MANAGER (D)	39.95	28.95

Timeworks

0176 INVENTORY MANAGE (D)	$69.95	$38.95
0180 ACCOUNTS. RECEIVABLE/ INVOICING (D)	69.00	38.95
0182 ACCOUNTS PAYABLE/ CHECKWRITING (D)	69.00	38.95
0184 PAYROLL MANAGEMENT (D)	69.00	38.95
0188 GENERAL LEDGER (D)	69.00	38.95
0207 ELECTRONIC CHECKBOOK (D& T)	29.95	19.95
0231 MONEY MANAGER (D&T)	29.95	19.95
0235 DATA MANAGER (D&T)	29.95	19.95
0928 EVELYN WOOD SPEED READ (D)	69.96	32.95
3743 SYLVIA PORTER (D)	59.95	38.95

C128 Software From Timeworks

5022 WORD WRITER/ SPELL CHECKER (D)	$69.96	$59.95
5024 DATA MANAGER II (D)	69.96	49.95
5026 SWIFTCALC WITH SIDEWAYS (D)	69.96	49.95

Phone Orders

CALL 8 to 8 C.S.T. - M-F

312-382-5244

EDUCATION

American Educational Computer

2482 ELEM. SCIENCE FACTS (D)	$29.95	$14.95
2492 VOCABULARY WORD BUILD (D)	29.95	14.95
2493 GRAMMAR WORD SKILLS (D)	29.95	14.95
2494 WORLD GEOGRAPHY FACTS (D)	29.95	14.95
2495 SPANISH VOCAB. SKILLS (D)	29.95	14.95
2496 FRENCH VOCAB. SKILLS (D)	29.95	14.95
2497 WORLD HISTORY (D)	29.95	14.95
2498 U.S. HISTORY FACTS (D)	29.95	14.95
2499 BIOLOGY FACTS (D)	29.95	14.95
2519 U.S. GEOGRAPHY FACTS (D)	29.95	14.95
2520 U.S. GOVERNMENT FACTS (D)	29.95	14.95
2521 AEC SPELLING (D)	39.95	24.95
3745 PHONICS (D)	39.95	24.95
3747 LEARN TO READ (D)	39.95	24.95
3749 READING COMPRENSION (D)	39.95	24.95

Designware

0824 GRAMMAR EXAMINER (D)	$39.95	$24.95
0828 SPELLAKAZAM (D)	34.95	19.95
0832 STATES & TRAITS (D)	44.95	27.95
0836 SPELLICOPTER (D)	39.95	24.95
0840 CREATURE CREATOR (D)	34.95	19.95
0844 TRAP-A-ZOID (D)	39.95	25.95
2518 THE BODY TRANSPARENT (D)	44.95	27.95
2517 EUROPEAN NATIONS & LOCATIONS (D)	44.95	27.95
2062 MATH MAZE (D)	39.95	26.95
5100 ALGEBRA I (D)	39.95	28.95
5102 REMEMBER (D)	69.96	49.95
5104 WEBSTER'S NUMBERS (D)	39.95	28.95
5105 SPELLING & READ PRIMER (D)	39.95	24.95
5106 ALGEBRA 2 (D)	39.95	28.95
5107 ALGEBRA 3 (D)	39.95	28.95

Mindscape

5108 KEYBOARD CADET (D)	39.95	25.95
5110 BANK STREET MUSIC WRITER (D)	39.95	25.95
5112 CROSSWORD MAGIC (D)	49.95	29.95
5114 THE PERFECT SCORE (D)	69.96	45.95
5116 COLORME/RAINBOW BRITE (D)	34.95	18.95
5118 THE HALLEY PROJECT (D)	39.95	25.95
5120 INDIANA JONES IN THE LOST KINGDOM (D)	29.95	18.95
5122 BANK STREET STORYBOOK (D)	39.95	25.95
5910 THE DOLPHIN'S RUNE (D)	29.95	18.95
5912 THE LUSCHER PROFILE (D)	39.95	25.95
5914 QUAKE MINUS ONE (D)	29.95	18.95
5916 THE LORDS OF MIDNIGHT (D)	29.95	18.95
5918 SHADOWFIRE (D)	29.95	18.95

Weekly Reader

2511 STICKYBEAR BOP (D)	$34.95	$19.95
2512 STICKYBEAR NUMBERS (D)	34.95	19.95
2513 STICKYBEAR BASKETBOUNCE (D)	34.95	19.95
2514 STICKYBEAR OPPOSITES (D)	34.95	19.95
2515 STICKYBEAR ABC (D)	34.95	19.95
2516 STICKYBEAR SHAPES (D)	34.95	19.95
2600 PIC BUILDER (D)	29.95	19.95
5126 STICKYBEAR SPELLGRABBER (D)	29.95	19.95
5128 STICKYBEAR TOWN BUILDER (D)	29.95	19.95
5130 STICKYBEAR MATH (D)	29.95	19.95

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. No. C.O.D. to Canada, APO-FPO

Reader Service No. 300

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

FUJI Floppy Disk SALE

LOW AS **99¢*** ea.

Premium Quality Floppy Disks

Lifetime Guarantee

EXPIRES 6-30-86

Box of 10 *with hub rings, sleeves and labels*

List $29.95 **Reg. Sale $14.90** PER BOX/10

Famous Brand FUJI Floppy Disks for those who care about keeping their data.

Single Sided — Double Density for Commodore 64, Atari, Apple

*** Special Deal** Buy 2 Boxes for $29.70 and we will give you 1 Box Free

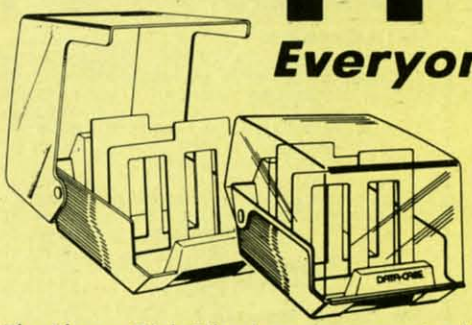
You Get 3 Boxes for $29.70

Net Cost 99¢ Each

*** $12.95** ☆☆ ***Flip-N-File*** ☆☆ *** $12.95**

Data-Case

Floppy Disk Filer



Everyone Needs a Floppy Disk Secretary

Facts:

- Dust and Dirt particles can hurt your disks
- Most disks go bad due to mishandling in storage
- Proper filing of your disk collection will reduce unnecessary handling of your disks

The Floppy Disk Filer is an inexpensive hard plastic Fliptop case that will allow for easy filing, and protect your disks from dust, smoke, and dirt. Plus, the Floppy Disk Filer will keep all your disks out of unwanted hands and in one place where you can easily find them. **(Holds Over 50 Disks)**

List $24.95 **Introductory Sale Price $14.95** **** Coupon $12.95***

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. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 300

GIANT PRINTER SALE!!

List $399.00 **10" Printer**

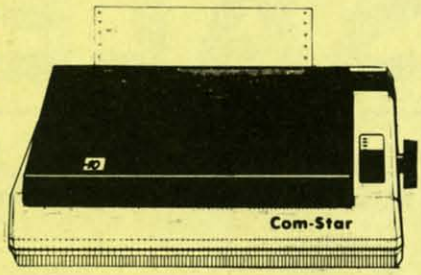
SALE **$148.00** SALE

1 Year Warranty
120-140 CPS
Premium Quality

List $599.00 **15½" Printer**

SALE **$229.00** SALE

10" Comstar 10X — This Bi-directional Tractor/Friction Printer prints standard sheet 8½"x11" paper and continuous forms or labels. High resolution bit image graphics, underlining, horizontal tab setting, true lower descenders, with super scripts and subscripts, prints standard pica, compressed, expanded, block graphics, etc. Fantastic value. (Centronics parallel interface.)
List $399.00. **Sale $148.00**



15½" Comstar 15X — Has all the features of the 10" Comstar 10X plus a wider 15½" carriage and more powerful electronics to handle large ledger business forms! (Better than FX-100). The 15½" Comstar 15X also prints on standard size paper and continuous forms and labels. Fantastic value. (Centronics parallel interface.)
List $599.00. **Sale $229.00**

List $499.00 **10" Printer**

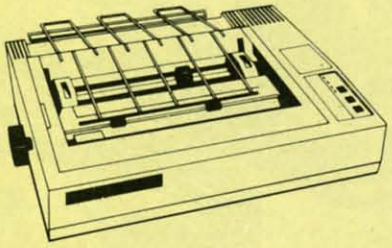
SALE **$179.00** SALE

1 Year Warranty
150-170 CPS
High Speed

List $699.00 **15½" Printer**

SALE **$259.00** SALE

10" Comstar 160+ High Speed — This Bi-directional Tractor/Friction Printer combines the above features of the 10" Comstar 10X with speed (150-170 cps) and durability. Plus you get a 2K buffer, 96 user definable characters, super density bit image graphics, and square print pins for clearer, more legible print (near letter quality). This is the best value for a rugged dependable printer. (Centronics parallel interface.)
List $499.00. **Sale $179.00**



15½" Comstar 160+ High Speed — This Bi-directional Tractor/Friction Printer has all the features of the 10" Comstar 160+ High Speed plus a wider 15½" carriage and the heavy duty electronics required for today's business loads. You can use large ledger business forms as well as standard sheets and continuous forms and labels. This is the best wide carriage printer in the U.S.A. (Centronics parallel interface.)
List $699.00. **Sale $259.00**

List $599.00 **10" Printer**

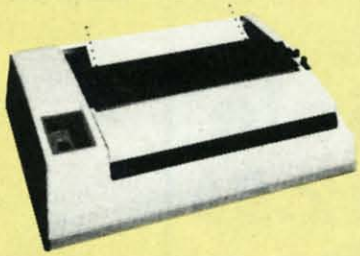
SALE **$229.00** SALE

Lifetime warranty*
165-185 CPS
High Speed & Letter Quality

List $599.00 **10" Printer**

SALE **$229.00** SALE

10" Comstar 2000 — The ultimate printer has arrived! This Bi-directional Tractor/Friction Printer gives you all the features of the Comstar 160 plus higher speed (165-185 cps), 256 downloadable characters, proportional setting, external dark printing mode and a *lifetime printhead warranty. **PLUS** ...



With the flip of a switch you can go into the letter quality mode which makes all your printing look like it came off a typewriter. Turn in term papers, do articles or just print programs. Have the best of letter quality and speed in one package. Fantastic printer (Centronics parallel interface.)
List $599.00. **Sale $229.00**

• ***15 Day Free Trial — 1 Year Immediate Replacement Warranty***

Parallel Interfaces

Commodore-64, VIC 20 — $39.00 | Atari — $59.00 | Apple II, II+, IIe — $59.00

Add $10.00 ($14.50 for 15½" Printers) for shipping, handling and insurance. Illinois residents please add 6% 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 Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D. No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 300

Sale

Economy* to Arcade Quality

JOYSTICK

SAVE

Low As

*$8.95 SALE

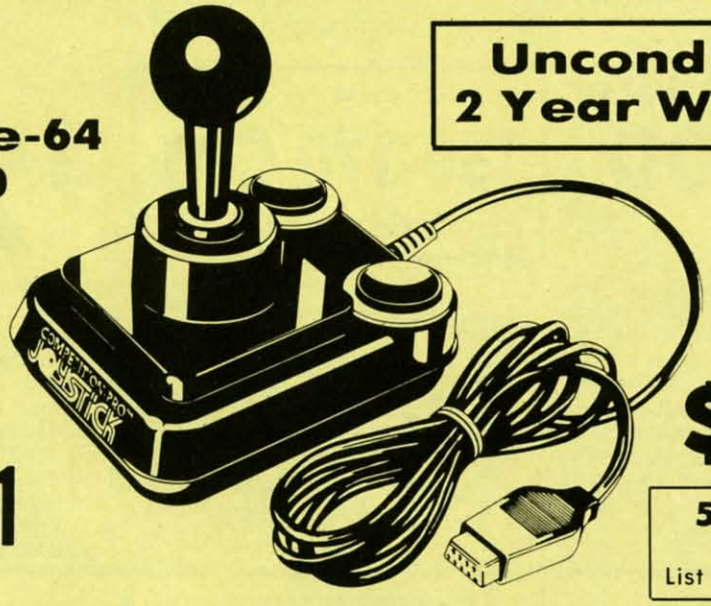
Low As

*$8.95

Ultimate Arcade Quality Model

Unconditional 2 Year Warranty

Commodore-64 & VIC-20



List $24.95

Sale $16.95

Coin Controls 5000

Rated No. 1

5200 Joystick
Allows keypad hook-up
List $29.95 **Sale $24.95**

Professional Cadillac Model

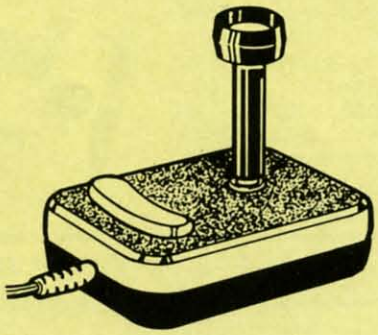
"Three Way Firing" Options

List $19.95

Sale $12.95

Coupon Price $10.00

*Single Button Economy Model



List $12.95

Sale $8.95

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. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

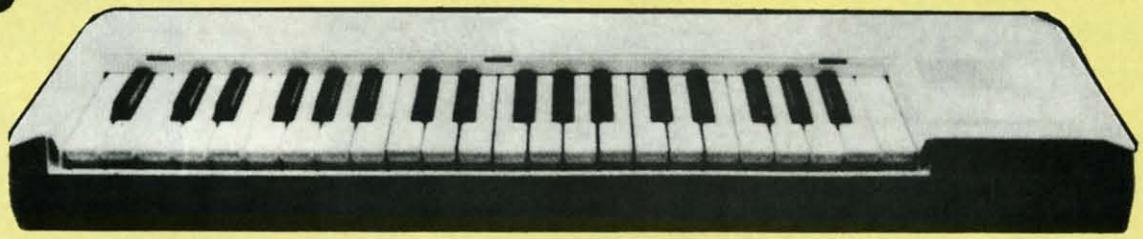
312/382-5244 to order

Reader Service No. 300

Full Size Piano/Organ

$69 KEYBOARD $69

SALE



SALE

THE COMMODORE PIANO MUSICAL KEYBOARD

Keyboard — 40 Keys (A-C) in professional gauge spring loaded to give the feel and response of a real keyboard instrument. Polyphonic.

Registers (with the Conductor) — Organ, Trumpet, Flute, Clarinet, Piano, Harpsicord, Violin, Cello, Bass, Banjo, Mandolin, Calliope, Concertino, Bagpipe, Synthesizer 1, Synthesizer 2, Clavier 1, Clavier 2, which can be played over a 7 octave range. Programmable sounds as well.

Recording (with the Conductor) — Three track sequencer (recorder) with over-dubbing and multitimbral (different instrument sounds at the same time) effects.

Interface — Built in interface for Commodore 64, Commodore 128, plugs right in to joystick port no. 2 and user port.

Finish — Table Model in white high-impact material, with carrying handle, protective key cover, and built in music stand. Size 29 1/8x 9 9/16 x 3-11/16, weighs 9 pounds.

List $159.95 **Sale $69.00**

The Conductor Software

List $29.95 **Sale $19.95**

The Conductor teaches how a composition is put together, note by note, instrument by instrument. You learn to play 35 pre-recorded songs from Bach to Rock. Then you can compose your own songs and record them right onto your floppy disk.

Teaches:
- Scales
- Bass lines
- Familiar Beginner Songs such as "Jingle Bells"
- Easy classical songs such as "Bach Minuet" and Ravel's "Bolero"
- Advanced classics like "A Mid-summer's Night Dream" by Mendelssohn
- Popular hits such as "Thriller"

Create New Instrument Sounds
- Choose from pulse, sawtooth, triangle and noise and sound sources.
- Control the sound envelope with attack, delay, sustain and release times.
- Ring Modulation and Syncronization effects.
- Set Low pass, band pass, and high pass filter frequencies.

Recording Functions:
- Monitor: Lets you use a track to play music live, without recording it.
- Record: Records a track as you play.
- Playback: Lets you hear whatever has been recorded or loaded into the track. You may playback one track while recording another to build layers of instruments.
- Mute: Turns a track off. This is useful when you want to listen to a record one or two tracks at a time.
- Save: Stores a track to the disk.
- Load: Loads a track from disk.
- Protect: Write protects a track.

The Conductor Requires:
- Commodore 64 or Commodore 128 with disk drive.
- The Commodore Piano Musical Keyboard is required to study the reading and playing of musical notes.

The Printed Song

List $29.95 **Sale $19.95**

With the Printed Song program your music can be printed out in music notation, which other musicians can read and play. Any music recorded with the Conductor program can be printed by the Printed Song.

The Printed Song Requires: • Commodore 64 or Commodore 128 with disk drive and printer compatible with the Commodore graphics mode such as the Commodore MPS 803, 1515, and 1525. • The Conductor program.

The Music Teacher Software

List $39.95 **Sale $29.95**

The Music Teacher teaches a beginner how to read music and play it correctly and in rhythm on the musical keyboard.
The Music Teacher will have you reading and playing musical notes in minutes with fun and excitement.

Features: • Trumpet, organ, violin, and synthesizer instrument sounds. • Built in metronome. • Pause/Play control. • Set-up menu for customizing The Music Teacher.

Teaches: • How to read notes on the treble and bass musical staffs. • The names of the notes. • Where the notes are on the keyboard. • How to play whole notes, half notes, quarter notes, eighth notes and sixteenth notes in combinations in both 3/4 and 4/4 time. • How to play in different tempos.

Requires: • Commodore 64 or Commodore 128 with disk drive. • The Commodore Piano Musical Keyboard

The Technician

List $29.95 **Sale $24.95**

Contains programs, and BASIC source listings for reading the Commodore Piano Musical Keyboard, and for reading and creating music files for the Conductor.

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6¼% 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. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 300

C-64 Sale MINDSCAPE C-64 Sale

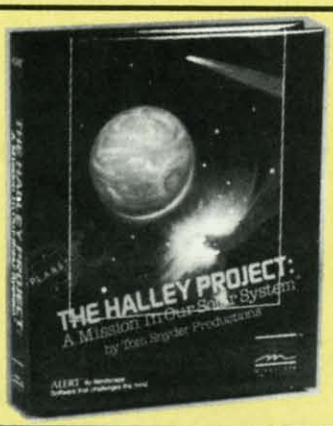
SOFTWARE THAT CHALLENGES THE MIND

Crossword Magic — Create your own crossword puzzles. A unique way to study any subject in any language. Crossword Magic can be used again and again by every member of the family. (Disk) List $49.95. **Sale $29.95.**

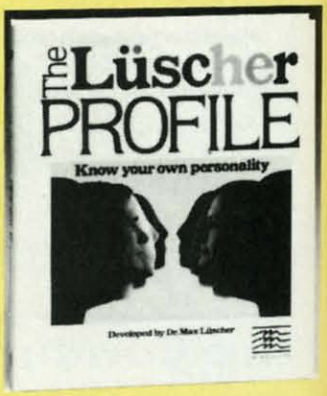
The Perfect Score: Computer Preparation For The SAT — The most complete Computer program for preparing for the SAT test. This package contains **six** double sided disks covering all sections of the test. Timed exam included. (Disk)
List $69.96. **Sale $45.95.** *Coupon $42.95.*

ColorMe with Rainbow Brite Picture Disk — Your young children can enjoy hours of creativity as they draw and color. They can draw freehand or use predrawn pictures from the Rainbow Brite picture disk included with ColorMe (ltd. qty.) (Disk) List $34.95. **Sale $18.95.**

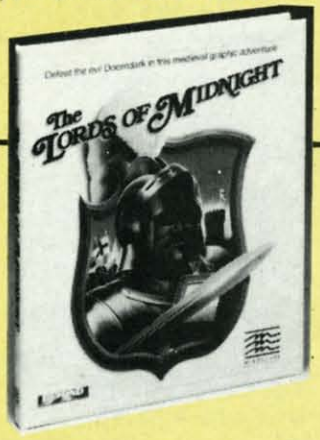
The Halley Project: A Mission In Our Solar System — Pilots are needed for a top-secret space exploration mission. Only the most skilled will be accepted. To qualify you must pass a series of tests. As you travel through the solar system your only guide is a radarscope and an ability to navigate by the stars. (Disk) List $39.95.
Sale $25.95.
Coupon $24.95.



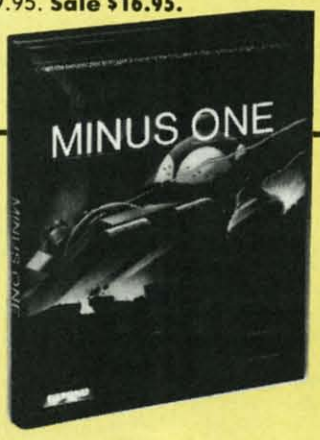
The Luscher Profile — Wouldn't it be great if you could run a personality profile on those who confuse you most? To understand the traits behind the reactions of your spouse, child, parents, in-laws, or best friend? If someone has you in a quandry then this program will help you to answer the probing question, "what makes him tick?" With 35 years of research on the theory of how psychologically revealing a person's color and shape choices can be, Mindscape created the program The Luscher Profile to reflect that discovery. (Disk) List $39.95. **Sale $28.95.**



The Lords Of Midnight — The land of Midnight is controlled by Doomdark and your mission in causing his fall from power is to destroy the source of his strength, The Ice Crown. Battle 32,000 panaramas creatures. Courage and bravery will hopefully see you through. (Disk) List $19.95. **Sale $16.95.**



Quake Minus One — You must stand up to a vicious terrorist group bent on gaining control of the Titan Power Station and causing an extremely destructive earthquake. You have only ten hours to figure out a solution to this probable devastation. First you must destroy four Titan computers, then stop the quake. (Disk) List $19.95. **Sale $16.95.**



Shadowfire — This game allows you 100 real-time minutes to rescue Ambassador Kryxix and demolish the enemy starship. The aliens that confront you are beyond the realm of the most vivid imagination The pace is fast and the action intense. (Disk) List $19.95. **Sale $16.95.**

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. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 300

Commodore 64

Buy 1 Get One FREE

(at Sale Price!)

Expires 4-30-86

Limit one free program per customer

Commodore 64 SALE Weekly Reader Commodore 64 SALE

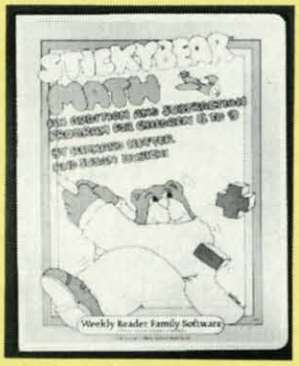
Educational Programs

Buy any Weekly Reader program from Protecto and choose another Weekly Reader program FREE! ($29.95 value)

Stickybear Typing — Stickybear makes learning the keyboard both easy & fun for anyone. A 3 game format provides different levels so that as typing skills improve so does the level of difficulty. ***Stickybear Keypress*** is geared to the needs of the child or the beginner, the stress is on the keyboard. To increase typing speed and accuracy the fast-paced ***Stickybear Thump*** will provide a real challenge. ***Stickybear Stories*** allows further practice with real text for timed practice. Everything from jokes, stories and bright amusing graphics make this one of the most delightful typing programs ever, plus you can keep track of your progress and store on disk. (Disk) List $29.95. **Sale $19.95.**

C64 SALE

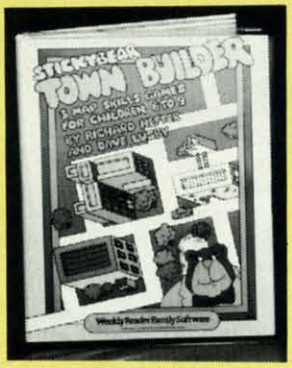
SALE C64

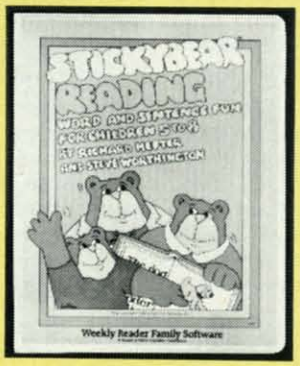


Stickybear Spellgrabber — 3 games in one: Picture Spell contains over 200 word/picture combinations. Word Spell is a scramble game, players recreate words shown on the screen. Bear Dunk involves guessing a word to save Stickybear from a dunking. (Disk) List $29.95. **Sale $19.95.**

Stickybear Math — Kids learn math skills as they help the Stickybear Family out of sticky jams. For every set of addition and subtraction problems youngsters get right, Stickybear gets a step closer to getting out of the jam. (Disk) List $29.95. **Sale $19.95.**

Stickybear Town Builder Kids develop essential map skills as they build a town with roads, parks, airports, bridges, etc. and save on disk. That's just the first game. In Take A Drive, Compass-reading and directions are stressed. With Find The Keys, follow clues to locate mystery keys. (Disk) List $29.95. **Sale $19.95.**





Stickybear Reading
3 fun activities that build reading comprehension skills. Match The Words: Kids practice vocabulary as they match up word and picture sets. Find The Word: Turns the words in a sentence into animated action. Build A Sentence: Children choose a subject, a verb and an object, then watch them turn into a picture. (Disk) List $29.95. **Sale $19.95.**

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. No. C.O.D. to Canada, APO-FPO

Reader Service No. 300

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

EPYX
COMPUTER SOFTWARE

Commodore 64

First 1000 Purchasers of Epyx product from Protecto will receive a Free Scrabble game $29.95 Value

Fast Load Cartridge — Load, save and copy disks five times faster than normal. It plugs into the cartridge port ot the Commodore 64 and goes to work automatically loading disks with ease. And that's only the beginning. It can copy a single file, copy the whole disk, send disk commands, and even list directories without erasing programs stored in memory. (Cartridge). List $39.95. **Sale $23.95.**

EPYX
COMPUTER SOFTWARE

Commodore 64

First 1000 Purchasers of Epyx product from Protecto will receive a Free Scrabble game $29.95 Value

Buy 1 Get 1 FREE to 1st 1000

Limited Quantities — Limit 1 per Customer

ORDER NOW **ORDER NOW**

The Best Arcade Games Around

Rescue On Fractalus! — Your mission is to fly your Valkyrie Fighter through the Jaggi defenses and rescue the downed Ethercorps pilots. Sounds easy, but don't let it fool you. It's tough enough just to navigate the mountains and canyons of Fractalus, but try doing it while destroying enemy gun emplacements or dodging suicide saucers. We supply the Long Range Scanner, Dirac Mirror Shield and Anti-Matter Bubble Torpedoes ... YOU supply the skill and guts! One Player. (Disk) List $29.95. **Sale $20.95.**

The Eidolon — The Eidolon is an ancient time machine that transports you to an unusual world. It allows you control over the strange creatures you encounter. You may even change the flow of time. (Disk) List $39.95. **Sale $20.95.**

The World's Greatest Football Game — This one has it all — strategy, action, three views of the field. It's in a class all its own! Finally, a football game that not only puts you on the field, but also on the sidelines in the coach's shoes. Use the "Playbook" or design your own offensive and defensive plays. Then, grab the joystick and put your strategy to the test. You control key players to run a sweep, make a tackle, throw a pass and even kick a fieldgoal. All the action and all the strategy make this your favorite football game. (Disk) List $39.95. **Sale $23.95.**

Winter Games — Experience the challenge of six winter sporting events. In the Bobsled, you're right in the tube careening along the walls. At the SkiJump, you control your form in take-off, flight and landing. In Figure Skating, timing counts for the transitions, the jumps and landings. Choreograph your own routine in Free Style Skating. Hot Dog Aerials push your agility to new heights. The Biathlon challenges your endurance in cross-country skiing. There's even an opening ceremony, complete with national anthems. One to Eight Players.
(Disk) List $39.95. **Sale $20.95.**

Jet Combat Flight Simulator — Flight simulation programs are the most requested in the country. The key to a good one is realism, the sensation of being in the cockpit — guiding the plane through take-offs, landings and air to air — air to ground combat. You are an Air Force pilot and your mission is critical. Your success in completing your orders depends on how quickly and accurately you react. Very intense — Fantastic Graphics, animation and control elements. (Disk) List $39.95. **Sale $20.95.**

Summer Games II — The original Summer Games was last Summer's No. 1 seller. Carry on the tradition with another chance to "Go for the Gold!" Introducing Summer Games II with 8 new Olympic events including Kayaking, cycling, fencing, diving, track & field, gymnastics and equastrian. The excitement of Olympic competion is present in this new version as it was in the original. Great graphics and sound effects. This one's a winner! (1 to 8 players). (Disk)
List $39.95. **Sale $20.95.**

Koronis Rift — The setting is a remote planet surrounded by radiation. To discover the secrets of the Ancients you must battle alien guards. For protection you must obtain weapons from the ruins of interplanetary civilization. (Disk) List $39.95. **Sale $20.95.**

Ballblazer — Unique split-screen, 3-D graphics give you and your opponent a first person view of the field of play. You race across the playfield in your Rotofoil trying to capture the ball and fire it through the goal before your opponent. The winner is the player with the most points at the end of the timed competition. Hold onto your joystick and keep that finger on the fire button, this is the type of two player head-to-head action you've been waiting for. Two Players. (Disk) List $29.95. **Sale $20.95.**

Super Graphics

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. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

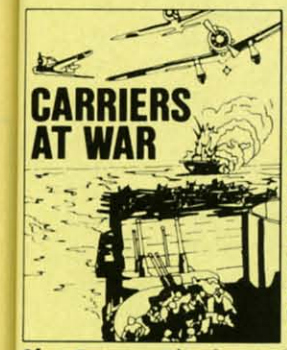
312/382-5244 to order

Reader Service No. 300

Buy any one of these Electronics Arts classics, get another from this list Free!*

• Dr. J & Larry Bird Go One-on-One • Financial Cookbook • Music Constr. Set • Pinball Constr. Set • Seven Cities of Gold • Movie Maker • M.U.L.E. • Mail Order Monsters • Racing Destruction Set

* Manufacturer Mail-in offer expires 5-31-86. These 9 titles only



Carriers At War — Recreate five crucial battles of WWII, or make up your own. Play the role of the great leaders such as MacArthur, Halsey and Yama-moto. There are 7 built-in battles such as Pearl Harbor and Midway and you must face the same challenges that arose in the original battles. Accurate battle conditions, including intelligence reports, speed of movement and fatigue levels. (Disk) List $42.95. **Sale $32.95.**

Europe Ablaze — A strategy program with you as the commander of the air force. Planning the bombing missions, plotting the course and setting the speed. The geography and weather conditions are historically accurate. Design your own scenario to keep the play exciting. Will you change the course of history? (Disk) List $42.95. **Sale $34.95.**

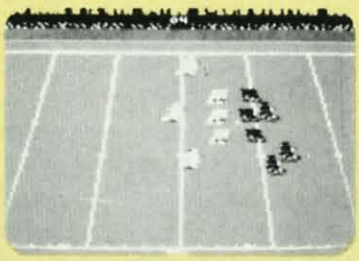
Sky Fox — Fly the unfriendly skies in this incredibly realistic flight simulation program. You are fighting to protect your home base with five kinds of weapons and working gauges. The high speed animation and breathtaking graphics are up to arcade standards. Features 5 skill levels and 15 different scenarios, each requiring a different strategy. This best selling game is a sure winner. (Disk) List $29.95. **Sale $23.95.**
Coupon $22.95.

Super Boulderdash — The exciting sequel to one of the most action packed strategy games ever. With 16 new maze levels, so complex and challenging that you will soon become a fan. The original was an award winner with characters like the enchanted wall and the boulders that fall when you least expect it. You must find the way through the caves while gathering jewels, all the time working against the clock. (Disk) List $29.95. **Sale $22.95.**

Commodore 64 SALE

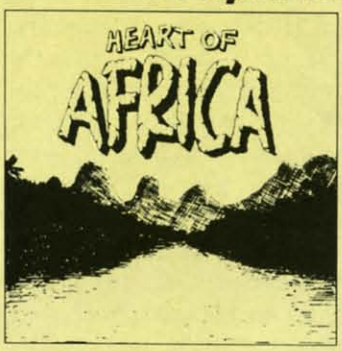
Action

Touchdown Football — Enjoy football all year long with this fast action program that brings all the elements of the game to your home computer. It's all here, the realism of great graphics and sound that put you on the turf with the NFL all stars. 100's of plays for both the offensive and defensive teams, plus the ability to play a game against another fan or the computer. (Disk) List $29.95. **Sale $22.95.**



Racing Destruction Set — Put yourself behind the wheel of the vehicle of your choice, build the racetrack and add the obstacles. Then prepare for the race of your life. This split-screen, computer slot car racing set is for two players. You'll encounter oil slicks and weapons and have to be on guard for ice, and hairpin turns. (Disk) List $29.95. **Sale $22.95.**

Graphics



Heart Of Africa — The exiting sequel to "The Seven Cities Of Gold" has you exploring the Dark Continent in the 1890's. Your mission is to find the lost temple of Ankh-Ankh. You'll encounter cannibals, slave traders, hunger and wild animals, to name a few. Historically and geographically accurate, so you gain education as you have fun. (Disk) List $29.95. **Sale $23.95.**

Movie Maker — Create your own animated movies complete with sound effects. Animate your own drawings or select from an endless supply of built-in pictures. After you're finished, save your creations on disk. Up to six characters per movie and 10 built-in movies to let your imagination run wild. (Disk) List $29.95. **Sale $23.95.**

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. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 300

COMMODORE ROOTS

MAKING HEADLINES— THE EASY WAY

By Mark Andrews

Sometimes laziness is the mother of invention in computer programming. The program we'll examine in this column, and another that will be presented next month, will demonstrate how true that statement can be. When I created this pair of programs, I was designing a title screen for a computer game, and needed a routine that would display the name of the game in big, flashy characters. But I didn't have a headline-size character set at my disposal, and I had no desire to design a complete set of giant-size characters from scratch. So I sat down, started thinking, and eventually asked myself a question that contained my answer: Why not just copy my Commodore's built-in character set from ROM to RAM, and then blow it up to several times its normal size?

With that thought in mind, I turned on my Commodore, booted up my *Merlin* machine language assembler, and went to work. The results of my efforts were the two programs that we'll look at here and next month. This month's program, BIGCHRS, can be roughly divided into two parts. Part 1 (Lines 1 to 367) copies the C-64's built-in character set from ROM into RAM, using a number of routines discussed in previous columns. Part 2 (Lines 368 to 471) uses some new techniques, along with some old ones, to print a character on the screen.

Even the so-called new techniques may not look completely new to *Commodore Roots* readers. BIGCHRS, just like the screen-printing program presented last month, is designed to display a character on the screen by copying it dot by dot from a block of data stored in memory. But unlike last month's program, BIGCHRS copies each dot *twice*—and then, each time a row of dots has been copied, copies a second row just beneath it on the screen.

The result of this operation is a character four times the size of a normal C-64 text character—twice as wide because each dot has been copied twice, and twice as deep because two rows of dots have been copied onto the screen for every single row stored in memory. Study the nested loops at the end of BIGCHRS carefully, and you'll see how easy it is to transform ordinary text characters into giant-size text characters on a C-64 screen. The task is so straightforward, in fact, that there's not much more to say about it. So now we'll move on to another topic—how to mix headline-size characters and sprites on a Commodore 64/128 screen.

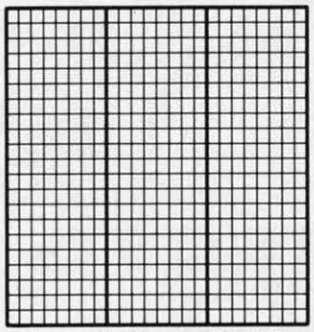


Figure 1: Sprite Bit Map

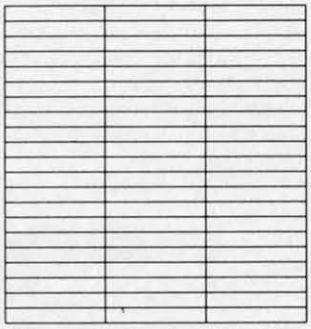


Figure 2: Sprite Byte Map

Programming sprites is also a fairly uncomplicated task—once you know how to program in assembly language. In fact, according to most of the expert C-64 programmers I know, it's much easier to program sprites using assembly language than with BASIC. That's because sprites are considerably easier to work with using binary and hexadecimal calculations than with decimal numbers.

WHAT'S IN A SPRITE

Sprites, as every *Ahoy!* reader knows by now, are graphics characters that can be created, colored, and animated fairly easily, and moved around on command, completely independent of other kinds of characters displayed onscreen. Using ordinary programming techniques, up to eight sprites can be displayed on a C-64 screen simultaneously. By convention, these eight sprites are numbered 0-7.

Sprites are made of tiny dots, just like programmable text characters; and like programmable characters can be created using standard bit-mapping techniques. But sprites are larger than text characters: a sprite can measure up to 24 horizontal screen dots wide and up to 21 vertical screen dots high.

A sprite can be displayed in any of the 16 colors available to the VIC-II chip. It is also possible to create multicolored sprites. Space does not permit me to provide instructions for programming multicolored sprites in this column, but much additional information can be found in my book *Commodore 64/128 Assembly Language Programming* (Sams, 1985), as well as in the *Commodore 64 Programmer's Reference Guide*.

Sprites, like text characters, can be expanded to twice their normal width and twice their normal height, or four times their standard size. The sprite used in this column will be an expanded one.

As stated, a sprite can measure up to 24 dots (or bits) wide, and up to 21 dots (or bits) high, for a total of 504 dots, as pictured in Figure 1.

A sprite bit can also be pictured as a *byte* map—a ma-

TOMORROW'S TECHNOLOGY IS HERE TODAY . . .

TOTALLY NEW! AN IDEA WHOSE TIME HAS COME!

THE COPY PROGRAM

- WORKS ON 1 OR 2 DRIVES
- EXACT COPIES IN ONLY 21 SECONDS
- SIMPLE-FULL BACK-UP IN A SINGLE KEY STROKE
- RELIABLE—WORKS VIRTUALLY EVERY TIME

SUPER SPECIAL **$29.95**

the 21 Second Back-up

CALL TOLL FREE! 800-468-4689

COMMODORE 1200 MODEM ONLY $139.95

PANASONIC 1091 PRINTER $239.00

COMMODORE SPECIAL

C64 COMPUTER	CALL
1541 DISK DRIVE	$169.95
C-128 COMPUTER	$259.00
1571 DISK DRIVE	$239.00

— MONITORS —
COLOR MONITOR
$148.00
12″ AMBER/GREEN
$58.00

ALSO AVAILABLE
1572 DRIVE
MOUSE
1702 MONITOR
1101 D-W PRINTER
1902 RGB MONITOR
1901 MONOCHROME MONITOR
MEMORY EXPANSION

ENHANCER 2000
FOR COMMODORE C64

ONLY $159.00

MEMOREX

MEMOREX PAPER
1000 COUNT
MEMOREX—UNIVERSAL PRINTER STAND
MEMOREX
5¼ HEAD CLEANING KIT

14.95 YOUR CHOICE

COLOR PAPER AVAILABLE

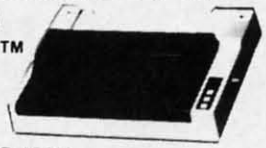
ACCESSORIES

GIM JOYSTICK $9.95
C64 POWER SUPPLY $24.95
SURGE STRIP 6 OUTLET $27.95
EMI-RMI FILTER

inland
STORAGE CASE
HOLDS 60 $10.95

SUPER PRINTER PACKAGE

FREE $100.00 WORTH OF ACCESSORIES
WITH PURCHASE OF THE
Write Hand™
FROM
COPAL



WRITE HAND 1200L
10″ CARRIAGE 120 CPS **$279.00**
OR SG-10 PRINTER $289.00

INCLUDES
- **PACKAGE OF PAPER**
- **EXTRA RIBBON**
- **PRINTER STAND**
- **XETEX INTERFACE**

OLYMPIA

OLYMPIA NP DOT MATRIX PRINTER
165 CPS, 83 NLQ $279.00
OLYMPIA RO DAISY WHEEL
14 CPS $279.00

INTERFACES

XETEC SUPER GRAPHICS $68.00
8K BUFFER, NLQ MODE, 20 FONTS
XETEC SUPER GRAPHICS JR. **$44.00**
NLQ MODE

PROFESSIONAL SOFTWARE

FLEET SYSTEM III
WITH THESAURUS $59.95

— EPYX FAST LOAD —

SUMMER GAMES I & II
WINTER GAMES
BALL BLAZERS
LUCAS FILM SERIES
FOOTBALL GAMES
$25.95 YOUR CHOICE
TOOL KIT $29.95
MICRO SOFT MULTIPLAN $49.95

— ACTIVISION —

ALTER EGO
FAST TRACKS
BORROWED TIME
COMPUTER PEOPLE
$25.95 YOUR CHOICE
HACKER
BASEBALL
FOOTBALL
BOXING

CALL FOR ALL SOFTWARE TITLES

FISHER VCR MODEL 805

WIRELESS REMOTE 10 FUNCTION
CABLE READY / 3 EVENT ONLY $299.00

DISKS

NASHUA SS $24.95
3 BOXES OF 10
NASHUA DS/DD $14.95
NASHUA MF-1 3½″ $29.95
MAXELL SS/DD $13.95
MAXELL DS/DD $21.95

TRY OUR OWN TOP QUALITY
COLOR UNIVERSAL
DISKETTES ONLY $14.95 bx

UNIVERSAL MKTG.

475 MAIN ST.
FARMINGDALE, NY 11735

RUSH ORDER HOTLINE

800-468-4689
LOCAL CALL (516) 753-0110

MASTERCARD, VISA, M.O. OR CHECK
NY ORDERS ADD SALES TAX
NO EXTRA CHARGE FOR VISA OR MASTER CARD

SEND $5.00 FOR CATALOG & ADDITIONAL SPECIALS
SCHOOL AND DEALERS WELCOME

No returns without authorization number. All products are new merchandise sold with manufacturers warrantee.
All Prices & Availability subject to change without notice. Prices are mail order only and do not include shipping & handling.

CALL FOR OUR OTHER N.Y. LOCATIONS

Reader Service No. 140

trix that measures three bytes wide by 21 bytes high, for a total of 63 bytes. Actually, the bytes that make up a sprite are in consecutive order in RAM, starting with the byte in the upper left hand corner and ending with the 63rd byte, the one in the lower right hand corner. But when a sprite appears on the screen, it looks more like Figure 2.

Although it takes only 63 bytes to form a sprite, each sprite consumes 64 bytes in RAM. The 64th byte of each sprite map is used to mark the end of its location in memory.

Sprites can be placed anywhere in free RAM, and a pointer is provided to mark the location for each sprite. Each sprite pointer is one byte long, so it takes eight bytes of RAM to hold the eight pointers needed to address the C-64's eight sprites. These eight pointers are always the last eight bytes of whatever block of RAM has been designated as screen memory. When the location of screen memory is moved, the addresses of the C-64's eight sprite pointers also change. But it's easy to find them, since they always take up the last eight bytes of whatever block of RAM is being used as screen memory.

A one-byte value is all that's ever needed to define the starting address of a sprite map, since sprites always fall into whatever 16K bank of memory is currently accessible to the VIC-II chip. That means that a sprite pointer is actually an *offset* that must be added to the starting address of the graphics bank currently in use to determine the starting address of the bit map that is to be used to form the sprite.

When the 64 is first turned on, its VIC-II chip is set to retrieve graphics information from bank 0 and to get its screen map from memory registers $0400 through $0800 (1024 through 2048 in decimal notation). At power-up time, therefore, the default address of the first sprite pointer, or sprite pointer 0, is $07FB (or 1020 in decimal notation). And the next eight bytes in RAM are the pointers for sprites 1 through 7. So the default addresses of the pointers for the 64's eight sprite pointers are memory addresses $07FB through $07FF—the last eight bytes in the block of RAM designated as screen memory.

To find the data that it needs to display a sprite, then, all the Commodore 64 has to do is look at the 8-bit value stored in the appropriate sprite pointer. When that value is added to the address of the graphics bank currently in use, the result will be the address of the bit map that must be used to define the sprite.

Before a sprite can be displayed, it must be turned on. Sprites are turned on and off with a *sprite enable register* (abbreviated SPENA) situated at memory address $D015. Each bit of the SPENA register is associated with one sprite; bit 0 is used to turn sprite 0 on and off, bit 1 is used to control sprite 1, and so on. If the bit associated with a sprite is set, the sprite is enabled. If the bit is not set, the sprite is not enabled and cannot be used.

POSITIONING SPRITES ON A SCREEN

Each of the C-64's eight sprites has two position registers: an X position register used to determine its horizontal placement on the screen, and a Y position register used to determine its vertical position. These registers are abbreviated SP0X through SP7X and SP0Y through SP7Y. In addition, there is a *most significant X position register* (abbreviated MSIGX) used to designate the horizontal positions of all eight sprites. This register is needed because a sprite can be placed in 512 possible horizontal screen positions—too many positions for an eight-bit register to keep track of. If a sprite is to be placed in a position that can be stored as a value in an 8-bit register—that is, in a position with a value of less than 255—the MSIGX register is not used. But if the horizontal position of a sprite has a value of more than 255, a bit in the MSIGX register is set. Each bit of the MSIGX register equates to the number of a sprite; bit 0 is used for sprite 0, bit 1 for sprite 1, and so on.

There is no MSIGY register because there is no need for one. A sprite can be placed in only 256 vertical positions, so only one 8-bit register per sprite is needed to handle the vertical positioning of sprites on the C-64's screen.

When you store values in a horizontal or vertical position sprite register, that value is used to determine the position of the *upper left hand corner* of the sprite. But storing a value in a horizontal or vertical position register does not ensure that a sprite will be displayed on the

screen. Of the 512 possible horizontal positions of a sprite, only positions 24 through 343 are visible on the screen. Of the 255 vertical positions available, only 50 through 249 are actually visible on the screen. It's therefore quite easy to make a sprite disappear; all you have to do is store the value of an offscreen position in its horizontal or vertical position register.

Here are the locations of all of the sprite position registers used by the Commodore 64:

SPRITE POSITION REGISTERS

Hex Address	Position Register	Hex Address	Position Register
D000	SP0X	D008	SP4X
D001	SP0Y	D009	SP4Y
D002	SP1X	D00A	SP5X
D003	SP1Y	D00B	SP5Y
D004	SP2X	D00C	SP6X
D005	SP2Y	D00D	SP6Y
D006	SP3X	D00E	SP7X
D007	SP3Y	D00F	SP7Y

D010—MSIGX (Most Significant X Position Register)

SELECTING COLORS FOR SPRITES

In addition to its two 1/8 position registers, each sprite also has a color register. The color register for sprite 0 is at memory address $D027, and the addresses of the color registers for the other seven sprites follow in consecutive order. The color address for sprite 7 is therefore at memory address $D02E.

To select the color of a sprite, all you have to do is store the standard value of one of the Commodore 64's 16 colors in that sprite's color register. Every bit that is set on the sprite's bit map will then be displayed in the selected color. Every dot that has a value of 0 will be transparent, and will not cover up anything beneath it on the screen.

EXPANDING SPRITES

By using two registers called XXPAND and YXPAND, a sprite can be expanded to twice its normal width, twice its normal height, or both. The XXPAND register is at memory address $D01D, and the YXPAND register is at $D017. Each bit in each register corresponds to a sprite number, with bit 0 controlling the size of sprite 0, bit 1 controlling the size of sprite 1, and so on.

Next month we'll take a closeup look at a sprite that has been programmed in assembly language using the techniques outlined in this column. The sprite, shaped like a heart, will be out of viewing range when the program begins. But it will slowly descend into view, parachuting down until it becomes part of a message displayed in the center of the screen.

After we have programmed and watched this descent of a sprite, we'll start exploring another fascinating topic for assembly language programmers: the music and sound capabilities of the Commodore 64. □

SEE PROGRAM LISTING ON PAGE 110

SCUTTLEBUTT

Continued from page 14

same name, the *Quick Brown Box* cartridge plugs into the expansion port of the C-64 or C-128, allowing the user to write or load up to 8K of BASIC or machine language programs that will be immediately available upon power-up. Included are a write protect switch and reset button, as well as auto-start, BASIC utilities, and ML monitor. Price is $39.00 plus $3.00 shipping.

Brown Boxes, Inc., 617-275-0090 (see address list, page 130).

PRINTER ENHANCER

Xetec's *Printer Enhancer* for parallel printers features a buffer expandable in 8K or 32K increments for a total memory of 64K or 256K respectively, selectable fonts (eight included; others available on request), intelligent printer switch for operation of one or two printers with independent selection of fonts and data, a full front control panel including controls for clear (clear buffer data), copy (print copies of buffer data), pause, printer selection, and font. Internal diagnostics include a test for the buffer RAM, front panel switches, and LED's, plus helps for printer setups.

Xetec, Inc., 913-827-0685 (see address list, page 130).

COLOR INKJET PRINTER

The Canon PJ-1080A Inkjet Color Printer produces impressive graphics at the rate of three minutes per dump (six minutes for double density). Included is a three-color ink pack rated at over three million characters, or about 7200 single density images. Price will be about $500, which includes printer, screen dump software, and an improved version of *Peripheral Vision* (the light pen-driven graphics package from the now-defunct Futurehouse).

Progressive Peripherals and Software, 303-825-4144 (see address list, page 130).

64 MODE ENHANCER

Matrix will allow you to access most of the C-128's outstanding features while in C-64 mode, including the numeric keypad, the extra top row keys, built-in 80-column RGB display, and the FAST mode running at a 2 MHz clock speed. *Matrix* also includes a built-in fast loader for C-64 mode, built-in DOS utility menu, and a high speed disk copy utility for the 1571 disk drive. This routine uses special data compaction techniques which in most cases allow for the storage of an entire double-sided 1571 disk in the C-128's RAM, permitting a full disk copy in a single pass in under three minutes.

Progressive Peripherals and Software, 303-825-4144 (see address list, page 130).

Covers operating system, 6502, more.
READER SERVICE NO. 149

LEARNING PROGRAMS

Three of the 20 Learning Technologies titles preannounced in December's *Scuttlebutt* have come available, each priced at $19.95:

Math in a Nutshell provides practice in problem solving for students in grades 4-8. The program is broken into three difficulty levels, each of which presents equations in a horizontal format and requires the student to insert the appropriate signs.

Bike Hike provides preschoolers through third graders with exercise in specific recall, visual memory, one-to-one relationships, counting, number recognition, and visual discrimination.

Pipeline lets second through sixth graders practice visualizing the whole, analyzing problems, and planning and experimenting with solutions.

Learning Technologies, 214-991-4958 (see address list, page 130).

PLUS/4 GUIDE

Face it, owners of that little charcoal gray computer—the Plus/4 has peaked. But you haven't been totally abandoned. Scott, Foresman's *Programmer's Reference Guide for the Plus/4* ($21.95) reviews programming techniques (including each BASIC 3.5 command), the machine language monitor, 6502 assembly language, and the operating system, with information not found in the Plus/4 manual. Additionally, the appendices provides the Plus/4 memory and register maps and other technical specifications.

Scott, Foresman and Company, 312-729-3000 (see address list, page 130).

UPGRADED MANAGER

The latest version of *Grade Manager III* allows C-128 owners to use their computer's additional features (separate cursor keys and numeric keypad) in C-64 mode. Among the other improvements are the option to include graphs and progress evaluation checklists in reports and to prepare worksheets listing students by room or by hour.

Smoky Mountain Software, 704-885-2516 (see address list, page 130).

KMMM USER NOTES

Wilserv Industries is living up to its name by publishing periodic User Notes for owners of its KMMM Pascal program. The latest edition, consisting of three typewritten pages, covers a range of topics of certain interest to devoted users. To receive the next issue, send a self-addressed, stamped envelope to Wilserv Industries (see address list, page 130).

RECHARGED

Batteries Included has updated the following products:

The C-128 version of *PaperClip* ($59.95) now stores up to 999 lines

Continued on page 130

TAX TIME AGAIN

Including *1040A Calculator* for the C-64, C-128, VIC 20, and Plus/4

By Cheryl Peterson

The month of April brings joyous thoughts of getting even with the government once again. That is, we all sit down and try to figure out how much of our money we should have given to the government in the last year. If we're lucky, we find that we've given Uncle Sam more than he deserves and send him a 1040 that says he owes us money. If not, we sit down with our checkbook and make up the difference.

For folks like me, who must keep track of two businesses and a husband's full-time job, April is a time of calculators, piles of forms, and a few reference works. But if you can get by with filling out the 1040A Short Form, this month's column includes (on page 113) a program that will let you do exactly that. It will work on a C-64, C-128, VIC 20, or Plus/4 (though the *Bug Repellent* line codes listed beside the program lines are for the 64 only).

In our continuing attempt to gently teach BASIC programming, we'll take a look at how the 1040A calculator works and answer a letter from a reader who asked for a modification to the *Compuloan* program we ran a few months ago.

For those whose taxes can't be done on the 1040A form, several software houses offer tax calculating programs (see list). The Lasser's tax package includes their tax guide as part of the documentation.

A SHORT PROGRAM FOR THE SHORT FORM

This month's program will take the figures that you fill in on the short form and deduce your refund or taxes due.

Several steps are involved in figuring your tax liability. You need to know how much taxable income you had for the year. Once that's figured, you can check to see how much in taxes you should have paid. You then must figure how much taxes you've already paid and subtract that from what you owe. Though this is simple math and can easily be done on a calculator, it is also a great opportunity to take a look at how math functions are handled in BASIC.

The program starts out by giving you an opportunity to request a listing of the information to put in each blank on the tax form. Another question asks (if appropriate) whether you want the information on the screen or the printer.

Before we can manipulate figures, we need to get them into the program. To do this we use lines 100-900 to input the values the program needs to make its computations. We use the variables:

W	Wages
I	Interest earned
D	Dividends earned
E	Excluded interest and dividends
U	Taxable unemployment compensation
P	Political contributions
F	Federal income tax withheld
C	Earned income credit

The GOSUB command in line 1100 sends execution down to our first equation. The IRS considers wages, interest, and dividends to be income, and all three are taxable. Certain types of interest and dividends aren't taxable, but the government still wants a full accounting of every dollar. You must declare the income and then subtract the excludable amount.

In line 2100 of our program, the variable Z is the total of the interest and dividends (Z=I+D). Most of us learned to write equations back in algebra class, and our

TAX PROGRAMS

Tax Aid ($39.95; yearly updates $13.50)
Northland Accounting
606 2nd Ave.
Two Harbours, MN 55616

Comtax ($34.95)
Milo Software
P.O. Box 569
Boston, MA 02130

Income Tax Assistant ($49.95)
H & E Computronics, Inc.
50 N. Pascack Road
Spring Valley, NY 10977

It's Tax Time ($45.00; yearly updates $10.00)
2797 Meridian Ave.
Redwood City, CA 94061

Tax Aid ($33.00; $45 for professional version; $22 and $30 for upgrades)
KSoft Co.
845 Wellner Rd.
Naperville, IL 60540

Tax Command Professional ($45.95)
P.O. Box 93104
Milwaukee, WI 53203

J. K. Lasser Your Income Tax
Simon and Schuster
1230 Avenue of the Americas
New York, NY 10020

Tax Master ($28.00)
Master Software
6 Hillery Ct.
Randallstown, MD 21133

teachers generally preferred to see the calculations on the left side of the equals sign. BASIC requires that the equations be written backwards from this. The variable which will hold the results must be stated before the equation that figures the calculation. So Z=I+D.

The second equation (Y=Z−E) uses the Z variable that we just computed. We need to subtract the excluded interest and dividend (variable E) from the total interest and dividends earned (Z). Because the government won't allow us to exclude more interest and dividend income that we've made, we use the IF/THEN statement to compare Y with zero. If it is less than zero, we reassign the value to be equal to zero.

Line 2200 computes the total income for the year (GI=W+Y+U). In this case, the gross income is equal to wages plus interest income plus taxable unemployment received.

Line 2300 demonstrates how to do a multiplication equation. Again, the variable to hold the result must come first (X2=1000*X). This calculates the deductions that go on line 11 of the tax form. Since you are given a $1000 standard deduction per exemption claimed, we must multiply 1000 times the exemptions entered for the X variable.

It is perfectly "legal" in BASIC to use the same variable before and after the equals sign. This equation could have been stated X=1000*X. However, I want to reuse the X variable later in the program when we create a printout of the numbers to fill in on the 1040A form. So I used a new variable, rather than lose the value stored in X.

This is a fairly important concept in BASIC programming. If you have figures in your program that need only be used once, you can reuse a variable to handle them all. In some programs you will see a variable used repeatedly for temporary input. In most of these programs, the input is then reassigned to a new variable or calculated for immediate use. Since I have such a small program and all my figures will be reused by the print statements at the end, I have assigned each value its own permanent variable.

Line 2400 (TN=GI−X2) computes taxable net income, or the gross income minus the exemption amount calculated in line 2300, and assigns this value to variable TN. This figure is used in line 1200 to let you know how much taxable income you need to look up in the tax tables.

Line 2500 (TP=P+F+C) calculates how much tax you've already paid (F) and how much credit (P and C) you have coming. The total of these figures is held by the variable TP (total paid) which is entered on line 14 of the 1040A.

The figure entered in program line 1200 fails to take into account advanced earned income credit which may have been claimed for the previous year, so this must be calculated into your total tax due. Line 2600 of the program takes care of this for us. And we're almost done!

Line 1500 clears the screen in preparation for the printout that may be coming up. Line 1560 checks whether a screen listing has been requested. If so the program jumps to the print routine just after the "Open the printer channel" statement. In this way we can still use the routine without actually creating a hard copy.

To really understand this line you must realize that IF/THEN statements are only executed if the exact conditions specified are true. In this case we are testing to see that Q2$ is not equal to "P" *and* that Q$ is not equal to "N". Both conditions must be true in order for the program to skip to line 1930. We are checking to see that the user has said "No, I don't want the printer turned on" and "Yes, I do want the listing for filling in the blanks."

Stop a moment and think of another, shorter way to achieve the same goal. There is one, and I'll tell you what it is at the end of the column.

If the screen listing has not been requested, the program continues to line 1570 which checks to see if a "hard copy" printout has been requested. If so, the program jumps down to line 1921. The printer channel is opened, the information is printed, and line 1997 checks to see whether we are creating a hard copy or not. If so, it assigns the value 1 to the variable CL and returns to line 1600.

The routine that starts at line 1600 is used to figure out whether you owe tax or have a refund coming by comparing the values in TP and TT. It prints the appropriate message to the screen.

Lines 1797, 1897, and 1917 check to see if the channel to the printer is open. If not, they jump to an end state-

IT DON'T GET NO BETTER THAN THIS

EPYX

Summer Games	23.95
Summer Games II	23.95
Winter Games	23.95
Pitstop II	23.95
Eidolon	23.95
Rescue on Fractalus	23.95
Fast Load	23.95
Koronis Rift	23.95
Ball Blazer	23.95
Greatest Football Game	23.95
Multiplan 64/128	40.95
Program Tool Kit	24.95

INFOCOM

Wishbringer	23.95
Enchanter	23.95
Zork 1-2-3	25.95
Suspect	23.95
Witness	23.95
Sorcerer	25.95
Cutthroats	23.95
Infidel	25.95
Spellbreaker	28.95
ALL INVISICLUES	6.00

ACCESS

Beachead II	21.95
Raid over Moscow	23.95
Mach 5	21.95

SIERRA

Wiztype	21.95
Champ Boxing	21.95
Stunt Flyer	16.95
Homeword	29.95
Homeword w speller	40.95
Ultima II	35.95

RIBBONS

MPS 803	6.95
SG10-15	2.35
Oki 82,92,83,93	2.35
Oki 182, 192	9.75
1526	6.50
Legend 880, 1080, 808	6.50
Epson FX, RX, MX, LX 80	4.50
Epson FX, RX, MX, 100	6.50
Epson 85	6.50
Epson 185	6.50
Brother 15,25,35	4.99
Juki 6000,6100	2.50
1525	6.75
Banana	7.95
MPS 801	7.95
Axiom Elite 5 LCD	7.99

MIN 6 Ribbons per order of any type.

BRODERBUND

Print Shop	27.95
P/S Library 1, 2,&3	17.95
Whistlers Bros.	18.95
Karateka	19.95
Music Shop	27.95
Bank St. Writer	29.97
Stealth	18.95

ACTIVISION

Slot Car Construction	21.95
Cross Ctry Road Race	19.95
Master of the Lamps	19.95
On Track Racing	19.95
Countdown to Shutdown	19.95
Gamemaker	24.95

BATTERIES INCLUDED

Home Pak	29.95
Consultant	41.95
Paper Clip 64/128	41.95
P/C w/Spell 64/128	54.95

MICRO LEAGUE

Micro League Baseball	31.95
Managers Disk	31.95
Team Disks	14.95

MICROPROSE

Crusade in Europe	24.95
F-15 Strike Eagle	21.95
Kennedy Approach	21.95
Silent Service	21.95
Acro Jet	21.95
Gunship	Call
Decision in Desert	24.95

TIMEWORKS

Data Manager II	29.95
Data Mgr. II 128	42.95
Electronic Checkbook	16.95
Money Manager	16.95
Evelyn Wood Reading	29.95
SwiftCalc 128 w/Sideways	42.95
Word Writer 128	42.95
Swift Calc w/Sideways 64	29.95
Sideways	19.95
Word Writer w/Speller	29.95

ACCOLADE

Fight Night	19.95
Dam Busters	19.95
Hard Ball	19.95
Law of the West	19.95

DISKS
Guaranteed Error Free

Generic No Label Pack of 25 w/sleeves	18.75
Data Case (50)	11.95
Drive Head Cleaning Kit	11.95

MISC.

Kung Fu Masters	23.95
Spy vs Spy	18.95
Spy vs Spy II	18.95
Tapper	27.95
Doodle	23.95
Jet	27.95
Typing Tutor III	24.95
I am the 64	18.95
CompuServe Starter	22.95
Superbase 64 V.2	46.95
Superbase 128	56.95
Practicalc 64	18.95
Tax Advantage '85	36.95
Practifile 64	18.95
Sargon III	31.95
Koala Printer	16.95
Super Script	Call
Super Bowl Sunday	21.95
Kung FU	19.95
Karate Champ	22.95
C-64 Koala Pads	42.95
Flight Sim. II	30.95
Fl. Sim. Scenery	13.95
Intl. Hockey	16.95
Halley Project	23.95
Ultima III	35.95
Ultima IV	39.95
Superman	19.95
Carriers at War	33.95
Star Rank Boxing	19.95

COMPLETE LINE OF APPLE & IBM SOFTWARE

ACCESSORIES & DISKS

Disk Notchers	6.50
Drive and Printer Cables (specify length)	1.20 per foot (6 ft. min.)
Kraft Joysticks	9.95

Buy any *6* Software Titles

at our dynamic prices
and purchase either a

C-128 for $265.00

or

1571 Disk Drive for $240.00

Plus $20.00 S&H on complete order
C.O.D. M/C Visa Int'l restrictions apply
See ordering info

TRY OUR GREAT EXPRESS SERVICE

****SPECIAL INTRODUCTORY OFFER****

FREE 1541 Disk Drive Cleaning Program With Every Order

AMERICAN INTERNATIONAL COMPUTER PRODUCTS

P.O. Box 1758, STATEN ISLAND, NY 10314
PHONE ORDERS 718-351-1864 Reader Service No. 146

Ordering Info: In stock items shipped within 24 hrs. for prepaid orders (cert. chk. - visa - m/c). C.O.D. orders accepted please add $4.00 must be cash or cert. chk. only. NYS residents add 8¼% sales tax. Visa & Master Card add 4%. All orders add 3.50 shipping and handling. International Orders add $5.00 $ S & H. School inquiries invited. Personal checks allow 2 wks. for clearance. Availability and pricing subject to change without notice. Orders over $150.00 receive free shipping in continental U.S. Make all checks payable to: American International Computer Products. Defective Mdse: Handled by authorized Warranty Service Center in your area. Hardware special subject to availability.

SUPER HARDWARE PACKAGE

C-128 Computer
1571 Disk Drive
Super Epson Compatible
NLQ PRINTER

$685.00 S & H Included
PRINTER INTERFACE ADD. $44.95

ment. If the channel is open, the next line closes it.

The rest of the program (lines 1930-1990) just prints out the information that needs to be put into your 1040A.

I DO GET LETTERS!

Helen Eichman of Oregon wrote me recently about a problem she was having with the *Compuloan* program that accompanied my column several months ago.

Her letter reads:

Sirs:

I just finished keying in *Compuloan* from the Nov. '85 issue. I'm convinced there is an error in the math formula somewhere. I used the *Bug Repellent* program and it says I typed correctly.

Below is the starting of the program for a $90,000 loan for 25 years at 11.5%. Below that is one of the printouts of another similar program. You can see the results are not the same and in fact the *Compuloan* program adds the first month's interest to the principal and fails to subtract the payment. It goes on to figure the interest on the new loan amount and of course continues to be wrong.

Also included is a printout of the program from my disk. Is the program wrong or am I?

```
loan payment calculator
amount financed $ 90000.00
interest rate 11.5%
# of payments per year 12
total number of payments is 300
each payment is 914.82
total interest paid is $ 184446.61
total amount paid is $ 274446.61
```

bal	payment	interest	principal
99862.50	914.82	862.50	52.32
90810.17	914.82	870.26	44.55
90765.62	914.82	869.83	44.98

The other program's information is

month	balance	interest
1.	89947.67	862.50
2.	89894.84	862.00
3.	89841.50	861.49
4.	89787.65	860.98

I'd appreciate any help because I like the printout better but need accuracy.

The printout of the program showed that she had entered the critical lines correctly. Gee, that *Bug Repellent* program does work.

I must be from the only state in the union that requires the first interest payment in advance. Unfortunately, I misunderstood the local real estate agent when he said that this interest payment could be added to the amount of the loan and be figured as part of the loan. So, as she points out, my calculations do come up different.

I think most of you will benefit from her letter, since the problem is easily solved and most of you would probably prefer that it ran her way. Merely delete line 687 from the program and it will work just as requested.

For those who haven't entered the program and would like a copy of it, it's available (the modified version, that is) on Viewtron in the newly activated download section. The download section is located on page 4000, or you can type the keywords C64 DOWNLOAD to get there. Named *Loan Calculator*, it is in the business programs section under accounting programs. It doesn't take very long to download.

After deleting line 687, I tried Helen's sample run. The initial figures are all the same. The payment schedule looked like this:

Balance	Payment	Interest	Principle
90000.00	914.82	862.50	52.32
89947.67	914.82	861.99	52.82
89894.85	914.82	861.49	53.32
89841.52	914.82	860.98	53.84

I want to thank those of you who have been corresponding with me through Viewtron. I want the column to be as useful to new users as possible. I'm always looking for ideas to use. I've received quite a few suggestions, but most of them don't fall into the beginning (or even intermediate) user category.

Look for me in the For Starters SIG of Viewtron. I will be available in our CB section on Saturday nights from 8 to 11 EST, hiding behind the handle Cherp!. Although I frequently spend my evenings in CB, this is my scheduled night to appear. Hope to see you there.

The answer to the earlier puzzle is IF Q2$="S" GOSUB 1930. □ *SEE PROGRAM LISTING ON PAGE 113*

COMMODARES

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, where appropriate. Be sure to tell what makes your solutions unique or interesting, if they are.

Programs on diskettes (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 #28-1: PRINT FORMATTER

This problem is from the combined suggestions of Francisco Vellejo (Bayamon, PR) and Robert Croswell (Trappe, MD). Francisco wants to be able to show any zeroes behind the decimal point when printing an amount of money. For example, 15 and 2.4 should be printed as 15.00 and 2.40. Robert suggested a routine to round off any decimal value properly to the nearest hundredth. Your challenge is to come up with the handiest way to take any number, round it to the nearest hundredth, and display the result always using two decimal places. (C-128 users have it made!) Your solution can be more general, if desired.

For an advanced challenge, Chuck Slotter (Philadelphia, PA) suggested creating a PRINT USING routine in machine language. For example, this sequence of instructions:

```
V=12.3456 : F$=" ###.##" : PRINT USING F
$; V
```

produces the result 12.35 on the C-128. What can you come up with?

PROBLEM #28-2: SIMPLE SCROLLER

The user inputs a message which is then scrolled horizontally from right to left on one line of the screen. Letters appear at the right edge of the screen. They move across to the left edge dragging the rest of the message behind. They disappear at the left edge only to reappear later at the right edge. The message may be up to 80 characters long.

PROBLEM #28-3: MATH MYSTERY

Here is a good math and/or string challenge from Ted Grondski (Springfield, MA). Simply find each seven-digit number which has all of the following properties: a) divisible by eleven, b) contains no zeroes, and c) has no two digits alike.

PROBLEM #28-4: ELEGANT ENCODER

We can count on Jim Speers (Niles, MI) to come up with interesting challenges. Here's another one. Write a program which encodes a word such that the numeric value of each letter is the sum (modulo 29) of the numeric values of the other letters in the word. (Modulo 29 simply means to take any result greater than 29, divide it by 29, discard the whole-number quotient, and keep the remainder. For example, 123 MODULO 29 = 7 since 123 divided by 29 equals 4 and 7/29. The 4 is ignored. The 7 in the numerator of the fraction is the result. Any number MODULO 29 is between 0 and 28, inclusive.)

We are adding these characters to the encoded alphabet with the following values to make a total of 29 characters: @=0, [=27, £=28. The values of A through Z are 1 through 26 respectively. Consider the word "CAT" with letter values 3, 1, and 20. The C is replaced by U, since the sum of the other two letters is 21, and U has a value of 21. The A is replaced by W, since 3+20=23. The T is replaced by D, since 3+1=4. If the user types CAT, the computer responds UWD. CATS becomes KMWX. Why 29 letters? Because 29 is prime. Next month, you will use that fact to solve the second part of this problem. You guessed it—a decoding program!

This month we will look at solutions to the *Commodares* from the December 1985 issue of *Ahoy!* Before we begin, here is a neat solution to *Problem #23-2: Rooting Routine* all the way from Ludwigsberg, Federal Republic of Germany. Although we saw solutions to this problem last month, the following program from Stephan Fassbender is different from the others, and it came with a nice mathematical explanation.

```
1 REM
2 REM PROBLEM #23-2 :
3 REM  ROOTING ROUTINE
4 REM SOLUTION BY
5 REM  STEPHAN FASSBENDER
6 REM
10 INPUT "INTEGER ";X
20 UG=INT(X/256)
```

```
30 FOR N=UG TO X
40 IF N*(N+1)<X THEN NEXT N
50 PRINT X,"INTEGER SQUARE ROOT = ";N
60 GOTO 10
```

If you are interested in the math, send a self-addressed stamped envelope to me at the address above with your request clearly specified. This is further proof that BASIC is a universal language.

Chuck Slotter (Philadelphia, PA) responded to the challenge to find a keyword which, when misspelled, still works. Chuck points out that if you type RUN over the READY. prompt, you end up with RUNDY. If the first line of your program is line 0, then RUNDY. works as desired!

The solution from Jim Speers (Niles, MI), who originally posed the question, was the END statement. If it is misspelled within a program, you will get a SYNTAX ERROR response, but the program will end all the same. Be sure to add these to your volumes of "Computer Trivia"!

Problem #24-1: Printing Permutations, suggested by Bill O'Rourke (Crystal River, FL), is a pretty tough problem. Given a quantity, the computer prints out every possible arrangement of that many numbers. The shortest solution is listed below from Allan Flippin (San Jose, CA).

```
1 REM
2 REM PROBLEM #24-1 :
3 REM  PRINTING PERMUTATIONS
4 REM SOLUTION BY
5 REM  ALLAN FLIPPIN
6 REM.
100 CLR:INPUT N:DIMC(N-1):FOR I=1 TO N:S
$=S$+CHR$(48+I):NEXT
110 I=N-1:PRINT S$,
120 S$=LEFT$(S$,I-1)+RIGHT$(S$,N-I)+MID$
(S$,I,1)
130 C(I)=C(I)+1:IF C(I)<=N-I GOTO 110
140 C(I)=0:IF I>1 THEN I=I-1 : GOTO 120
```

Allan said that his program could be modified to allow the input of letters in order to solve anagrams and other word games. Simply use the following for line 100:

```
100 INPUT S$ : N=LEN(S$) : DIM C(N-1)
```

Solutions to this problem also came from Mark Maples (Powell, TN), Matt Shapiro (Fort Lee, NJ), David Hoffner (Brooklyn, NY), Jim Speers (Niles, MI), Frank T. Smith (Wilmington, DE), James Borden (Carlisle, PA), and John Gidusko (Fern Park, FL). John Gidusko mentions that you may enter Commodore graphics symbols as well as text or numbers into his program and generate an interesting screen display. Use line 100 above and hold down the SHIFT or COMMODORE keys when you enter a string.

My approach to this problem was to implement an algorithm contained in the book *Algorithms* (Addison-Wesley, 1983) by Robert Sedgewick. The algorithm interested me because it is recursive, meaning that it calls itself. Here is a slightly modified statement of the algorithm called "visit":

```
procedure visit(k)
  begin
  now=now+1 : val(k)=now
  if now=n then print all values of val(k)
  for t=1 to n
    if val(t)=0 then call visit(t)
  next t
  now=now-1 : val(k)=0
  end.
```

The routine is initially called with k=0, now=−1, n equals the number of items to be permutated, and all elements of val( ) are 0 (i.e. k=0 : now=−1 : n=3 : gosub visit). The BASIC language does not handle recursive calls the way that some other languages such as Pascal and C do. Consequently my program had to keep track of its current "depth" into the visit routine.

The argument k in **procedure** visit(k) is like the argument in a BASIC function definition. When the procedure is called (in the IF/THEN statement), the variable k throughout the procedure is replaced by t. I will leave it as a challenge to you to implement this algorithm in BASIC or any other language. I will show my solution

next month.

Choosing from the dozens of solutions to *Problem #24-2: Pyramidal Printout* proposed by Nolan Whitaker (Jeffersonville, KY) was not an easy task. Many were very similar, varying in degrees of user friendliness, flexibility in printing to the screen or a printer, and the selection of BASIC functions used. The solution from Ron Weiner (Levittown, PA) listed below is representative of the approach taken by many readers.

```
1 REM
2 REM PROBLEM #24-2 :
3 REM  PYRAMIDAL PRINTOUT
4 REM SOLUTION BY
5 REM  RON WEINER
6 REM
10 OPEN 4,3:INPUT"OUTPUT TO SCREEN OR PR
INTER (S/P) ";A$:IF A$="S"GOTO 30
20 CLOSE 4 : OPEN 4,4
30 INPUT "ENTER ANY WORD WHICH HAS SAME
FIRST AND LAST LETTERS";W$
40 PRINT#4,SPC(LEN(W$)-1)LEFT$(W$,1)
50 FORI=2 TO LEN(W$)-1:PRINT#4,SPC(LEN(W
$)-I)MID$(W$,I,1)SPC(I*2-3)MID$(W$,I,1)
60 NEXT:FOR I=1 TO LEN(W$):PRINT#4,MID$(
W$,I,1)" ";:NEXT:PRINT#4:CLOSE4
```

Ron's program shows a handy way of diverting the output to a printer when desired. Although Ron chose the word "Senseless" to use as an example to show that his program works, he admitted that he had fun writing the program. And that's the idea behind *Commodares.*

A second example for *Pyramidal Printout* is shown from Lonnie Welch. His program runs on a C-64 with the COMAL 2.01 cartridge.

```
1 REM
2 REM PROBLEM #24-2 :
3 REM  PYRAMIDAL PRINTOUT
4 REM COMAL SOLUTION BY
5 REM   LONNIE WELCH
6 REM
10 :  INPUT W$
20 :  PAGE
30 :  L:=LEN(W$)
40 :  FOR T:=1 TO L-1 DO
50 :     PRINT AT T,21-T: W$(T)
60 :     PRINT AT T,19+T: W$(T)
70 :  ENDFOR T
80 :  CURSOR L,21-L
90 :  FOR T:=1 TO L DO PRINT W$(T)," ",
```

This COMAL solution shows the power of a PRINT AT function, although such a solution would not work on a printer. Evidently w$(t) refers to the "t"th character of the string w$.

There were a few solutions to *Problem #24-3: Fractional Fun* which was originally suggested by Jim Speers (Niles, MI). The problem was to find the fractional equivalent of a repeating decimal. The programs from Allan Flippin (San Jose, CA) and Ronald Jordan (Florence, OR) both determined the non-repeating and repeating parts of the input decimal by analyzing it. Ronald's program is listed below. Notice his use of the ON/GOTO as a conditional statement. The remaining statements on line 10 would have to be on a separate line if an IF/THEN statement were used instead.

```
1 REM
2 REM PROBLEM #24-3 :
3 REM  FRACTIONAL FUN
4 REM SOLUTION BY
5 REM  RONALD JORDAN
6 REM
10 INPUT"DECIMAL";N$:ON -(LEFT$(N$,1)<>"
.") GOTO 10:L=LEN(N$):FORI=L TO 1STEP-1
20 R$=MID$(N$,I):R=LEN(R$):L$=MID$(N$,I-
R,R):IF L$<>R$ THEN NEXT
30 F$=MID$(N$,2,L-R-R-1):F=LEN(F$):IF F>
2 THEN I=I-1:GOTO 20
40 D=INT(10^R-1)*10^F:N=VAL(R$)+VAL(F$)*
INT(10^R-1)
50 FOR I=2 TO 41:IF N/I=INT(N/I) THEN IF
 D/I=INT(D/I) THEN N=N/I:D=D/I:GOTO 50
60 NEXT:PRINT N"/"D
```

The other solutions, including one from Matt Shapiro (Fort Lee, NJ), required the user to specify the repeating and non-repeating parts separately. The most general solution came from Tom Gantner (Dayton, OH). His program uses Euclid's Algorithm in lines 130-160 to reduce the fraction to lowest terms. Other readers used a similar method.

```
1 REM
2 REM PROBLEM #24-3 :
3 REM  FRACTIONAL FUN
4 REM SOLUTION BY
5 REM  TOM GANTNER
6 REM
30 INPUT"NON-REPEATING DIGITS: 0.";A$
40 INPUT"    REPEATING DIGITS: ";B$
50 IF B$="" THEN B$="0"
60 L=LEN(A$):M=LEN(B$)
70 IF L+M>9 THEN PRINT"TOO MANY DIGITS":
GOTO 30
80 P=VAL(A$+B$)-VAL(A$):Q$=""
90 FOR I=1 TO M:Q$=Q$+"9":NEXT I
100 IF L=0 GOTO 120
110 FOR I=1 TO L:Q$=Q$+"0":NEXT I
120 Q=VAL(Q$):A=P:B=Q
130 QU=INT(A/B):RM=A-B*QU
140 IF RM=0 THEN D=B:GOTO 160
150 A=B:B=RM:GOTO 130
160 P=P/D:Q=Q/D
170 PRINT:PRINT"0."+A$+"'"+B$+"' =";P;"/
";Q
180 END
```

There were two approaches to *Problem #24-4: Sector Status.* The problem was to determine whether a specific track and sector of a disk was being used or not. The less desirable approach was to send a "block-allocate" command to the disk drive. The error channel was checked to see if an error number 65 was received, indicating that the sector had already been allocated ("used"). The method should work, but if the sector was previously unused, the block-allocate command now marks it as being used. Consequently a block-free command must be sent to de-allocate that sector.

According to the *1541 User's Guide* (Datamost, 1984) by Dr. Gerald Neufeld, the block-allocate command works correctly only if the designated block is free in the Block Availability Map (BAM). Otherwise the entire sector is allocated. A Validate command is needed to restore these marked but unused sectors.

A program which is simply to *read* the availability of a given track and sector on a disk should not have the capability of modifying the disk (or even the BAM image in memory) in any way. Therefore we will look at the solutions which merely read and interpret the BAM.

After the disk drive is initialized, the BAM is read from the disk and stored in a buffer at addresses $0700 through $07FF (the $ indicates hexadecimal values). Each track has three bytes associated with it which provide a bit map representation of the sectors within the track. Since no track contains more than 20 sectors, three bytes (24 bits) is more than adequate.

The location of the byte corresponding to track T and sector S is at address $0700 plus (4*T + INT(S/8) + 1). The bit corresponding to sector S has a value (2 (S AND 7)). A bit value of 0 indicates that the corresponding sector is allocated, and a bit value of 1 means the sector is free.

The disk Memory-Read command has the format "M-R"CHR$(low byte)CHR$(high byte). "Low byte" and "high byte" refer to the least significant and most significant bytes of the desired memory address. For example, address $0705 has a most significant byte of $07 and a least significant byte of $05.

With the explanation behind, here's a program which implements the above sequence, from Barry Parris (Gaffney, SC). It is very similar to programs received from Rick Nash (Millersburg, OH), Allan Flippin (San Jose, CA), and Matt Shapiro (Fort Lee, NJ).

```
1 REM
2 REM PROBLEM #24-4 :
3 REM  SECTOR STATUS
4 REM SOLUTION BY
5 REM  BARRY PARRIS
6 REM
10 OPEN15,8,15,"I0":INPUT"TRACK, SECTOR"
;T,S
20 PRINT#15,"M-R"CHR$(4*T+INT(S/8)+1)CHR
$(7):GET#15,A$:CLOSE15
30 A=ASC(A$+CHR$(0)):IF (A AND 2^(S AND
7)) THEN PRINT"UNUSED":END
40 PRINT"USED"
```

The "I0" command in the OPEN statement reads the BAM into memory. The other statements perform the remaining operations. You should add some error-checking if you are writing a program that someone else might use. The following lines from Jim Speers (Niles, MI) check for valid track and sector values, and could be added to the program above.

```
1 REM
2 REM PROBLEM #24-4 :
3 REM  SECTOR STATUS
4 REM ERROR CHECKING BY
5 REM    JIM SPEERS
6 REM
7 REM ADD THESE LINES TO
8 REM THE PREVIOUS PROGRAM
9 REM
11 IF T<36 AND S<17 THEN 20
12 IF S>20 OR T>35 THEN 50
13 IF S=17 AND T>30 THEN 50
14 IF S=18 AND T>24 THEN 50
15 IF S>18 AND T>17 THEN 50
50 PRINT"IMPROPER TRACK OR SECTOR NUMBER
"
```

The world of the 1541 disk drive can be very mysterious. Thanks to Larry West (Prince George, BC), Jim Cagle, Jose Molina (Homestead, FL), Carlton Burton (Easton, TX), and Peter Spearing (Northfield, OH), in addition to the people mentioned earlier for their work on this problem and for their very informative descriptions of the solutions to this problem.

Other readers who submitted valid solutions to *Commodares* this month and have not already been mentioned include Jack O. Foley (Goldsboro, NC), Douglas Underwood (Walla Walla, WA), Richard MacDonald (Danbury, CT), John Freel (Merced, CA), Charles Grady(Cleveland, TN), Daniel Daugherty (Parsons, TN), Paul L. Ramos, Steven Steckler (Columbia, MD), James Killman (Memphis, TN), Gary Hudach (Youngstown, OH), Wallace Leeker (Lemay, MO), Brian Wilcox (Coldwater, OH), B. Gregg Price (Beaumont, TX), Mark Breault (Brandon, MAN), L. W. Brenneman (Erie, PA), Tom Frankson (Bolingbrook, IL), David Dill (Giddings, TX), and Pedro H. Ortiz (Bayamon, PR).

Your comments and challenges are always welcome. Some readers have sent good suggestions for *Commodares,* but they are problems which we have used before. If there is a way to modify your suggestion so that it is not an exact duplicate of a previous problem, we will use it. Since some of you readers have more than two additional years of programming experience under your belts now than when you solved the first *Commodares,* perhaps you would enjoy redoing some of the earlier challenges. Let me know what you think. Until next month, have fun with these challenges. □

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 VIC 20 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	

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** This page and page 105 explain these codes and provide other essential information on entering *Ahoy!* programs. Read these pages **before** entering programs!

BUG REPELLENT

This program will let you debug any *Ahoy!* program. Follow instructions for VIC 20 (cassette or disk) or C-64.

VIC 20 VERSION

By Michael Kleinert and David Barron

For cassette: type in and save the *Bug Repellent* program, then type RUN 63000[RETURN]SYS 828[RETURN]. If you typed the program properly, it will generate a set of two-letter line codes that will match those listed to the right of the respective program lines.

Once you've got a working *Bug Repellent*, type in the program you wish to check. Save it and type the RUN and SYS commands listed above once again, then compare the line codes generated to those listed in the magazine. If you spot a discrepancy, a typing error exists in that line. Important: you must use exactly the same spacing as the program in the magazine. Due to memory limitations on the VIC, the VIC *Bug Repellent* will register an error if your spacing varies from what's printed.

You may type SYS 828 as many times as you wish, but if you use the cassette for anything, type RUN 63000 to restore the *Repellent*.

When your program has been disinfected you may delete all lines from 63000 on. (Be sure the program you type doesn't include lines above 63000!)

For disk: enter *Bug Repellent*, save it, and type RUN:NEW [RETURN]. Type in the program you wish to check, then SYS 828.

To pause the line codes listing, press SHIFT.

To send the list to the printer type OPEN 4,4:CMD 4:SYS 828[RETURN]. When the cursor comes back, type PRINT#4:CLOSE 4[RETURN].

```
·63000 FORX=828TO1023:READY:POKEX,Y:NEXT:END   AC
·63001 DATA169,0,133,63,133,64,165,43,133,251  JL
·63002 DATA165,44,133,252,160,0,132,254,32,228 DF
·63003 DATA3,234,177,251,208,3,76,208,3,230     OE
·63004 DATA251,208,2,230,252,169,244,160,3,32   OH
·63005 DATA30,203,160,0,177,251,170,230,251,20
 8                                              KO
·63006 DATA2,230,252,177,251,32,205,221,169,58  JJ
·63007 DATA32,210,255,169,0,133,253,230,254,32  OK
·63008 DATA228,3,234,165,253,160,0,170,177,251  LG
·63009 DATA201,32,240,6,138,113,251,69,254,170  BP
·63010 DATA138,133,253,177,251,208,226,165,253
 ,41                                            DD
·63011 DATA240,74,74,74,74,24,105,65,32,210     EK
·63012 DATA255,165,253,41,15,24,105,65,32,210   FO
·63013 DATA255,169,13,32,210,255,173,141,2,41   PK
·63014 DATA1,208,249,230,63,208,2,230,64,230    CB
·63015 DATA251,208,2,230,252,76,74,3,169,236    KH
·63016 DATA160,3,32,30,203,166,63,165,64,32     DP
·63017 DATA205,221,169,13,32,210,255,96,230,25
 1                                              EL
·63018 DATA208,2,230,252,96,0,76,73,78,69       OI
·63019 DATA83,58,32,0,76,73,78,69,32,35         FG
·63020 DATA32,0,0,0,0,0                         LE
```

C-64 VERSION

By Michael Kleinert and David Barron

Type in, SAVE, and RUN the *Bug Repellent*. Type NEW, then type in or LOAD the *Ahoy!* program you wish to check. When that's done, SAVE your program (don't RUN it!) and type SYS 49152 [RETURN].

To pause the listing depress and hold the SHIFT key.

Compare the codes your machine generates to the codes listed to the right of the respective program lines. If you spot a difference, an error exists in that line. Jot down the number of lines where contradictions occur, LIST each line, spot the errors, and correct them.

```
·5000 FORX=49152TO49488:READY:POKEX,Y:NEXT:END GJ
·5001 DATA32,161,192,165,43,133,251,165,44,133 DL
·5002 DATA252,160,0,132,254,32,37,193,234,177  DB
·5003 DATA251,208,3,76,138,192,230,251,208,2    OF
·5004 DATA230,252,76,43,192,76,73,78,69,32      KN
·5005 DATA35,32,0,169,35,160,192,32,30,171      CA
·5006 DATA160,0,177,251,170,230,251,208,2,230   CE
·5007 DATA252,177,251,32,205,189,169,58,32,210  JE
·5008 DATA255,169,0,133,253,230,254,32,37,193   CL
·5009 DATA234,165,253,160,0,76,13,193,133,253   NB
·5010 DATA177,251,208,237,165,253,41,240,74,74  MB
·5011 DATA74,74,24,105,65,32,210,255,165,253    EP
·5012 DATA41,15,24,105,65,32,210,255,169,13     GH
·5013 DATA32,220,192,230,63,208,2,230,64,230    AN
·5014 DATA251,208,2,230,252,76,11,192,169,153   NG
·5015 DATA160,192,32,30,171,166,63,165,64,76    BF
·5016 DATA231,192,96,76,73,78,69,83,58,32       EP
·5017 DATA0,169,247,160,192,32,30,171,169,3     PJ
·5018 DATA133,254,32,228,255,201,83,240,6,201   FK
·5019 DATA80,208,245,230,254,32,210,255,169,4   FL
·5020 DATA166,254,160,255,32,186,255,169,0,133  CL
·5021 DATA63,133,64,133,2,32,189,255,32,192     GC
·5022 DATA255,166,254,32,201,255,76,73,193,96   NN
·5023 DATA32,210,255,173,141,2,41,1,208,249     NH
·5024 DATA96,32,205,189,169,13,32,210,255,32    IM
·5025 DATA204,255,169,4,76,195,255,147,83,67    KC
·5026 DATA82,69,69,78,32,79,82,32,80,82         DC
·5027 DATA73,78,84,69,82,32,63,32,0,76          ML
·5028 DATA44,193,234,177,251,201,32,240,6,138   GN
·5029 DATA113,251,69,254,170,138,76,88,192,0    JK
·5030 DATA0,0,0,230,251,208,2,230,252,96        NA
·5031 DATA170,177,251,201,34,208,6,165,2,73     DM
·5032 DATA255,133,2,165,2,208,218,177,251,201   JA
·5033 DATA32,208,212,198,254,76,29,193,0,169    FM
·5034 DATA13,76,210,255,0,0,0                   PA
```

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 previously 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.

```
·5 POKE53280,12:POKE53281,11·                   LL
·6 PRINT"[CLEAR][c 8][RVSON][15" "]FLANKSPEED[
 15" "]";                                       ED
·10 PRINT"[RVSON][5" "]MISTAKEPROOF ML ENTRY P
 ROGRAM[6" "]"                                  MC
·15 PRINT"[RVSON][9" "]CREATED BY G. F. WHEAT[
 9" "]"                                         DM
·20 PRINT"[RVSON][3" "]COPR. 1984, ION INTERNA
```

```
   TIONAL INC.[3" "]"                                              DH
 ·30 FORA=54272TO54296:POKEA,0:NEXT                                IM
 ·40 POKE54272,4:POKE54273,48:POKE54277,0:POKE5
   4278,249:POKE54296,15                                           NH
 ·70 FORA=680TO699:READB:POKEA,B:NEXT                              KO
 ·75 DATA169,251,166,253,164,254,32,216,255,96                     HJ
 ·76 DATA169,0,166,251,164,252,32,213,255,96                       JB
 ·80 B$="STARTING ADDRESS IN HEX":GOSUB2010:AD=
   B:SR=B                                                          HC
 ·85 GOSUB2520:IFB=0THEN80                                         FO
 ·86 POKE251,T(4)+T(3)*16:POKE252,T(2)+T(1)*16                     KE
 ·90 B$="ENDING ADDRESS IN HEX":GOSUB2010:EN=B                     IF
 ·95 GOSUB2510:IFB=0THEN80                                         FP
 ·96 POKE254,T(2)+T(1)*16:B=T(4)+1+T(3)*16                         MN
 ·97 IFB>255THENB=B-255:POKE254,PEEK(254)+1                        GE
 ·98 POKE253,B:PRINT                                               HN
 ·100 REM  GET HEX LINE                                            IL
 ·110 GOSUB3010:PRINT": [c P][LEFT]";:FORA=0TO8                    FG
 ·120 FORB=0TO1:GOTO210                                            MD
 ·125 NEXTB                                                        ME
 ·130 A%(A)=T(1)+T(0)*16:IFAD+A-1=ENTHEN310                        LH
 ·135 PRINT" [c P][LEFT]";                                         IK
 ·140 NEXTA:T=AD-(INT(AD/256)*256):PRINT" "                        PD
 ·150 FORA=0TO7:T=T+A%(A):IFT>255THENT=T-255                       LK
 ·160 NEXT                                                         IA
 ·170 IFA%(8)<>TTHENGOSUB1010:GOTO110                              FK
 ·180 FORA=0TO7:POKEAD+A,A%(A):NEXT:AD=AD+8:GOT
   O110                                                            MN
 ·200 REM    GET HEX INPUT                                         AB
 ·210 GETA$:IFA$=""THEN210                                         HO
 ·211 IFA$=CHR$(20)THEN270                                         GC
 ·212 IFA$=CHR$(133)THEN4000                                       MD
 ·213 IFA$=CHR$(134)THEN4100                                       KF
 ·214 IFA$=CHR$(135)THENPRINT" ":GOTO4500                          GE
 ·215 IFA$=CHR$(136)THENPRINT" ":GOTO4700                          BJ
 ·220 IFA$>"@"ANDA$<"G"THENT(B)=ASC(A$)-55:GOTO
   250                                                             GM
 ·230 IFA$>"/"ANDA$<":"THENT(B)=ASC(A$)-48:GOTO
   250                                                             LE
 ·240 GOSUB1100:GOTO210                                            LL
 ·250 PRINTA$"[c P][LEFT]";                                        OA
 ·260 GOTO125                                                      CG
 ·270 IFA>0THEN280                                                 OP
 ·272 A=-1:IFB=1THEN290                                            OB
 ·274 GOTO140                                                      CJ
 ·280 IFB=0THENPRINTCHR$(20);CHR$(20);:A=A-1                       HG
 ·285 A=A-1                                                        BE
 ·290 PRINTCHR$(20);:GOTO140                                       KH
 ·300 REM LAST LINE                                                AD
 ·310 PRINT" ":T=AD-(INT(AD/256)*256)                              GJ
 ·320 FORB=0TOA-1:T=T+A%(B):IFT>255THENT=T-255                     PL
 ·330 NEXT                                                         IA
 ·340 IFA%(A)<>TTHENGOSUB1010:GOTO110                              KF
 ·350 FORB=0TOA-1:POKEAD+B,A%(B):NEXT                              HN
 ·360 PRINT:PRINT"YOU ARE FINISHED!":GOTO4000                      ON
 ·1000 REM BELL AND ERROR MESSAGES                                 FL
 ·1010 PRINT:PRINT"LINE ENTERED INCORRECTLY":PR
   INT:GOTO1100                                                    DH
 ·1020 PRINT:PRINT"INPUT A 4 DIGIT HEX VALUE!":
   GOTO1100                                                        JA
 ·1030 PRINT:PRINT"ENDING IS LESS THAN STARTING
   !":B=0:GOTO1100                                                 HD
 ·1040 PRINT:PRINT"ADDRESS NOT WITHIN SPECIFIED
    RANGE!":B=0:GOTO1100                                           AG
 ·1050 PRINT:PRINT"NOT ZERO PAGE OR ROM!":B=0:G
   OTO1100                                                         KN
 ·1060 PRINT"?ERROR IN SAVE":GOTO1100                              EI
 ·1070 PRINT"?ERROR IN LOAD":GOTO1100                              GL
 ·1080 PRINT:PRINT:PRINT"END OF ML AREA":PRINT                     PG
 ·1100 POKE54276,17:POKE54276,16:RETURN                            BH
 ·1200 OPEN15,8,15:INPUT#15,A,A$:CLOSE15:PRINTA
   $:RETURN                                                        IM
 ·2000 REM GET FOUR DIGIT HEX                                      PC
 ·2010 PRINT:PRINTB$;:INPUTT$                                      GM
 ·2020 IFLEN(T$)<>4THENGOSUB1020:GOTO2010                          II
 ·2040 FORA=1TO4:A$=MID$(T$,A,1):GOSUB2060:IFT(
   A)=16THENGOSUB1020:GOTO2010                                     AD
 ·2050 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)*16)+
   T(4):RETURN                                                     GF
 ·2060 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-55:RET
   URN                                                             EH
 ·2070 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-48:RET
   URN                                                             KP
 ·2080 T(A)=16:RETURN                                              NP
 ·2500 REM ADRESS CHECK                                            LI
 ·2510 IFAD>ENTHEN1030                                             MI
 ·2515 IFB<SRORB>ENTHEN1040                                        MG
 ·2520 IFB<256OR(B>40960ANDB<49152)ORB>53247THE
   N1050                                                           MI
 ·2530 RETURN                                                      IM
 ·3000 REM ADDRESS TO HEX                                          EB
 ·3010 AC=AD:A=4096:GOSUB3070                                      HG
 ·3020 A=256:GOSUB3070                                             CE
 ·3030 A=16:GOSUB3070                                              PN
 ·3040 A=1:GOSUB3070                                               MJ
 ·3060 RETURN                                                      IM
 ·3070 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):GOT03
   090                                                             CJ
 ·3080 A$=CHR$(T+48)                                               JP
 ·3090 PRINTA$;:AC=AC-A*T:RETURN                                   AC
 ·4000 A$="**SAVE**":GOSUB4200                                     AI
 ·4050 OPEN1,T,1,A$:SYS680:CLOSE1                                  LH
 ·4060 IFST=0THENEND                                               EO
 ·4070 GOSUB1060:IFT=8THENGOSUB1200                                FJ
 ·4080 GOTO4000                                                    FF
 ·4100 A$="**LOAD**":GOSUB4200                                     AB
 ·4150 OPEN1,T,0,A$:SYS690:CLOSE1                                  MF
 ·4160 IFST=64THEN110                                              JH
 ·4170 GOSUB1070:IFT=8THENGOSUB1200                                CM
 ·4180 GOTO4100                                                    FO
 ·4200 PRINT" ":PRINTTAB(14)A$                                     FG
 ·4210 PRINT:A$="":INPUT"FILENAME";A$                              OM
 ·4215 IFA$=""THEN4210                                             GF
 ·4220 PRINT:PRINT"TAPE OR DISK?":PRINT                            DF
 ·4230 GETB$:T=1:IFB$="D"THENT=8:A$="@0:"+A$:RE
   TURN                                                            IG
 ·4240 IFB$<>"T"THEN4230                                           FN
 ·4250 RETURN                                                      IM
 ·4500 B$="CONTINUE FROM ADDRESS":GOSUB2010:AD=
   B                                                               DK
 ·4510 GOSUB2515:IFB=0THEN4500                                     MA
 ·4520 PRINT:GOTO110                                               OI
 ·4700 B$="BEGIN SCAN AT ADDRESS":GOSUB2010:AD=
   B                                                               FH
 ·4705 GOSUB2515:IFB=0THEN4700                                     NK
 ·4706 PRINT:GOTO4740                                              DI
 ·4710 FORB=0TO7:AC=PEEK(AD+B):GOSUB3030:IFAD+B
   =ENTHENAD=SR:GOSUB1080:GOTO110                                  BK
 ·4715 PRINT" ";:NEXTB                                             EC
 ·4720 PRINT:AD=AD+8                                               GN
 ·4730 GETB$:IFB$=CHR$(136)THEN110                                 MN
 ·4740 GOSUB3010:PRINT": ";:GOTO4710                               JD
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 105 and 106 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

HIDDEN CAVERN
FROM PAGE 57

```
·10 REM    HIDDEN CAVERN                                    OK
·12 REM    BY JAMES C.HILTY                                 LI
·14 POKE52,48:POKE56,48                                     IC
·16 PRINT"[CLEAR]":POKE53281,15:POKE53280
,0                                                          BM
·20 REM    TITLE SCREEN                                     NG
·22 PRINT:PRINTTAB(4)"[BLACK][RVSON] [RVS
OFF] [RVSON] [RVSOFF]  [RVSON] [RVSOFF]
 [RVSON]  [c *][RVSOFF]  [RVSON]  [c *][
RVSOFF]  [RVSON][3" "][RVSOFF]  [RVSON][
c *][RVSOFF] [RVSON] "                                      HK
·24 PRINTTAB(4)"[BLACK][RVSON] [RED] [BLA
CK] [RED] [RVSOFF] [RVSON][BLACK] [RED]
[RVSOFF] [RVSON][BLACK] [RED]  [c *][RVS
OFF] [RVSON][BLACK] [RED]  [c *][RVSOFF]
 [RVSON][BLACK] [RED][3" "][RVSOFF] [RVS
ON][BLACK] [RED][c *][BLACK] [RED] "                        GN
·26 PRINTTAB(4)"[RVSON][BLACK] [RED] [BLA
CK] [RED] [RVSOFF] [RVSON][BLACK] [RED]
[RVSOFF] [RVSON][BLACK] [RED][3" "][RVSO
FF] [RVSON][BLACK] [RED][3" "][RVSOFF] [
RVSON][BLACK] [RED] [RVSOFF][3" "][RVSON
][BLACK] [RED] [BLACK] [RED] "                              DP
·28 PRINTTAB(4)"[RVSON][BLACK] [RED] [BLA
CK] [RED] [RVSOFF] [RVSON][BLACK] [RED]
[RVSOFF] [RVSON][BLACK] [RED] [BLACK] [R
ED] [RVSOFF] [RVSON][BLACK] [RED] [BLACK
] [RED] [RVSOFF] [RVSON][BLACK] [RED] [B
LACK] [RVSOFF]  [RVSON][BLACK] [RED][3"
"]"                                                         KK
·30 PRINTTAB(4)"[RVSON][BLACK] [RED][3" "
][RVSOFF] [RVSON][BLACK] [RED] [RVSOFF]
[RVSON][BLACK] [RED] [BLACK] [RED] [RVSO
FF] [RVSON][BLACK] [RED] [BLACK] [RED] [
RVSOFF] [RVSON][BLACK] [RED]  [RVSOFF]
[RVSON][BLACK] [RED][3" "]"                                 FL
·32 PRINTTAB(4)"[RVSON][BLACK] [RED] [BLA
CK] [RED] [RVSOFF] [RVSON][BLACK] [RED]
[RVSOFF] [RVSON][BLACK] [RED] [BLACK] [R
ED] [RVSOFF] [RVSON][BLACK] [RED] [BLACK
] [RED] [RVSOFF] [RVSON][BLACK] [RED] [R
VSOFF][3" "][RVSON][BLACK] [RED][3" "]"                     PB
·34 PRINTTAB(4)"[RVSON][BLACK] [RED] [BLA
CK] [RED] [RVSOFF] [RVSON][BLACK] [RED]
[RVSOFF] [RVSON][BLACK] [RED][3" "][RVSO
FF] [RVSON][BLACK] [RED][3" "][RVSOFF] [
RVSON][BLACK] [RED] [BLACK] [RVSOFF]  [R
VSON][BLACK] [RED] [RVSOFF][c *][RVSON]
"                                                           JM
·36 PRINTTAB(5)"[RVSON][RED] [RVSOFF] [RV
SON] [RVSOFF]  [RVSON] [RVSOFF]  [RVSON]
 [RVSOFF][sEP]  [RVSON]  [RVSOFF][sEP]
[RVSON][3" "][RVSOFF]  [RVSON] [RVSOFF]
 [c *]"                                                     DP
·38 PRINT:PRINT                                             GJ
·40 PRINTTAB(6)"[RVSON][BLUE][3" "][RVSOF
F][3" "][RVSON] [RVSOFF][3" "][RVSON] [R
VSOFF] [RVSON] [RVSOFF]  [RVSON][3" "][R
VSOFF]  [RVSON][3" "][RVSOFF]  [RVSON][c
 *][RVSOFF] [RVSON] "                                       JN
·42 PRINTTAB(6)"[RVSON][BLUE] [BLACK][3"
"][RVSOFF] [RVSON][BLUE][sEP] [c *][RVSO
FF]  [RVSON] [BLACK] [BLUE] [BLACK] [RVS
OFF] [RVSON][BLUE] [BLACK][3" "][RVSOFF]
 [RVSON][BLUE] [BLACK] [BLUE] [BLACK] [R
VSOFF] [RVSON][BLUE] [c *] [BLACK] "                        DP
·44 PRINTTAB(6)"[RVSON][BLUE] [BLACK] [RV
SOFF][3" "][RVSON][BLUE] [BLACK][sEP][BL
UE] [BLACK][c *][RVSOFF] [RVSON][BLUE] [
BLACK] [BLUE] [BLACK] [RVSOFF] [RVSON][B
LUE] [BLACK] [RVSOFF][3" "][RVSON][BLUE]
[3" "][BLACK] [RVSOFF] [RVSON][BLUE][3"
"][BLACK] "                                                 MA
·46 PRINTTAB(6)"[RVSON][BLUE] [BLACK] [RV
SOFF][3" "][RVSON][BLUE][3" "][BLACK] [R
VSOFF] [RVSON][BLUE] [BLACK] [BLUE] [BLA
CK] [RVSOFF] [RVSON][BLUE]  [RVSOFF][3"
"][RVSON]  [BLACK]  [RVSOFF] [RVSON][BLU
E][3" "][BLACK] "                                           CO
·48 PRINTTAB(6)"[RVSON][BLUE] [BLACK] [RV
SOFF][3" "][RVSON][BLUE] [BLACK] [BLUE]
[BLACK] [RVSOFF] [RVSON][BLUE] [BLACK] [
BLUE] [BLACK] [RVSOFF] [RVSON][BLUE] [BL
ACK]  [RVSOFF]  [RVSON][BLUE][3" "][RVSO
FF]  [RVSON][3" "][BLACK] "                                 BL
·52 PRINTTAB(6)"[RVSON][BLUE][3" "][RVSOF
F]  [RVSON] [BLACK] [BLUE] [BLACK] [RVSO
FF]  [RVSON][BLUE] [BLACK] [RVSOFF][sEP]
 [RVSON][BLUE][3" "][RVSOFF]  [RVSON] [B
LACK] [BLUE] [BLACK] [RVSOFF] [RVSON][BL
UE] [BLACK] [RVSOFF][BLUE][c *][RVSON][B
LACK] "                                                     KB
·54 PRINTTAB(7)"[RVSON][BLACK][3" "][RVSO
FF]  [RVSON] [RVSOFF] [RVSON] [RVSOFF][3
" "][RVSON] [RVSOFF][3" "][RVSON][3" "][
RVSOFF]  [RVSON] [RVSOFF] [RVSON] [RVSOF
F]  [RVSON] [RVSOFF] [c *]"                                 OJ
·56 PRINT:PRINTTAB(9)"[BLUE]LAND OF THE C
AVE GLOBS"                                                  JN
·60 FORL=54272TO54295:POKEL,0:NEXT:POKE54
296,15:POKE54277,8:POKE54278,255                            KC
·62 POKE54276,21:F1=2:FORZ=1TO28:F2=80:PO
KE54273,F1:FORY=1TO5:POKE54287,F2                           LC
·64 F2=F2/0.9:NEXTY:F1=F1+9:NEXTZ:POKE542
78,15                                                       PL
·66 PRINTTAB(6)"PLEASE WAIT[4"."]READING
DATA"                                                       EE
```

```
·68 REM    READ SPRITE SHAPE DATA                FG
·70 FORX=12288TO12350:READA:POKEX,A:NEXT         NM
·72 FORX=12352TO12414:READA:POKEX,A:NEXT         LH
·74 FORX=12416TO12478:READA:POKEX,A:NEXT         LO
·76 FORX=12480TO12542:READA:POKEX,A:NEXT         KD
·78 FORX=12544TO12606:READA:POKEX,A:NEXT         LH
·80 FORX=12608TO12670:READA:POKEX,A:NEXT         LH
·82 FORX=12672TO12734:READA:POKEX,A:NEXT         KL
·84 FORX=12736TO12798:READA:POKEX,A:NEXT         LO
·86 FORX=12800TO12862:READA:POKEX,A:NEXT         IK
·88 REM    READ SCROLL DATA                      KL
·90 FORX=49152TO49528:READA:POKEX,A:NEXT         LP
·92 REM    READ SPRITE MOTION ROUTINE            EL
·94 FORX=50880TO51116:READA:POKEX,A:NEXT         ND
·96 V=53248                                      AD
·97 HS=0                                         IC
·98 REM    SET UP PLAYING SCREEN                 CB
·99 SC=0:M=0:SH=3                                JF
·100 POKE53280,0:POKE53281,0:PRINT"[CLEAR
 ][CYAN][7"[c O]"] [7"[c O]"]  [8"[c O]"]
 [6" "][8"[c O]"]"                               JP
·102 PRINT"[c J]SCORE[c L] [c J]SHIPS[c L
 ]  [c J]HIDDEN[c L][6" "][c J]MISSES[c L
 ]"                                              IP
·104 PRINT"[7"[c U]"] [7"[c U]"]  [c J]CA
 VERN[c L][6" "][8"[c U]"]"                      BK
·106 PRINTTAB(17)"[8"[c U]"]"                    BA
·108 PRINT"[GREEN][RVSON][39" "]"                GP
·110 PRINTTAB(2)"[RVSON][3" "][RVSOFF][5"
  "][RVSON][4" "][RVSOFF][4" "][RVSON][4"
  "][RVSOFF][3" "][RVSON]  [RVSOFF][3" "]
 [RVSON]  [RVSOFF][4" "][RVSON]  "               FL
·112 PRINTTAB(2)"[RVSON][3" "][RVSOFF][5"
  "][RVSON]  [RVSOFF][7" "][RVSON]  [RVSO
 FF][4" "][RVSON] [RVSOFF][4" "][RVSON] [
 RVSOFF][5" "][RVSON]  "                         JL
·114 PRINTTAB(3)"[RVSON] [RVSOFF][6" "][R
 VSON] [RVSOFF][8" "][RVSON]  [RVSOFF][4"
  "][RVSON] [RVSOFF][4" "][RVSON] [RVSOFF
 ][5" "][RVSON]  "                               JC
·116 PRINTTAB(3)"[RVSON] [RVSOFF][6" "][R
 VSON] [RVSOFF][RVSOFF][8" "][RVSON]  [RV
 SOFF][4" "][c K][4" "][c K][5" "][RVSON]
  "                                              OF
·118 PRINTTAB(3)"[RVSON] [RVSOFF][6" "][c
  K][8" "][RVSON]  [RVSOFF][9" "][c K][6"
  "][RVSON] "                                    OO
·120 PRINTTAB(3)"[c K][6" "][c K][8" "][R
 VSON] [RVSOFF][10" "][c H][6" "][RVSON]
 "                                               FJ
·122 PRINTTAB(3)"[c K][15" "][RVSON] [RVS
 OFF][17" "][c L]"                               FA
·124 PRINTTAB(3)"[c K][15" "][RVSON] [RVS
 OFF][17" "][c L]"                               FA
·126 PRINTTAB(3)"[c K][15" "][RVSON] [RVS
 OFF][17" "][c L]"                               FA
·128 PRINTTAB(19)"[c H]"                         LK
·130 PRINT:PRINTTAB(11)"[c J][20" "][c L]
 "                                               JK
·132 PRINTTAB(11)"[c J][20" "][c L]"             OO
·134 PRINTTAB(2)"[c K][8" "][RVSON] [RVSO
 FF][12" "][c K][7" "][c L] "                    LE
·136 PRINTTAB(2)"[c K][8" "][RVSON] [RVSO
 FF][12" "][RVSON] [RVSOFF][6" "][RVSON]
  "                                              ED
·138 PRINTTAB(2)"[RVSON] [RVSOFF][8" "][R
 VSON]  [RVSOFF][11" "][RVSON] [RVSOFF][6
 " "][RVSON]  "                                  NP
·140 PRINTTAB(2)"[RVSON] [RVSOFF][8" "][R
 VSON]  [RVSOFF][11" "][RVSON] [RVSOFF][5
 " "][RVSON][3" "]"                              BP
·142 PRINTTAB(2)"[RVSON] [RVSOFF][7" "][R
 VSON][4" "][RVSOFF][9" "][RVSON][3" "][R
 VSOFF][4" "][RVSON][3" "]"                      PD
·144 PRINT"[RVSON][39" "]"                       GO
·146 POKE49522,2:POKE49523,0:POKE49524,38
 :POKE49525,5:POKE49526,22                       DD
·148 POKEV+39,3:POKEV+40,10:POKEV+41,13:P
 OKEV+42,13:POKEV+43,13:POKEV+44,13              PB
·150 POKEV+45,7:POKEV+46,1                       DG
·152 POKE2040,192:POKE2041,192:POKE2042,1
 96:POKE2043,196    -                            CD
·154 POKE2044,197:POKE2045,197:POKE2046,1
 98:POKE2047,199                                 MK
·155 POKEV+29,60:POKEV+23,60                     GK
·156 POKEV+0,40:POKEV+1,154:POKEV+2,50:PO
 KEV+3,220                                       LJ
·158 POKEV+4,40:POKEV+5,82:POKEV+6,200:PO
 KEV+7,82                                        ON
·160 POKEV+8,46:POKEV+9,196:POKEV+10,230:
 POKEV+11,196                                    MI
·162 POKEV+14,246:POKEV+15,161                   PG
·164 POKEV+27,63:POKEV+21,191                    MO
·166 POKE50433,2:POKE50434,254:POKE50435,
 1:POKE50436,3                                   MH
·168 POKE50437,4:POKE50438,0:POKE50439,4:
 POKE50440,0                                     DI
·170 POKE50441,4:POKE50442,0:POKE50443,4:
 POKE50444,0                                     JF
·172 POKE50445,0:POKE50446,0:POKE50447,0:
 POKE50448,0                                     CF
·175 PRINT"[HOME][3"[DOWN]"][WHITE] "SC"[
 5" "]"SH:PRINT"[HOME][3"[DOWN]"]"TAB(33)
 M                                               MI
·176 FORL=54272TO54295:POKEL,0:NEXT              KD
·177 POKE54287,15:POKE54284,160:POKE54285
 ,252                                            CL
·178 POKE54280,80:POKE54279,40:POKE54283,
 129                                             DF
·180 W1=PEEK(V+30):W2=PEEK(V+31)                 OE
·190 POKE50432,255:SYS51104                      HG
·199 REM    MAIN LOOP                            PK
·200 JY=PEEK(56321)AND15                         HM
·204 IFJY=13THENPOKE50448,1:POKE50688,128        NJ
·208 IFJY=14THENPOKE50448,255:POKE50688,1
 28                                              FG
```

```
·210 FB=-((PEEK(56321)AND16)=0):IFFB=1THEN300                    KE
·211 SYS49152                                                    KF
·212 W1=PEEK(V+30):IFW1>128THEN350                               JO
·214 W2=PEEK(V+31):IFW2>127THEN350                               OC
·215 IF SC>2500THENSYS49152                                      PG
·216 SC=SC+1:PRINT"[HOME][3"[DOWN]"] "SC                         PL
·218 POKE50448,0:POKE50688,128                                   EE
·219 IFSC>500THENSYS49152                                        KL
·220 GOTO200                                                     BO
·300 REM    FIRE                                                 MH
·304 X=PEEK(V+14)-20:Y=PEEK(V+15)                                LB
·306 POKEV+12,X:POKEV+13,Y:POKEV+21,255                          EI
·307 POKE50445,255:POKE50688,64:POKE50432,64                     EA
·310 C=PEEK(V+12):IFC<32THEN320                                  OE
·311 W1=PEEK(V+30):IFPEEK(V+30)>64THEN330                        AP
·312 GOTO310                                                     BM
·320 POKE50445,0:POKE50688,64:POKE50432,255:POKEV+21,191         IM
·322 M=M+1:PRINT"[HOME][3"[DOWN]"]"TAB(33)M:IFM>2THEN500         LP
·323 GOTO200                                                     BO
·330 REM    HIT CAVE GLOB                                        MI
·331 Q=192                                                       JN
·332 POKEV+21,191:POKE50445,0:POKE50688,64                       OH
·334 FORX=54272TO54295:POKEX,0:NEXT:POKE54296,15                 MH
·336 POKE54277,8:POKE54278,255:POKE54276,23:F1=202               JB
·338 FORZ=1TO16:F2=30:Q=Q+1:IFQ=196THENQ=192                      BE
·339 POKE2040,Q:POKE2041,Q                                       BO
·340 POKE54273,F1:FORY=1TO10:POKE54287,F2:F2=F2*1.01:NEXTY:F1=F1-8:NEXTZ  CL
·344 SC=SC+250:PRINT"[HOME][3"[DOWN]"][WHITE] "SC                GD
·346 FORX=54272TO54295:POKEX,0:NEXT                              JL
·347 POKE54287,15:POKE54284,160:POKE54285,252                    CL
·348 POKE54280,80:POKE54279,40:POKE54283,129                     DF
·349 POKEV+1,52:POKEV+3,220:W1=PEEK(V+30):POKE50432,255:GOTO200  FM
·350 REM    CAVE CRUISER HIT                                     OP
·351 POKE50432,0                                                 AG
·352 FORX=54272TO54295:POKEX,0:NEXT:POKE54296,15:POKE54277,8:POKE54278,255  LK
·354 POKE54276,85:F1=100:F2=230                                  EO
·356 FORZ=1TO77:POKE54272,F1:POKE54287,F2                        MG
·358 F2=F2-2:F1=F1*.99:NEXT                                      GE
·360 FORX=54272TO54295:POKEX,0:NEXT                              JL
·362 POKE54287,15:POKE54284,160:POKE54285,252                    CL
·363 POKE2047,200                                                CI
·364 POKE54280,80:POKE54279,40:POKE54283,129:FORT=0TO1000:NEXT   GD
·366 FORX=54272TO54295:POKEX,0:NEXT                              JL
·367 SH=SH-1:IFSH=0THEN500                                       DO
·368 POKEV+21,63:FORT=0TO500:NEXT:POKEV+21,0:POKEV+16,0:GOTO100  GC
·500 REM    GAME OVER-REPLAY                                     OM
·501 FORL=54272TO54295:POKEL,0:NEXT                              KD
·502 PRINT"[CLEAR][CYAN]":POKEV+21,0                             KN
·503 IFSC>HS THEN HS=SC                                          AE
·504 IFM=3THENPRINT"[4"[DOWN]"]"TAB(8)"GAME OVER[4"-"]YOU MISSED 3 SHOTS":GOTO510  FA
·505 PRINT"[4"[DOWN]"]"TAB(8)"GAME OVER[4"-"]YOU LOST 3 SHIPS"   OI
·510 PRINT"[6"[DOWN]"]"TAB(12)"S C O R E ";SC                    KH
·512 PRINT"[DOWN]"TAB(12)"HIGH SCORE  ";HS                       HL
```

MAKING HEADLINES FROM PAGE 92

Use of an assembler required for entry!

BIGCHRS

```
                *
                * BIGCHR
                *
                        ORG     $8000
                *
COLOR           EQU     $10
COLMAP          EQU     $8400
BASE            EQU     $A000
VICTRL          EQU     $D011
CI2PRA          EQU     $DD00
CIADIR          EQU     $DD02
VICMEM          EQU     $D018
                *
HMAX            EQU     320
HMID            EQU     160-4
VMID            EQU     100-4
                *
SCRLEN          EQU     8000
MAPLEN          EQU     1000
                *
TEMPA           EQU     $FB
TEMPB           EQU     TEMPA+2
                *
TABPTR          EQU     TEMPA
TABSIZ          EQU     $02A7
                *
HPSN            EQU     TABSIZ+2
VPSN            EQU     HPSN+2
CHAR            EQU     VPSN+1
ROW             EQU     CHAR+1
LINE            EQU     ROW+1
BYTE            EQU     LINE+1
BITT            EQU     BYTE+2
                *
MPRL            EQU     BITT+1
MPRH            EQU     MPRL+1
MPDL            EQU     MPRH+1
MPDH            EQU     MPDL+1
PRODL           EQU     MPDH+1
PRODH           EQU     PRODL+1
                *
FILVAL          EQU     PRODH+1
                *
R6510           EQU     $0001
NEWADR          EQU     $8800
CHRBAS          EQU     $D000
CIACRE          EQU     $DC0E
                *
TABLEN          EQU     $800
                *
MVSRCE          EQU     $61
MVDEST          EQU     MVSRCE+2
BYTPTR          EQU     MVDEST+2
                *
LENPTR          EQU     $9000
CHCODE          EQU     LENPTR+2
HPTR            EQU     CHCODE+2
```

```
58 ONEBYT   EQU   HPTR+1
59 COUNT    EQU   ONEBYT+2
60 *
61          JMP   START
62 *
63 * BLOCK FILL ROUTINE
64 *
65 BLKFIL   LDA   FILVAL
66          LDX   TABSIZ+1
67          BEQ   PARTPG
68          LDY   #0
69 FULLPG   STA   (TABPTR),Y
70          INY
71          BNE   FULLPG
72          INC   TABPTR+1
73          DEX
74          BNE   FULLPG
75 PARTPG   LDX   TABSIZ
76          BEQ   FINI
77          LDY   #0
78 PARTLP   STA   (TABPTR),Y
79          INY
80          DEX
81          BNE   PARTLP
82 FINI     RTS
83 *
84 * 16-BIT MULTIPLICATION
85 * ROUTINE
86 *
87 MULT16   LDA   #0
88          STA   PRODL
89          STA   PRODH
90          LDX   #17
91          CLC
92 MULT     ROR   PRODH
93          ROR   PRODL
94          ROR   MPRH
95          ROR   MPRL
96          BCC   CTDOWN
97          CLC
98          LDA   MPDL
99          ADC   PRODL
100         STA   PRODL
101         LDA   MPDH
102         ADC   PRODH
103         STA   PRODH
104 CTDOWN  DEX
105         BNE   MULT
106         RTS
107 *
108 * PLOT ROUTINE
109 *
110 * ROW=VPSN/8
111 * (8-BIT DIVIDE)
112 *
113 PLOT     LDA   VPSN
114         LSR   A
115         LSR   A
116         LSR   A
117         STA   ROW
118 *
119 * CHAR=HPSN/8
120 * (16-BIT DIVIDE)
121 *
122         LDA   HPSN
123         STA   TEMPA
124         LDA   HPSN+1
125         STA   TEMPA+1
126         LDX   #3
127 DLOOP   LSR   TEMPA+1
128         ROR   TEMPA
129         DEX
130         BNE   DLOOP
131         LDA   TEMPA
132         STA   CHAR
133 *
134 * LINE=VPSN AND 7
135 *
136         LDA   VPSN
137         AND   #7
138         STA   LINE
139 *
140 * BITT=7-(HPSN AND 7)
141 *
142         LDA   HPSN
143         AND   #7
144         STA   BITT
145         SEC
146         LDA   #7
147         SBC   BITT
148         STA   BITT
149 *
150 * BYTE=BASE+ROW*HMAX
151 * +8*CHAR+LINE
152 *
153 * FIRST MULTIPLY
154 * ROW * HMAX
155 *
156         LDA   ROW
157         STA   MPRL
158         LDA   #0
159         STA   MPRH
160         LDA   #<HMAX
161         STA   MPDL
162         LDA   #>HMAX
163         STA   MPDH
164         JSR   MULT16
165         LDA   MPRL
166         STA   TEMPA
167         LDA   MPRL+1
168         STA   TEMPA+1
169 *
170 * ADD PRODUCT TO BASE
171 *
172         CLC
173         LDA   #<BASE
174         ADC   TEMPA
175         STA   TEMPA
176         LDA   #>BASE
177         ADC   TEMPA+1
178         STA   TEMPA+1
179 *
180 * MULTIPLY 8 * CHAR
181 *
182         LDA   #8
183         STA   MPRL
184         LDA   #0
185         STA   MPRH
186         LDA   CHAR
187         STA   MPDL
188         LDA   #0
189         STA   MPDH
190         JSR   MULT16
191         LDA   MPRL
192         STA   TEMPB
193         LDA   MPRH
194         STA   TEMPB+1
195 *
196 * ADD LINE
197 *
198         CLC
199         LDA   TEMPB
200         ADC   LINE
201         STA   TEMPB
202         LDA   TEMPB+1
203         ADC   #0
204         STA   TEMPB+1
205 *
206 * TEMPA + TEMPB = BYTE
207 *
208         CLC
209         LDA   TEMPA
210         ADC   TEMPB
211         STA   TEMPB
212         LDA   TEMPA+1
213         ADC   TEMPB+1
214         STA   TEMPB+1
215 *
216 * POKE BYTE,PEEK(BYTE)
217 * OR2^BIT
218 *
219         LDX   BITT
220         INX
221         LDA   #0
222         SEC
223 SQUARE   ROL
224         DEX
225         BNE   SQUARE
226         LDY   #0
227         ORA   (TEMPB),Y
228         STA   (TEMPB),Y
229         RTS
230 *
231 * CALCULATE CHCODE'S
```

```
232 * ADDRESS
233 *
234 GETADR   LDA   #0
235           STA   CHCODE+1
236           LDA   CHCODE
237           CLC
238           ASL   A
239           ROL   CHCODE+1
240           ASL   A
241           ROL   CHCODE+1
242           ASL   A
243           ROL   CHCODE+1
244           STA   CHCODE
245 *
246           CLC
247           LDA   CHCODE
248           ADC   #<NEWADR
249           STA   BYTPTR
250           LDA   CHCODE+1
251           ADC   #>NEWADR
252           STA   BYTPTR+1
253           RTS
254 *
255 * MAIN ROUTINE STARTS
256 * HERE
257 *
258 START     LDA   VICMEM
259           ORA   #8
260           STA   VICMEM
261 *
262           LDA   VICTRL
263           ORA   #32
264           STA   VICTRL
265 *
266 * USE BANK 2
267 *
268           LDA   CIADIR
269           ORA   #3
270           STA   CIADIR
271 *
272           LDA   CI2PRA
273           AND   #252
274           ORA   #1
275           STA   CI2PRA
276 *
277 * CLEAR BIT MAP
278 *
279           LDA   #0
280           STA   FILVAL
281           LDA   #<BASE
282           STA   TABPTR
283           LDA   #>BASE
284           STA   TABPTR+1
285           LDA   #<SCRLEN
286           STA   TABSIZ
287           LDA   #>SCRLEN
288           STA   TABSIZ+1
289           JSR   BLKFIL
290 *
291 * SET BKG AND LINE
292 * COLORS
293 *
294           LDA   #COLOR
295           STA   FILVAL
296           LDA   #<COLMAP
297           STA   TABPTR
298           LDA   #>COLMAP
299           STA   TABPTR+1
300           LDA   #<MAPLEN
301           STA   TABSIZ
302           LDA   #>MAPLEN
303           STA   TABSIZ+1
304           JSR   BLKFIL
305 *
306 * TURN OFF KB INTERRUPT
307 * TIMER
308 *
309 MVCHRS    LDA   CIACRE
310           AND   #$FE
311           STA   CIACRE
312 *
313 * SWITCH BASIC OUT
314 *
315           LDA   R6510
316           AND   #$FE
317           STA   R6510
318 *
319 * SWITCH I/O OFF,
320 * CHAR ROM ON
321 *
322           LDA   R6510
323           AND   #$FB
324           STA   R6510
325 *
326 * COPY CHARACTERS
327 * INTO RAM
328 *
329           LDA   #<CHRBAS
330           STA   MVSRCE
331           LDA   #>CHRBAS
332           STA   MVSRCE+1
333 *
334           LDA   #<NEWADR
335           STA   MVDEST
336           LDA   #>NEWADR
337           STA   MVDEST+1
338 *
339           LDA   #<TABLEN
340           STA   LENPTR
341           LDA   #>TABLEN
342           STA   LENPTR+1
343 *
344 * START MOVE
345 *
346           LDY   #0
347           LDX   LENPTR+1
348           BEQ   MVPART
349 MVPAGE    LDA   (MVSRCE),Y
350           STA   (MVDEST),Y
351           INY
352           BNE   MVPAGE
353           INC   MVSRCE+1
354           INC   MVDEST+1
355           DEX
356           BNE   MVPAGE
357 MVPART    LDX   LENPTR
358           BEQ   MVEXIT
359 MVLAST    LDA   (MVSRCE),Y
360           STA   (MVDEST),Y
361           INY
362           DEX
363           BNE   MVLAST
364 MVEXIT
365 *
366 * SWITCH I/O BACK IN
367 *
368           LDA   R6510
369           ORA   #4
370           STA   R6510
371 *
372 * TURN TIMER BACK ON
373 *
374           LDA   CIACRE
375           ORA   #1
376           STA   CIACRE
377 *
378 * DRAW A CHARACTER
379 *
380           LDA   #<HMID
381           STA   HPSN
382           STA   HPTR
383           LDA   #>HMID
384           STA   HPSN+1
385           STA   HPTR+1
386           LDA   #VMID
387           STA   VPSN
388 *
389           LDA   #1
390           STA   CHCODE
391           JSR   GETADR
392 *
393 * A NESTED LOOP:
394 *
395 * X IS THE OUTSIDE
396 * LOOP
397 *
398           LDX   #8
399 *
400 * SET UP COUNTER FOR
401 * 2 VERT LINES
402 *
403 SETLIN   LDA   #2
404           STA   COUNT
405 *
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 105 and 106 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
406 DRAWLN    LDY   #0
407           LDA   (BYTPTR),Y
408           STA   ONEBYT
409 *
410 * THE INSIDE LOOP:
411 *
412 * (Y IS ZERO AT START)
413 *
414 RSHIFT    LDA   ONEBYT
415           ASL   A
416           STA   ONEBYT
417           BCS   SHOW
418 *
419           INC   HPSN
420           BNE   ITSOK
421           INC   HPSN+1
422 ITSOK     JMP   NOSHOW
423 *
424 * DISPLAY BIT
425 *
426 * SAVE X AND Y
427 * REGISTERS
428 *
429 SHOW      TXA
430           PHA
431           TYA
432           PHA
433 *
434           JSR   PLOT
435 *
436 * NOW DO IT AGAIN
437 *
438           INC   HPSN
439           BNE   NOINC
440           INC   HPSN+1
441 *
442 NOINC     JSR   PLOT
443 *
444 * RETRIEVE X AND Y
445 * REGISTERS
446 *
447           PLA
448           TAY
449           PLA
450           TAX
451 *
452 NOSHOW    INC   HPSN
453           BNE   LEAP
454           INC   HPSN+1
455 *
456 LEAP      INY
457           CPY   #8
458           BCC   RSHIFT
459 *
460           INC   VPSN
461 *
462           LDA   HPTR
463           STA   HPSN
464           LDA   HPTR+1
465           STA   HPSN+1
466 *
467 * 2 VERT LINES
468 * DONE YET?
469           DEC   COUNT
470           BNE   DRAWLN
471 *
472           INC   BYTPTR
473           BNE   OKMSB
474           INC   BYTPTR+1
475 OKMSB     DEX
476           BNE   SETLIN
477 *
478 INF       JMP   INF
479 *
480           END
481 *
```

1040A CALCULATOR
FROM PAGE 97

```
·10 PRINT"FORM 1040A CALCULATOR"                                EO
·20 PRINT"COPYRIGHT 1986 CHERYL PETERSON"                       IA
·30 PRINT"PRINT AN INSTRUCTION LIST FOR F
 ILLING OUT 1040A WHEN FINISHED?"                               ID
·40 INPUT"[RVSON]Y[RVSOFF]ES OR [RVSON]N[
 RVSOFF]O";Q$                                                   DD
·45 IF Q$<>"Y"THEN90                                            DO
·50 INPUT"TO [RVSON]S[RVSOFF]CREEN OR [RV
 SON]P[RVSOFF]RINTER";Q2$                                       DL
·90 REM INPUT STATEMENTS                                        BM
·100 INPUT"HOW MANY EXEMPTIONS ARE YOU CL
 AIMING";X                                                      LF
·200 INPUT"INCOME FROM W-2 FORMS";W                             EO
·300 INPUT"INTEREST COLLECTED THIS YEAR";
 I                                                              EC
·400 INPUT"DIVIDENDS COLLECTED THIS YEAR"
 ;D                                                             LL
·500 INPUT"EXCLUSIONS";E                                        HP
·600 INPUT"TAXABLE UNEMPLOYMENT COMPENSAT
 ION";U                                                         CN
·700 INPUT"CONTRIBUTIONS TO POLITICIANS T
 HIS YEAR";P                                                    FD
·800 INPUT"FEDERAL INCOME TAX WITHHELD";F                       NG
·900 INPUT"EARNED INCOME CREDIT";C                              OM
·1100 GOSUB 2100                                                FE
·1140 PRINT"[CLEAR]"                                            HH
·1150 PRINT"ENTER ";GI;"ON LINE 10 OF THE
  TAX FORM."                                                    KJ
·1160 PRINT"ENTER ";X2;"ON LINE 11 OF THE
  TAX FORM."                                                    PK
·1170 PRINT"ENTER ";TN;"ON LINE 12 OF THE
  TAX FORM."                                                    CL
·1200 PRINT"LOOK UP THE AMOUNT OF TAXES D
 UE"                                                            NA
·1210 PRINT"ON $";TN;"IN THE TAX TABLES."                       IP
·1300 INPUT"TAXES DUE ACCORDING TO TABLE"
 ;TD                                                            NE
·1400 INPUT"ADVANCED EARNED INCOME CREDIT
 ";A                                                            PJ
·1500 GOSUB 2600                                                FB
·1550 PRINT"[CLEAR]"                                            HH
·1560 IF Q2$<>"P"AND Q$<>"N"THEN GOSUB 19
 30                                                             II
·1570 IF Q2$="P" THEN GOSUB 1921                                IJ
·1600 IF TP>TT THEN R=TP-TT                                     BE
·1650 IF TP=TT THEN 1800                                        AC
·1660 IF TP<TT THEN 1900                                        AA
·1700 PRINT"YOUR TOTAL REFUND IS $";R                           PC
·1710 PRINT"ENTER THIS AMOUNT ON LINE 17.
 "                                                              IE
·1797 IF CL<>1 THEN 1799                                        BJ
·1798 PRINT#4:CLOSE4                                            OA
·1799 END                                                       IC
·1800 PRINT"YOU SEEM TO HAVE COME OUT EVE
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 105 and 106 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
N."                                                    KJ
•1810 PRINT"ENTER 0 ON LINES 17 AND 18."               HF
•1897 IF CL<>1 THEN 1799                               BJ
•1898 PRINT#4:CLOSE4                                   OA
•1899 END                                              IC
•1900 R=TT-TP                                          PG
•1910 PRINT"YOU OWE THE GOVERNMENT $";R                HG
•1915 PRINT"ENTER THIS AMOUNT ON LINE 18.
"                                                      IF
•1917 IF CL<>1 THEN 1799                               BJ
•1918 PRINT#4:CLOSE4                                   OA
•1920 END                                              IC
•1921 IF Q$<>"Y"THEN RETURN                            FG
•1922 OPEN4,4:CMD4                                     EL
•1930 PRINT"FILL OUT QUESTIONS 1-5 AS APP
ROPRIATE."                                             EM
•1931 PRINT"ENTER";X;"IN BOX NUMBER 6."                MN
•1932 PRINT"ENTER";W;"IN BOX NUMBER 7."                ON
•1940 PRINT"ENTER";I;"IN BOX NUMBER 8A."               HP
•1941 PRINT"ENTER";D;"IN BOX NUMBER 8B."               EB
•1942 PRINT"ENTER";Z;"IN BOX NUMBER 8C."               LM
•1943 PRINT"ENTER";E;"IN BOX NUMBER 8D."               NK
•1944 PRINT"ENTER";Y;"IN BOX NUMBER 8E."               BD
•1955 PRINT"ENTER";GI;"IN BOX NUMBER 10."              CH
•1960 PRINT"ENTER";X2;"IN BOX NUMBER 11."              MK
•1965 PRINT"ENTER";TN;"IN BOX NUMBER 12."              IP
•1970 PRINT"ENTER";P;"IN BOX NUMBER 13A."              ON
•1971 PRINT"ENTER";F;"IN BOX NUMBER 13B."              OI
•1972 PRINT"ENTER";C;"IN BOX NUMBER 13C."              GA
•1975 PRINT"ENTER";TP;"IN BOX NUMBER 14."              BH
•1980 PRINT"ENTER";TD;"IN BOX NUMBER 15A.
"                                                      GJ
•1985 PRINT"ENTER";A;"IN BOX NUMBER 15B."              AD
•1990 PRINT"ENTER";TT;"IN BOX NUMBER 16."              ON
•1997 IF Q2$="P" THEN CL=1                             HF
•1998 RETURN                                           IM
•1999 END                                              IC
•2000 REM COMPUTATIONS                                 FP
•2100 Z=I+D:Y=Z-E:IF Y<0 THEN Y=0                      FO
•2200 GI=W+Y+U                                         JG
•2300 X2=1000*X                                        BB
•2400 TN=GI-X2                                         BG
•2500 TP=P+F+C                                         JK
•2550 RETURN                                           IM
•2600 TT=TD+A                                          PO
•2700 RETURN                                           IM
```

JCALC
FROM PAGE 28

ML SUBROUTINE BUILDER

```
•0 REM---THIS IS WHERE THE MACHINE LANGUA
GE WILL BE                                             HD
•10 SB=PEEK(43)+256*PEEK(44):L1=SB+5:L2=L
1+43                                                   HL
•20 FORI=L1TOL2:READA:POKEI,A:CK=CK+A:NEX
TI                                                     KH
•30 IFCK<>4670THENPRINT"ERROR IN DATA STA
TEMENTS -- CHECK YOUR PROGRAM":END                     IN
•40 PRINT"DATA ENTERED CORRECTLY -- DELET
E ANY LINE BUT 0 AND CONTINUE ENTERING J
CALC"                                                  OC
•50 DATA165,122,141,112,3,165,123,141,113
,3,165,10,133,122,169,2,133,123,32,121                 JF
•60 DATA165,165,10,133,122,169,2,133,123,
32,165,169,173,112,3,133,122,173,113,3                 BC
•70 DATA133,123,96,0                                   ND
```

BASIC PORTION

```
•1 PRINTCHR$(8);CHR$(14):GOTO1000                      DP
•5 POKE54276,17:FORTT=1TO50:NEXT:POKE5427
6,16:RETURN                                            BP
•10 IFCC>BCTHENBC=CC                                   LA
•12 IFCR>BRTHENBR=CR                                   PM
•14 RETURN                                             IM
•16 PRINTCH$;A$;"[c @]";:T$=A$                         DP
•18 GETB$:IFB$=""THEN18                                DA
•20 IF(B$=CHR$(20))AND(LEN(T$)>1)THENT$=L
EFT$(T$,LEN(T$)-1):PRINTB$;"[LEFT]";"[c
@]";:GOTO18                                            HM
•24 IFB$="[HOME]"THEN16                                AH
•26 IFB$=CHR$(13)THENRETURN                            EH
•28 PRINT"[LEFT]"B$"[c @]";:T$=T$+B$:ON-(
LEN(T$)>=SL)GOSUB5:GOTO18                              AD
•30 IFT$=""THENGOSUB5:RETURN                           CD
•32 C=ASC(T$)-65:R=VAL(MID$(T$,2))-1:RETU
RN                                                     ME
•34 HC=CC:HR=CR:RETURN                                 LI
•36 CC=HC:CR=HR:RETURN                                 JA
•38 PRINTCH$;:A$="RANGE?":GOSUB16:T$=MID$
(T$,7)                                                 GF
•39 IFT$=""THENR1=CR:R2=CR:C1=CC:C2=CC:RE
TURN                                                   FL
•40 GOSUB30:R1=R:C1=C                                  HK
•42 FORII=1TOLEN(T$):II$=MID$(T$,II,1):IF
II$=":"THEN46                                          EI
•43 IFII$=")"THEN48                                    HM
•44 NEXT:GOTO48                                        LH
•46 T$=MID$(T$,II+1):GOSUB30                           HC
•48 R2=R:C2=C:RETURN                                   PF
•50 PRINTCH$" ";:POKELL,2*W                            NP
•51 IFV$(CC,CR)="#"THENPRINTV(CC,CR)"[HOM
E]";:INPUTV(CC,CR):GOTO68                              HP
•52 PRINT" "Q$;V$(CC,CR);Q$"[HOME]";:INPU
TV$(CC,CR):GOSUB900:GOTO68                             KA
•54 PRINT"[CLEAR][RVSON]WAIT[RVSOFF]":GOS
UB890:GOSUB76:GOSUB68:GOTO84                           LJ
•56 IFCA>0THENFORI=CATOCA+CW-1:POKEI,PEEK
(I)AND127:NEXT                                         HO
•58 GOSUB84:GOTO68                                     CN
•62 T$=V$(CC,CR):IFT$=""THENT$=BL$:RETURN              HJ
```

```
·64 IFASC(T$)=39THENT$=LEFT$(MID$(T$,2)+BL$,CW):RETURN OE
·66 T$=RIGHT$(BL$+STR$(INT(100*V(CC,CR)+.5)/100),CW):RETURN BO
·68 GOSUB62:POKE782,3+(CC-LC)*CW:POKE781,4+CR-TR:POKE783,0:SYS65520 NI
·70 PRINT"[RVSON]"T$"[RVSOFF]"CH$;V$(CC,CR);:IFV$(CC,CR)="#"THENPRINTV(CC,CR); CO
·72 PRINT"[HOME]";:RETURN OF
·76 PRINT"[CLEAR]":PRINT:PRINT:GOSUB34 FF
·78 PRINT"[RVSON]"SPC(7);CHR$(LC+65);:FORI=LC+1TOLC+NC:PRINTSPC(CW);CHR$(I+65);:NEXT:PRINT"[RVSOFF]" JN
·80 FORCR=TRTOTR+NR:PRINT"[RVSON]"RIGHT$(BL$+STR$(CR+1),3)"[RVSOFF]"; FD
·82 FORCC=LCTOLC+NC:GOSUB62:PRINTT$;:NEXT:PRINT:NEXT:CA=0:GOSUB36:RETURN BK
·84 CA=SO+3+W*(4+CR-TR)+CW*(CC-LC):RETURN FJ
·86 PRINTCH$;:INPUT"DEVICE#";D:IFD=0THEN100 DF
·87 IFD<>1ANDD<>8THENOPENDI,D,7:RETURN DO
·88 PRINTCH$;:INPUT"FILENAME";F$:SA=-(DI=1):IFD=8THENSA=4 PG
·90 IFD=8ANDDI=1THENF$="@0:"+F$+",S,W" GG
·92 IFD=8ANDDI=2THENF$=F$+",S,R" ND
·94 OPENDI,D,SA,F$:GOTO68 HK
·96 DI=1:GOSUB86:PRINT#1,BR","BC:FORI=0TOBR HM
·97 FORJ=0TOBC:PRINT#1,V(J,I)","Q$;V$(J,I);Q$:NEXTJ,I DA
·98 CLOSE1:RETURN JG
·100 DI=2:GOSUB86:INPUT#2,BR,BC BH
·110 FORI=0TOBR:FORJ=0TOBC:INPUT#2,V(J,I),V$(J,I):NEXTJ,I LB
·112 CLOSE2:GOSUB76:GOSUB68:RETURN PC
·120 DI=2:GOSUB86:GOSUB34:GOSUB38:GOSUB900 FI
·122 FORCR=R1TOR2:FORCC=C1TOC2:GOSUB62:PRINT#2,T$;:NEXT:PRINT#2:NEXT PB
·124 CLOSE2:GOSUB36:RETURN MD
·126 GOSUB34:GOSUB38:FORCR=R1TOR2:FORCC=C1TOC2:V$(CC,CR)="":V(CC,CR)=0:NEXTCC,CR PE
·128 GOSUB36:GOSUB76:GOSUB68:RETURN FH
·130 A$="FROM?":GOSUB16:T$=MID$(T$,6):GOSUB39:RS=R1:CS=C1 LI
·132 A$="TO?":GOSUB16:T$=MID$(T$,4):GOSUB39:TR$=LEFT$(TR$,1) IC
·134 FORC=C2TOC1STEP-1:FORR=R2TOR1STEP-1:V$(C,R)=V$(CS,RS):V(C,R)=V(CS,RS):NEXTR,C BE
·136 GOTO54 PP
·140 A$="QUITTING -- ARE YOU[3" "]SURE?":GOSUB16:T$=MID$(T$,28,1) HN
·142 IFT$="[s Y]"ORT$="Y"THENPRINT"[CLEAR]":END OC
·144 GOSUB68:RETURN MP
·800 CM$=LEFT$(T$,3):T$=MID$(T$,5):GOSUB40 MB
·802 SU=0:AV=0:MA=-1E20:MI=1E20:K=0 ML
·804 FORII=R1TOR2:FORJJ=C1TOC2:IFV$(JJ,II)=""THEN812 PK
·806 IFASC(V$(JJ,II))=39THEN812 BG
·808 V=V(JJ,II):SU=SU+V:K=K+1:IFMA<VTHENMA=V OO
·810 IFMI>VTHENMI=V GJ
·812 NEXTJJ,II:IFK<>0THENAV=SU/K EJ
·814 V=-(CM$="SUM")*SU-(CM$="AVG")*AV-(CM$="MAX")*MA-(CM$="MIN")*MI:F$=STR$(V) OP
·816 I=I+1:N$=MID$(OL$,I,1):IFI>=OLTHEN922 EJ
·818 IFN$<>")"THEN816 CN
·820 I=I+1:N$=MID$(OL$,I,1):N=0:IFN$<>""THENN=ASC(N$) EB
·822 GOTO910 CK
·890 GOSUB34:FORCR=0TOBR:FORCC=0TOBC:GOSUB900 GE
·892 GOSUB84:NEXTCC,CR:GOSUB36:RETURN OF
·900 IFV$(CC,CR)=""THENRETURN OB
·901 IFASC(V$(CC,CR))<>61THENRETURN NN
·902 B$="V("+STR$(CC)+","+STR$(CR)+")" LJ
·904 OL$=V$(CC,CR):OL=LEN(OL$) MA
·906 F$=LEFT$(OL$,1):F=ASC(F$):I=2 FC
·908 N$=MID$(OL$,I,1):N=ASC(N$):IF(F>64ANDF<91)AND(N>47ANDN<58)THEN914 HK
·909 IFF$="@"THENT$=MID$(OL$,I):GOTO800 KJ
·910 B$=B$+F$:F$=N$:F=N:I=I+1:IFI>OLTHENB$=B$+F$:GOTO924 GP
·912 GOTO908 DF
·914 C=F-65:R=VAL(MID$(OL$,I))-1:F$="V("+STR$(C)+","+STR$(R)+")":B$=B$+F$ AE
·916 I=I+1:IFI>OLTHEN924 NE
·918 F$=MID$(OL$,I,1):F=ASC(F$):IF(F>47ANDF<58)THEN916 HF
·920 IFI<OLTHENI=I+1:GOTO908 HN
·922 B$=B$+F$ JD
·924 XX=512:FORI=1TOLEN(B$):F$=MID$(B$,I,1):F=ASC(F$):IFF<>32THENPOKEXX,F:XX=XX+1 AD
·926 NEXTI:POKEXX,0:SYSCL:RETURN MD
·1000 W=40:NC=3:NR=17:MC=15:MR=30:SO=1024:LC=0:TR=0:CW=9:LL=213:POKE36876,200:POKE10,0 KH
·1002 DIMV(MC,MR),V$(MC,MR):GOSUB76:BL$="":FORI=1TO2*W:BL$=BL$+" ":NEXT:CH$="[HOME]"+BL$+"[HOME]" GK
·1004 BL$=LEFT$(BL$,CW):GOSUB84:Q$=CHR$(34):POKE54273,68:POKE54272,149:POKE54277,64 KF
·1005 POKE54278,128:POKE54296,15 OM
·1006 FORI=1TOW+1:CR$=CR$+"[RIGHT]":CD$=CD$+"[DOWN]":NEXT BH
·1010 REM PLACE FUNCTION DEF. HERE KF
·1090 GOSUB68:PL=65520:CL=PEEK(43)+5+PEEK(44)*256 OB
·1100 GETA$:IFA$=""THEN1100 KA
·1102 A=ASC(A$):IF(A>132ANDA<141)THENONA-132GOSUB54,50,126,130,96,108,120,140 NK
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 105 and 106 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·1104 IFA$<>"[UP]"THEN1112                                KB
·1106 IFCR=0THEN1172                                      JK
·1108 CR=CR-1:IFCR<TRTHENTR=TR-1:GOSUB76                  NC
·1110 GOTO1134                                            FE
·1112 IFA$<>"[DOWN]"THEN1120                              CB
·1114 IFCR=MRTHEN1172                                     PI
·1116 CR=CR+1:IFCR>TR+NRTHENTR=TR+1:GOSUB
 76                                                       GI
·1118 GOTO1134                                            FE
·1120 IFA$<>"[LEFT]"THEN1128                              JO
·1122 IFCC=0THEN1172                                      GF
·1124 CC=CC-1:IFCC<LCTHENLC=LC-1:GOSUB76                  JG
·1126 GOTO1134                                            FE
·1128 IFA$<>"[RIGHT]"THEN1136                             DG
·1130 IFCC=MCTHEN1172                                     MO
·1132 CC=CC+1:IFCC>LC+NCTHENLC=LC+1:GOSUB
 76                                                       AB
·1134 GOSUB56:GOTO1100                                    IP
·1136 IFA$<>"'"THEN1142                                   DE
·1138 SL=CW:GOSUB16                                       BP
·1140 V$(CC,CR)=T$:GOTO1154                               KE
·1142 IFA$<>"="THEN1148                                   DM
·1144 SL=2*W:GOSUB16                                      MC
·1146 V$(CC,CR)=T$:GOTO1154                               KE
·1148 IF(A$<"0"ORA$>"9")ANDA$<>"-"THEN116
0                                                         DB
·1150 SL=W:GOSUB16                                        OH
·1152 V$(CC,CR)="#":V(CC,CR)=VAL(T$)                      GA
·1154 GOSUB10:GOSUB900:GOSUB68:GOTO1100                   AK
·1160 IFA$<>">"THEN1168                                   ED
·1162 SL=W:A$="GOTO?":GOSUB16:T$=MID$(T$,
 6):GOSUB30:CC=C:CR=R                                     JF
·1163 IF(CC>(LC+NC))OR(CR>(TR+NR))THENLC=
 CC:TR=CR                                                 GL
·1164 IFTR>(MR-NR)THENTR=MR-NR                            PN
·1165 IFLC>(MC-NC)THENLC=MC-NC                            PL
·1166 GOSUB76:GOSUB68:GOTO1100                            MA
·1168 IFA$<>"[HOME]"THEN1172                              BE
·1170 CC=0:CR=0:LC=0:TR=0:GOSUB76:GOSUB68
 :GOSUB84:GOTO1100                                        CB
·1172 GOSUB5:GOTO1100                                     FG
```

SCREEN WINDOW
FROM PAGE 54

```
·10 S=49152:E=49940                                       CO
·11 POKE 53281,0:POKE 53280,0                             ED
·15 PRINT"[CLEAR][4"[DOWN]"] [YELLOW][RVS
 ON]PATIENCE IS A VIRTUE[4"."]"                           BH
·20 FORI=STOE:READ A:POKE I,A:NEXT                        DA
·30 PRINT"[HOME]SYS ";S;" :REM [4"*"] STA
 RT IT UP [4"*"]"                                         JN
·90 END                                                   IC
·1000 DATA 120,173,21,3,201,192,240,35,14
 1,164,194,173,20,3,141,163                               DL
·1001 DATA 194,169,192,141,21,3,169,45,14
 1,20,3,169,0,141,165,194                                 HG
·1002 DATA 141,168,194,169,255,141,166,19
 4,141,167,194,88,96,173,165,194                          JN
·1003 DATA 208,21,169,1,141,165,194,169,1
 92,72,169,80,72,8,72,72                                  MB
·1004 DATA 72,108,163,194,78,165,194,173,
 13,220,104,168,104,170,104,64                            AE
·1005 DATA 165,198,240,240,173,119,2,201,
 17,208,18,165,214,201,24,208                             BK
·1006 DATA 227,32,149,192,32,174,192,32,5
 9,193,76,68,192,201,145,208                              LC
·1007 DATA 16,165,214,208,207,32,149,192,
 32,203,192,32,50,193,76,68                               CD
·1008 DATA 192,201,133,208,6,32,149,192,7
 6,148,193,201,134,208,181,32                             GC
·1009 DATA 149,192,76,195,193,169,1,133,2
 04,165,207,240,12,165,206,174                            MJ
·1010 DATA 135,2,160,0,132,207,32,19,234,
 169,0,133,198,96,169,25                                  GH
·1011 DATA 133,214,32,33,193,48,18,32,68,
 193,240,246,32,57,194,208                                DM
·1012 DATA 241,32,107,194,240,3,32,122,19
 4,24,96,169,255,133,214,32                               LM
·1013 DATA 15,193,176,57,32,68,193,240,24
 6,32,57,194,208,241,165,97                               FI
·1014 DATA 133,95,165,98,133,96,240,37,16
 9,2,133,99,198,99,48,21                                  OJ
·1015 DATA 32,33,193,16,247,169,0,141,165
 ,2,32,129,233,165,217,9                                  CD
·1016 DATA 128,133,217,48,231,169,1,141,1
 46,2,32,122,194,24,96,166                                AM
·1017 DATA 214,232,224,25,176,10,181,217,
 16,247,160,0,32,240,255,24                               FH
·1018 DATA 96,166,214,202,48,11,181,217,1
 6,249,24,160,0,32,240,255                                OL
·1019 DATA 74,96,24,162,0,160,0,32,240,25
 5,96,24,162,24,160,0                                     BA
·1020 DATA 32,240,255,96,169,0,133,20,133
 ,21,168,192,40,176,61,177                                GG
·1021 DATA 209,200,201,32,240,245,201,48,
 144,50,201,58,176,46,233,47                              KH
·1022 DATA 133,99,165,21,133,100,201,25,1
 76,39,165,20,10,38,100,10                                II
·1023 DATA 38,100,101,20,133,20,165,100,1
 01,21,133,21,6,20,38,21                                  BI
·1024 DATA 165,20,101,99,133,20,144,195,2
 30,21,176,191,165,20,5,21                                BC
·1025 DATA 96,169,0,96,78,165,194,32,122,
 166,169,249,160,194,32,30                                PK
·1026 DATA 171,32,96,165,162,0,134,122,32
 ,124,165,162,0,189,0,2                                   ED
·1027 DATA 157,168,194,240,3,232,208,245,
 169,43,133,95,169,0,133,96                               IC
```

```
1028 DATA 76,218,193,78,165,194,32,122,1
66,173,166,194,133,20,173,167                FB
1029 DATA 194,133,21,32,57,194,240,2,176
,74,32,107,194,240,69,160                    GL
1030 DATA 4,162,0,132,99,189,168,194,240
,18,177,95,240,236,221,168                   GK
1031 DATA 194,240,5,164,99,200,208,233,2
32,200,208,233,169,147,32,210                GH
1032 DATA 255,162,10,160,0,32,240,255,32
,122,194,165,20,141,166,194                  BI
1033 DATA 165,21,141,167,194,32,174,192,
32,174,192,32,174,192,32,203                 BM
1034 DATA 192,76,51,194,169,6,160,195,32
,30,171,169,255,141,166,194                  OM
1035 DATA 141,167,194,78,165,194,108,2,3
,169,0,133,97,133,98,165                     AI
1036 DATA 43,133,95,165,44,133,96,165,21
,160,3,209,95,144,27,208                     HK
1037 DATA 9,165,20,136,209,95,144,18,240
,16,165,95,133,97,165,96                     HB
1038 DATA 133,98,32,107,194,208,224,169,
1,56,96,160,0,177,95,170                     EE
1039 DATA 200,177,95,134,95,133,96,177,9
5,96,160,2,177,95,133,20                     AE
1040 DATA 200,177,95,133,21,173,0,3,72,1
73,1,3,72,169,154,141                        DC
1041 DATA 0,3,169,194,141,1,3,76,189,166
,104,141,1,3,104,141                         DP
1042 DATA 0,3,96,0,255,255,0,0,255,190,0
,0,255,255,0,0                               MH
1043 DATA 255,255,0,0,255,255,0,0,255,25
5,0,0,255,255,0,0                            HG
1044 DATA 255,255,0,0,255,255,0,0,255,25
5,0,0,255,255,0,0                            HG
1045 DATA 255,255,0,0,255,255,0,0,255,25
5,0,0,255,255,0,0                            HG
1046 DATA 255,255,0,0,255,255,0,0,255,19
1,0,0,255,255,0,0                            EN
1047 DATA 255,255,0,0,255,255,0,0,255,19
,83,69,65,82,67,72                           EO
1048 DATA 32,70,79,82,58,0,19,78,79,84,3
2,70,79,85,78,68                             LG
1049 DATA 32,32,13,0,255                     EK
```

COMET CATCHING
FROM PAGE 18

COMET CATCHER

Note: Comet Catcher must be entered on the C-128 in 128 mode. Therefore *Bug Repellent* line codes cannot be provided.

```
 1 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
 2 REM        - COMET CATCHER -
 3 REM        RUPERT REPORT #28
 4 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
 5 REM     A REAL-WORLD SIMULATION
 6 REM   OF ASTRONOMICAL PROPORTIONS
 7 REM ================================
 8 REM  << WRITTEN FOR COMMODORE 128 >>
 9 REM ================================
10 GRAPHIC 2,1,22    :REM STANDARD SPLIT
   SCREEN GRAPHICS - TEXT AT LINE 22
20 COLOR 0,1 : COLOR 1,2  :REM TEXT BACK
   GROUND AND BIT-MAP FOREGROUND
30 COLOR 5,2          :REM TEXT COLOR
40 PRINT CHR$(27)"M" :REM NO SCROLL
50 WINDOW 0,22,39,24 :REM TEXT WINDOW
60 CIRCLE 1,160,100,1    :REM  DRAW SUN
70 SD=60*60*24        :REM # SECONDS/DAY
80 AU=1.496E11 :REM 1 ASTRONOMICAL UNIT
   IN METERS
90 G=6.67E-11       :REM  GRAV. CONSTANT
100 M=1.99E30       :REM MASS OF SUN (KG)
110 K=-G*M          :REM FORCE CONSTANT
120 REM ==== INITIAL PARAMETERS ========
130 TM=10     :REM TIME INCREMENT (DAYS)
140 HY=4      :REM HT. OF SCREEN IN AU
150 :REM INITIAL POSITION (AU)
160 X0=-3
170 Y0=1.2
180 :REM INITIAL VELOCITY (M/S)
190 VX=1.255E4
200 VY=0
210 REM ================================
220 DT=TM*SD   :REM TIME INCREMENT (SEC)
230 SF=200/HY/AU     :REM PLOTTING SCALE
   FACTOR
240 X=X0*AU : Y=Y0*AU        :REM CONVERT
   AU TO METERS
250 REM ******** MAIN LOOP **************
260 DD=2  :REM INITIAL TIME DIVISOR
270 R2=X*X+Y*Y : R=SQR(R2)
280 AX=K/R2*X/R : AY=K/R2*Y/R :REM ACCEL
 ERATION
290 VX=VX+AX*DT/DD : VY=VY+AY*DT/DD
300 X=X+VX*DT : Y=Y+VY*DT  :REM POSITION
310 PX=160+SF*X : PY=100-SF*Y  :REM PLOT
   VALUES
320 IF PX<0 OR PX>320 OR PY<0 OR PY>200
 THEN 340
330 DRAW ,PX,PY  : REM  PLOT THE POINT
340 T=T+DT
350 VE=SQR(2*G/R*M)/1E3 :REM ESCAPE VELO
 CITY
360 PRINT USING"VE= [4"#"].## KM/S";VE
370 VT=SQR(VX*VX+VY*VY)/1E3
380 PRINT USING"V = [4"#"].## KM/S";VT
390 PRINT USING"R = [4"#"].## AU";R/AU;
400 PRINT TAB(25) : PRINT USING"T= [5"#"
 ].# DAYS"; T/SD
410 DD=1        :REM  TIME DIVISOR
420 GET A$      :REM  CHECK KEYBOARD
430 REM  <RETURN> STOPS PROGRAM ...
440 ON INSTR(CHR$(13)+"PSF[UP][DOWN]",A$
```

```
) GOSUB  520,460,470,480,490,500
·450 GOTO 270
·460 GETKEY A$ : RETURN  :REM  PAUSE
·470 SLOW : RETURN
·480 FAST : RETURN
·490 DT=DT*1.1 : RETURN  :REM SPEED UP
·500 DT=DT*.9 : RETURN   :REM SLOW DOWN
·510 REM ******************************
·520 CHAR ,0,22,"VE= "+STR$(VE)+" KM/S"
·530 CHAR ,0,23,"V = "+STR$(VT)+" KM/S"
·540 CHAR ,0,24,"R = "+STR$(R/AU)+" AU"
·550 CHAR ,25,24,"T= "+STR$(T/SD)+" DAYS"
·560 CHAR ,0,0,"  [3"-"][3" "]BACK TO 80
 COLUMN MODE[3" "][3"-"]":   GRAPHIC 5
```

BITMAPPER

```
·1 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=   CD
·2 REM    =-=      BITMAPPER       =-=       IG
·3 REM      C-64 BIT MAP GRAPHICS             IM
·4 REM        RUPERT REPORT #28               PB
·5 REM ======  FOR C-64 ONLY!  =======        PC
·6 REM  USE -- GOSUB 1000 - TO INITIALIZE
  BIT MAP MODE                                LH
·7 REM  USE -- GOSUB 2000 - TO PLOT POINT
  AT LOCATION PX,PY                           OG
·8 REM  USE -- GOSUB 3000 - TO RETURN TO
  TEXT MODE                                   BK
·9 REM--------------------------------        EM
·10 GOSUB 1000                                FO
·20 REM -- DRAW SUN --  (OR ANY CIRCLE OF
  RADIUS R)                                   LO
·30 R=2 : FOR TH=0 TO 2*[PI] STEP .5           BL
·40 PX=160+R*COS(TH)                          GG
·50 PY=100-R*SIN(TH)                          DN
·60 NEXT                                      IA
·65 REM ********************************      DN
·66 REM LINES 440 THROUGH 485 ARE FOR USE     ON
·67 REM        WITH COMET CATCHER             GB
·68 REM ********************************      DN
·435 IF SD=0 THEN A$=CHR$(13)                 FE
·440 IF A$="" THEN 270                        DK
·450 IF A$=CHR$(13) THEN GOSUB 3000 : PRI
 NT CHR$(147) : END                           BC
·460 IF A$="[UP]" THEN GOSUB 490 : GOTO 2
 70                                           LO
·470 IF A$="[DOWN]" THEN GOSUB 500 : GOTO
  270                                         PB
·475 IF A$<>"P" THEN 270                      EB
·480 GET A$ : IF A$="" THEN 480               HD
·485 GOTO 270                                 CJ
·486 REM ********************************     DD
·960 REM                                      JD
·970 REM >>> INITIALIZE BIT MAP MODE <<<      DI
·980 REM                                      JD
·990 REM  SET & RESET BIT FUNCTIONS           HI
```

```
·1000 DEF FNSB(N)=PEEK(MM) OR 2[UPARROW]N     LG
·1010 DEF FNRB(N)=PEEK(MM) AND (255-2[UPA
 RROW]N)                                      FP
·1020 VV=53248 :REM VIC-II REGISTER 0         FL
·1030 :REM >>> PUT BIT MAP AT 8192 <<<
       (SET BIT 3 OF VIC REGISTER 24)         JA
·1040 MM=VV+24 : POKE MM,FNSB(3)              OO
·1050 :REM >>> SELECT BIT MAP MODE <<<
       (SET BIT 5 OF VIC REGISTER 17)         DH
·1060 MM=VV+17 : POKE MM,FNSB(5)              JM
·1070 BASE=8192 :REM START BIT MAP MEMORY     KD
·1080 :REM >>> CLEAR BIT MAP <<<              HG
·1090 GOSUB 4000                              FB
·1100 :REM > SELECT COLORS C1 AND C0   <      KO
·1110 C1=1 : C0=0 : CC=16*C1 + C0             OM
·1120 :REM FILL SCREEN MEMORY WITH COLOR      DP
·1130 FOR MM=1024 TO 2023:POKE MM,CC:NEXT     OM
·1140 RETURN                                  IM
·1150 :                                       DI
·1970 REM ============================        PE
·1980 REM > TURN ON PIXEL AT (PX,PY) <<<      GC
·1990 REM ============================        PE
·2000 BIT=7-(PX AND 7)                        EL
·2010 MM=BASE+320*INT(PY/8)+8*INT(PX/8)+(
 PY AND 7)                                    MM
·2020 POKE MM,FNSB(BIT)                       MO
·2030 RETURN                                  IM
·2040 :                                       DI
·2970 REM ========================            ME
·2980 REM >>> RESET BIT MAP MODE <<<          HI
·2990 REM ========================            ME
·3000 MM=VV+17 : POKE MM,FNRB(5)              JN
·3010 REM  RESTORE SCREEN MEMORY BASE         FI
·3020 MM=VV+24 : POKE MM,FNRB(3)              MP
·3030 RETURN                                  IM
·3960 :                                       DI
·3970 ==========================              DN
·3980 >>> CLEAR BIT MAP SCREEN <<<            PL
·3990 ==========================              DN
·4000 FOR M=828 TO 828+43                     JD
·4010 READ B : POKE M,B                       FE
·4020 CK=CK+B                                 ML
·4030 NEXT                                    IA
·4040 IF CK<>5133 THEN PRINT "DATA ERROR
 IN LINES 4060 - 4110" : STOP                 DO
·4050 SYS 828: RETURN                         AM
·4060 DATA 76,68,3,0,32,31,64,0               HC
·4070 DATA 173,63,3,133,251,173,64,3          MB
·4080 DATA 133,252,174,65,3,160,0,173         FE
·4090 DATA 67,3,136,145,251,208,251,230       OC
·4100 DATA 252,202,48,7,208,244,172,66        OE
·4110 DATA 3,208,239,96                       LJ
```

MR. MYSTO
FROM PAGE 36

MAIN PROGRAM

```
·10 GOSUB830                                                    CP
·20 DIMQ$(500),YN$(500)                                         MC
·30 B$="[32" "]"                                                LO
·40 D$="[HOME][20"[DOWN]"]"                                     FE
·50 S$="[13"[RIGHT]"]"                                          PJ
·60 DB$=D$+B$+B$+B$+B$+D$                                       BE
·70 YN$(0)="Y":Q$(0)="NOTHING":Q$=Q$(0)                         DG
·80 REM  MR. MYSTO   1-16-85   :    C-64
 VERSION 2.0                                                    HG
·90 PRINT"[CLEAR][3"[DOWN]"]","[4" "]MR.
 MYSTO",,"[6" "]BY FRANK DINUNZIO 1/16/19
 85"                                                            DA
·100 PRINT:PRINT"AN EXPERIMENT IN ARTIFIC
 IAL INTELLIGENCE"                                              IM
·110 PRINT,"S-START A NEW CATEGORY",,"G-G
 ET AN EXISTING ONE "                                           BC
·120 GETA$:IFA$<>"S"ANDA$<>"G"THEN120                           BF
·130 INPUT"[DOWN]WHICH CATEGORY";O$                             IF
·140 IFA$="G"THENGOSUB770                                       PC
·150 GOSUB520:PRINTDB$"DO YOU WANT ME TO
 SPEAK?":GOSUB1140:TK$=A$                                       IO
·160 PRINTDB$"[8" "]PICK A WORD AND I WIL
 L TRY",,"TO DEDUCE","WHAT IT IS."                              PC
·170 FORI=0TO3000:NEXT:GOSUB650                                 EG
·180 R=0:GU$=YN$(0):TA$=YN$(0)                                  JJ
·190 IFLEN(YN$(R))=LEN(TA$)THEN   IFYN$(R
 )=TA$THENRR=R:GOSUB1110:GOSUB650:GOTO220                       CN
·200 IFR<NTHENR=R+1:GOTO190                                     FA
·210 GOTO230                                                    CF
·220 GOSUB1140: TA$=TA$+A$:GOTO190                              JD
·230 IFA$="Y"THENPRINTDB$"I AM RIGHT AGAI
 N!":GOSUB650:FORT=1TO999:NEXT:GOTO410                          CJ
·240 PRINTDB$"I AM SORRY BUT I DO NOT KNO
 W THE ANSWER,PLEASE TELL ME WHAT IT IS."                       IK
·250 S=1:GOSUB650                                               KN
·260 INPUT"[6"[DOWN]"]";AA$:IFLEN(AA$)<18
 THEN280                                                        CJ
·270 PRINTDB$"ANSWER MUST BE LESS THAN 18
  LETTERS.":FORI=0TO999:NEXT:GOTO240                            FD
·280 TS$=Q$(RR):TL=LEN(TS$)                                     FD
·290 PRINTDB$"TYPE A QUESTION THAT WILL H
 ELP ME TELL",                                                  AO
·300 PRINTAA$" FROM "TS$                                        KD
·310 GOSUB650:INPUT"[6"[DOWN]"]";QQ$                            IG
·320 IFRIGHT$(QQ$,1)<>"?"THENQQ$=QQ$+"?"                        EF
·330 PRINTDB$"ENTER THE ANSWER FOR ",,AA$
 ;" (Y/N)":GOSUB650                                             EE
·340 GOSUB1140:N=N+1:P$=LEFT$(TA$,LEN(TA$
 )-1)                                                           PH
·350 TS$=Q$(RR):Q$(RR)=QQ$                                      CF
·360 X$="N":Z$="Y"                                              MD
·370 IFA$="N"THENX$="Y":Z$="N"                                  KG
·380 YN$(N)=P$+X$:Q$(N)=TS$:N=N+1                               MH
·390 YN$(N)=P$+Z$:Q$(N)= AA$                                    CE
·400 BU=BU+1:IFBU>10THENBU=0:GOSUB1160                          FO
·410 PRINTDB$"TRY ANOTHER?":GOSUB650:PRIN
 T"[5"[DOWN]"]"                                                 NB
·420 GETA$                                                      AL
·430 IFA$="N"ANDS=1THENGOSUB480:END                             JP
·440 IFA$="N"THENEND                                            FB
·450 IFA$="Y"THEN160                                            KM
·460 IFPEEK(197)=4THEN1020                                      IH
·470 GOTO420                                                    CG
·480 REM ******SAVE TO DISK******                               FP
·490 PRINTDB$"SAVING "O$                                        OE
·500 OPEN1,8,2,"@0:"+O$+",S,W":PRINT#1,N                        IL
·510 FORW=0TON:PRINT#1,YN$(W):PRINT#1,Q$(
 W):NEXT:CLOSE1:RETURN                                          AG
·520 REM ******DRAW FACE******                                  PC
·530 PRINT"[CLEAR]"S$"[15"[c P]"]"                              PL
·540 FORF=1TO6:PRINTS$"[c H]"S$"[c M]":NE
 XT                                                             AL
·550 PRINTS$"[3"[LEFT]"][3"[c Y]"][s O][1
 3"[c Y]"][s P][3"[c Y]"]"                                      JB
·560 FORF=1TO6:PRINTS$"[c G]"S$"[c M]":NE
 XT                                                             MO
·570 PRINTS$"[s M][13" "][s N]"                                 GP
·580 PRINTS$" [s M][11" "][s N]"                                GE
·590 PRINTS$"  [s M][9" "][s N]"                                HG
·600 PRINTS$"[3" "][s M][7" "][s N]"                            CH
·610 PRINTS$"[4" "][7"[c Y]"]"                                  BO
·620 PRINTS$"[HOME]"S$"[3"[RIGHT]"][4"[DO
 WN]"]MR. MYSTO"                                                LN
·630 PRINTS$"[4"[DOWN]"][4"[RIGHT]"]O[5"[
 RIGHT]"]O"                                                     CL
·640 PRINTS$"[DOWN][DOWN][7"[RIGHT]"][s Q
 ]"                                                             BA
·650 PRINTD$+S$"[5"[UP]"][10"[RIGHT]"]";:
 IFTK$="N"THENPRINT:RETURN                                      JG
·660 S=54272:FORE=STOS+28:POKEE,0:NEXT                          LA
·670 FORL=1TO10:   PRINT"[5"[LEFT]"][5"[c
  Y]"]";:PRINT"[5"[LEFT]"][s M][3"[c P]"]
 [s N]";                                                        GA
·680 FORJ=1TO50:NEXT                                            GF
·690 PRINT"[5"[LEFT]"][5"[c Y]"]";                              KE
·700 REM ******SOUND******                                     AK
·710 POKE54296, 10 :POKE54277, 85 :POKE54
 278, 245                                                       MP
·720 P=INT(RND(0)*5)                                            EO
·730 FORI=0TOP :REM INFLECTION                                  HE
·740 RN=INT(RND(0)*6):REM PITCH                                 ME
·750 POKE 54276, 33 :POKE 54273,RN+I                            LO
·760 NEXT I:NEXT:POKES+24,0:PRINT:RETURN                        LH
·770 REM ******LOAD ROUTINE******                              LO
·780 OPEN1,8,2,O$+",S,R"                                        CL
·790 OPEN15,8,15:INPUT#15,A$:IFA$<>"00"TH
 ENCLOSE15:CLOSE1:GOTO820                                       NH
·800 PRINT"LOADING "O$:INPUT#1,N                                MO
·810 FORW=0TON:INPUT#1,YN$(W),Q$(W):NEXT:
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 105 and 106 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
CLOSE1:RETURN                                                    FE
·820 PRINT"SORRY-CAN'T FIND "O$" ON THIS
 DISK":FORI=1TO3000:NEXT:RUN                                     BG
·830 PRINT"[CLEAR][13" "]MR. MYSTO                               FH
·840 PRINT"A STUDY IN SEMI-ARTIFICIAL INT
 ELLIGENCE[DOWN]"                                                AD
·850 PRINT"[3" "]THINK OF A WORD AND I WI
 LL TRY TO[4" "]GUESS WHAT IT IS ";                              JC
·860 PRINT"BY PROCESS OF LOGIC.                                  JH
·870 PRINT"[3" "]YOU WILL BE ASKED SOME C
 LUE QUESTIONSTHAT MUST BE ANSWERED ";                           AB
·880 PRINT"IN ORDER FOR ME TOGUESS THE WO
 RD. IF I DON'T GET IT RIGHT THE FIRST";                         FC
·890 PRINT" TIME, I WILL ASK YOU ONE MORE
  QUESTION AND THEN I GUARANTEE THAT YOU                         DO
·900 PRINT"WON'T FOOL ME ON THAT WORD ANY
  MORE.                                                          AO
·910 PRINT"[3" "]TO BEGIN YOU ARE ASKED T
 O CHOOSE IF  YOU WANT TO USE PREVIOUSLY"
 ;                                                               FB
·920 PRINT" STORED DATA  FROM DISK OR TO
 START A NEW SET OF DATA.I SUGGEST";                             NC
·930 PRINT" THAT A SEPERATE FORMATTED DIS
 KBE USED FOR THE DATA FILES.                                    BF
·940 PRINT"[3" "]AS THE PROGRAM RUNS IT B
 ECOMES WISER WITH EACH QUESTION ASKED.";                        BM
·950 PRINT" WHEN YOU ARE  FINISHED WITH T
 HE PROGRAM JUST TYPE 'N'";                                      FP
·960 PRINT" WHEN ASKED 'ANOTHER' AND THE
 COMPILED[3" "]KNOWLEDGE ";                                      CM
·970 PRINT"WILL BE SAVED TO DISK FOR[5" "
 ]FUTURE USE.                                                    KJ
·980 PRINT"REMEMBER - I AM NO SMARTER THA
 N THE USERWHO GIVES ME INFORMATION.                             GE
·990 PRINT,,"[6" "]<HIT ANY KEY>";                               NE
·1000 GETA$:IFA$=""THEN1000                                      LL
·1010 RETURN                                                     IM
·1020 REM ******EDIT ROUTINE******                               OF
·1030 PRINT"[CLEAR][DOWN]","C=CHANGE - N=
 NEXT[DOWN]"                                                     GC
·1040 FORZZ=0TON:PRINTZZ;Q$(ZZ),"(C/N)"                          MI
·1050 GETA$:IFA$<>"C"ANDA$<>"N"THEN1050                          PO
·1060 IFA$="N"THEN1090                                           JA
·1070 PRINT"[DOWN]CHANGE "Q$(ZZ)                                 AM
·1080 INPUT"[DOWN]TO ";Q$(ZZ)                                    GD
·1090 IFZZ=NTHEN:PRINT"[CLEAR]":GOSUB480:
 RUN20                                                           GI
·1100 NEXT                                                       IA
·1110 REM ******IS IT A QUESS*****                               GG
·1120 PRINTDB$;:IFRIGHT$(Q$(R),1)="?"THEN
 PRINTQ$(R):RETURN                                               BB
·1130 PRINT"IS YOUR WORD "Q$(R)"?":RETURN                        IG
·1140 GETA$:IFA$<>"Y"ANDA$<>"N"THEN1140                          PG
·1150 RETURN                                                     IM
·1160 REM ******SORT ROUTINE******                               DD
·1170 S=N:PRINTDB$"SORTING"                                      GC
·1180 S=INT(S/2)                                                 DM
·1190 IFS>=1THEN1220                                             OG
·1200 FR=FRE(0)-(SGN(FRE(0))<0)*65535:IFF
 R>1200THENRETURN                                                AN
·1210 PRINT"MEMORY LOW - START A NEW FILE
 ":GOSUB480:END                                                  MB
·1220 FORK=0TOS:FORI=KTON-SSTEPS                                 GK
·1230 J=I:T$=YN$(I+S):TT$=Q$(I+S)                                CE
·1240 IFLEN(T$)>=LEN(YN$(J))THEN1270                             JJ
·1250 YN$(J+S)=YN$(J):Q$(J+S)=Q$(J)                              GI
·1260 J=J-S:IFJ>1THEN1240                                        II
·1270 YN$(J+S)=T$:Q$(J+S)=TT$                                    MH
·1280 NEXT:NEXT:GOTO1180                                         NL
```

STATES DATA

```
·5 DIMYN$(30),Q$(30)                                             GN
·10 FORN=0TO21:READYN$(N),Q$(N):NEXT                             ME
·20 OPEN1,8,2,"@0:STATES,S,W"                                    OE
·30 PRINT#1,N-1                                                  BL
·40 FORW=0TON                                                    LK
·50 PRINT#1,YN$(W)                                               GI
·60 PRINT#1,Q$(W)                                                BP
·70 NEXT                                                         IA
·80 CLOSE1                                                       ND
·100 END                                                         IC
·120 DATA Y,IS IT A STATE?                                       GK
·130 DATA YN,NOTHING                                             GK
·140 DATA YY,IS IT WEST OF THE MISSISSIPP
 I?                                                              FL
·150 DATA YYN,IS IT SOUTH OF THE MASON DI
 XON LINE?                                                       PO
·160 DATA YYY,IS IT NORTH OF THE MASON DI
 XON LINE?                                                       GP
·170 DATA YYNN,IS IT ON THE GREAT LAKES?                         OK
·180 DATA YYNY,IS IT A PLAINS STATE?                             PG
·190 DATA YYYY,DOES IT BORDER ON CANADA?                         NK
·200 DATA YYNNN,IS IT IN NEW ENGLAND?                            BO
·210 DATA YYNNY,MICHIGAN                                         GH
·220 DATA YYNYY,TENNESSEE                                        FA
·230 DATA YYYYN,OREGON                                           CL
·240 DATA YYYNN,IS IT ON THE PACIFIC COAS
 T?                                                              CL
·250 DATA YYYNY,KANSAS                                           CE
·260 DATA YYNYN,FLORIDA                                          GH
·270 DATA YYYYY,WASHINGTON                                       DG
·280 DATA YYNNNY,MAINE                                           BO
·290 DATA YYNNNN,IS IT ON THE ATLANTIC CO
 AST?                                                            AM
·300 DATA YYYNNY,CALIFORNIA                                      HP
·310 DATA YYYNNN,LOUISIANA                                       FN
·320 DATA YYNNNNN,PENNSYLVANIA                                   GP
·330 DATA YYNNNNY,NEW JERSEY                                     LE
```

AIR RESCUE
FROM PAGE 35

Starting address in hex: C000
Ending address in hex: C6FA
SYS to start: 49152

***Flankspeed* required for entry! See page 106.**

```
C000: 20 C9 C8 20 0A C5 AD BB 0C
C008: 02 38 C9 05 90 24 EE AA 5F
C010: 02 20 4B C7 A2 01 A0 F4 7E
C018: 20 FA C4 A9 00 8D BB 02 EC
C020: A9 14 8D B6 02 AD AA 02 7E
C028: C9 06 D0 06 EE AC 02 20 8C
C030: 5A C6 AD 11 D0 10 FB AD 9A
C038: 1E D0 29 3F 8D B8 02 AD 85
C040: 1F D0 8D C0 02 20 E0 C1 43
C048: 20 BC C2 20 1B C3 20 A1 A8
C050: C6 20 1D C4 20 59 C4 AD 05
C058: C0 02 29 01 F0 03 4C 57 DC
C060: C1 AD B8 02 F0 A0 C9 12 F7
C068: D0 22 A2 04 20 A1 C1 A9 2F
C070: 00 8D AB 02 A9 FD 2D 15 95
C078: D0 8D 15 D0 A2 00 A0 32 32
C080: 20 FA C4 CE B6 02 AD 1E B3
C088: D0 4C 06 C0 AD B8 02 C9 9E
C090: 0A D0 22 A2 03 20 A1 C1 B6
C098: A9 00 8D AB 02 A9 FD 2D 52
C0A0: 15 D0 8D 15 D0 A2 00 A0 3D
C0A8: 19 20 FA C4 CE B6 02 AD D6
C0B0: 1E D0 4C 06 C0 AD B8 02 1B
C0B8: C9 21 D0 12 AD FD 07 C9 03
C0C0: F8 D0 03 4C 57 C1 A9 D0 6D
C0C8: 8D 0C D0 4C 06 C0 AD B8 AC
C0D0: 02 C9 06 D0 35 A9 02 8D E1
C0D8: B1 02 A9 FA 85 A2 A9 03 06
C0E0: 8D 21 D0 A5 A2 D0 FC A9 20
C0E8: 00 8D AB 02 8D 21 D0 A2 46
C0F0: 00 A0 4B 20 FA C4 A9 FD 64
C0F8: 2D 15 D0 8D 15 D0 CE B6 05
C100: 02 20 33 C6 AD 1E D0 4C 05
C108: 06 C0 AD B8 02 C9 05 D0 D6
C110: 30 AD 01 D0 C9 82 F0 07 04
C118: AD B1 02 C9 02 D0 22 AD E5
C120: 25 D0 29 0F D0 03 4C 06 74
C128: C0 A2 00 A0 C8 20 FA C4 D4
C130: EE BB 02 A9 00 8D 25 D0 0A
C138: 20 5A C6 AD 1E D0 4C 06 68
C140: C0 AD B8 02 C9 05 F0 0F 38
C148: C9 09 F0 0B C9 11 F0 07 E9
C150: C9 31 F0 03 4C 06 C0 A2 F4
C158: 00 20 A1 C1 20 33 C6 20 16
C160: 74 C6 4C 91 C1 A9 00 8D 72
C168: AA 02 8D BB 02 8D AE 02 9E
C170: 8D AF 02 A9 0B 8D 21 D0 E3
C178: 20 28 C6 A5 C5 C9 04 D0 91
C180: FA 20 99 C5 A9 03 8D AC E1
C188: 02 A9 00 8D 21 D0 4C 06 06
C190: C0 CE AC 02 F0 CF A9 20 59
C198: 20 99 C5 AD 1E D0 4C 06 07
C1A0: C0 BD F8 07 8D B9 02 A9 12
C1A8: F4 9D F8 07 BD C1 C8 0D 90
C1B0: 1C D0 8D 1C D0 8A 48 20 0B
C1B8: 33 C6 68 AA A0 F9 84 A2 87
C1C0: A4 A2 D0 FC BD C1 C8 4D 6B
C1C8: 1C D0 8D 1C D0 A9 FF 38 12
C1D0: FD C1 C8 2D 15 D0 8D 15 0F
C1D8: D0 AD B9 02 9D F8 07 60 11
C1E0: AD 0C D0 C9 A0 D0 09 AD 5D
C1E8: 15 D0 29 FD 8D 15 D0 60 C9
C1F0: AD 00 DC AA 29 0F C9 0B 33
C1F8: D0 0E AD 00 D0 38 C9 1E 76
C200: 90 4F CE 00 D0 4C 29 C2 B7
C208: C9 07 D0 0E AD 00 D0 38 6E
C210: C9 EB B0 3D EE 00 D0 4C BF
C218: 29 C2 C9 0E D0 1B AD 01 76
C220: D0 38 C9 4B 90 2B CE 01 C9
C228: D0 CE BA 02 D0 08 A9 14 1B
C230: 8D BA 02 CE 0C D0 4C 51 C3
C238: C2 C9 0D D0 14 AD 01 D0 36
C240: 38 C9 C8 B0 0C EE 01 D0 88
C248: EE 03 D0 EE 03 D0 4C 29 43
C250: C2 AD B6 02 F0 46 AD AB 0A
C258: 02 D0 24 8A 29 10 D0 3C 20
C260: AD 00 D0 18 69 0F 8D 02 FE
C268: D0 AD 01 D0 18 69 0F 8D D6
C270: 03 D0 A9 02 0D 15 D0 8D 70
C278: 15 D0 A9 01 8D AB 02 EE 33
C280: 28 D0 EE 02 D0 F0 05 EE 20
C288: 02 D0 D0 10 A9 FD 2D 15 26
C290: D0 8D 15 D0 A9 00 8D AB B7
C298: 02 CE B6 02 CE B2 02 D0 76
C2A0: 1A A9 06 8D B2 02 A9 03 59
C2A8: 38 ED B0 02 8D B0 02 18 D9
C2B0: 69 F4 8D F8 07 AD 01 D0 1C
C2B8: 8D 00 D4 60 AD B4 02 30 10
C2C0: 59 AE AD 02 CA 8E AD 02 81
C2C8: D0 3C A9 05 8D AD 02 AE 70
C2D0: 0A D0 CA E0 FF D0 2C AD 02
C2D8: 10 D0 29 20 D0 1D AD B4 53
C2E0: 02 49 FF 10 16 8D B4 02 96
C2E8: AD 15 D0 29 DF 8D 15 D0 F8
C2F0: A9 DF 2D 10 D0 8D 10 D0 F6
C2F8: 4C 06 C3 A9 20 4D 10 D0 07
C300: 8D 10 D0 CE 0A D0 CE BF A6
C308: 02 D0 0F A9 04 8D BF 02 E6
C310: AD AD 02 0A 18 69 4B 8D D1
C318: 0B D0 60 AE 04 D0 CA E0 83
C320: FF D0 26 AD 10 D0 29 04 D2
C328: D0 17 A9 04 0D 15 D0 8D 3E
C330: 15 D0 A9 06 8D B1 02 A9 B0
C338: 38 8D 04 D0 A9 01 8D 25 30
C340: D0 AD 10 D0 49 04 8D 10 8A
C348: D0 CE 04 D0 AD B0 02 18 35
C350: 6D B1 02 8D 26 D0 AD B0 54
C358: 02 18 69 FC 8D FA 07 60 C8
C360: AD B4 02 10 79 AD B7 02 B5
```

```
C368: D0 35 AD 10 D0 29 10 D0 07
C370: 6D AD 08 D0 38 C9 FA 90 F1
C378: 65 AD 09 D0 8D 0B D0 AD 7C
C380: 08 D0 38 E9 05 8D 0A D0 E8
C388: A9 01 8D B7 02 A9 20 0D 51
C390: 15 D0 8D 15 D0 A9 F8 8D 1A
C398: FD 07 A9 01 8D 2C D0 AD 80
C3A0: 0A D0 38 E9 0F 8D 0A D0 15
C3A8: B0 34 AD 15 D0 29 DF 8D B7
C3B0: 15 D0 A9 FF 8D 0A D0 A9 52
C3B8: 50 8D 0B D0 AD 10 D0 09 0A
C3C0: 20 8D 10 D0 A9 00 8D B7 3E
C3C8: 02 8D B4 02 A9 07 8D 2C 79
C3D0: D0 A9 F9 8D FD 07 A9 20 A1
C3D8: 0D 15 D0 8D 15 D0 AD AA 97
C3E0: 02 18 69 03 8D B5 02 AA 57
C3E8: CA 8E C1 02 AD 08 D0 38 C4
C3F0: ED B5 02 8D 08 D0 B0 1B C8
C3F8: A9 10 4D 10 D0 8D 10 D0 4F
C400: 29 10 F0 0E AD 01 D0 8D 45
C408: 09 D0 A9 10 0D 15 D0 8D 1C
C410: 15 D0 60 8D 08 D0 8D 09 53
C418: D4 8D 05 D4 60 20 60 C3 F8
C420: AD 06 D0 38 ED C1 02 8D 1C
C428: 06 D0 8D 0A D4 8D 07 D4 D4
C430: B0 26 A9 08 4D 10 D0 8D 74
C438: 10 D0 AD 10 D0 29 08 F0 C9
C440: 16 20 97 E0 A5 8B 8D 07 B4
C448: D0 38 E9 71 8D 2A D0 A9 DE
C450: 08 0D 15 D0 8D 15 D0 60 1F
C458: 60 AD AE 02 38 CD A8 02 C7
C460: F0 04 90 19 B0 0B AD AF 18
C468: 02 38 CD A9 02 F0 02 90 9F
C470: 0C AD AE 02 8D A8 02 AD C0
C478: AF 02 8D A9 02 A2 01 A0 A7
C480: 06 18 20 F0 FF A9 12 20 8B
C488: D2 FF A9 05 20 D2 FF AD AA
C490: A8 02 AE A9 02 20 CD BD 41
C498: A2 01 A0 22 18 20 F0 FF 28
C4A0: AD AE 02 AE AF 02 20 CD 4D
C4A8: BD A9 9A 20 D2 FF A2 02 42
C4B0: A0 08 18 20 F0 FF AE AC DD
C4B8: 02 A9 00 20 CD BD A2 02 B4
C4C0: A0 11 18 20 F0 FF AE B6 01
C4C8: 02 A9 00 20 CD BD A9 20 E9
C4D0: 20 D2 FF A9 05 20 D2 FF 65
C4D8: A2 02 A0 19 18 20 F0 FF 60
C4E0: AE BB 02 A9 00 20 CD BD A2
C4E8: A2 02 A0 24 18 20 F0 FF 7B
C4F0: AE AA 02 E8 A9 00 20 CD CC
C4F8: BD 60 98 18 6D AF 02 8D 74
C500: AF 02 8A 6D AE 02 8D AE 96
C508: 02 60 A9 8E 20 D2 FF A9 3F
C510: 08 20 D2 FF AD 16 D0 29 C8
C518: F7 8D 16 D0 20 F3 C6 A9 09
C520: 00 8D A8 02 8D A9 02 8D 1F
C528: AE 02 8D AF 02 8D 17 D0 8D
C530: 8D 21 D0 8D AA 02 8D BB 33
C538: 02 A9 02 8D B0 02 8D BF 73
C540: 02 8D B2 02 8D BA 02 A9 78
C548: 04 8D 1B D0 A9 FC 8D F9 F3
C550: 07 A9 FD 8D FA 07 A9 FA 33
C558: 8D FC 07 8D FB 07 A9 F9 1E
C560: 8D FD 07 A9 FB 8D FE 07 2C
C568: A9 F7 8D FF 07 A9 0F 8D E4
C570: 27 D0 8D 20 D0 A9 03 8D 21
C578: 28 D0 8D AC 02 A9 0A 8D EE
C580: 2A D0 A9 07 8D 26 D0 8D 3E
C588: 2C D0 A9 01 8D 2E D0 A9 66
C590: 02 8D 2B D0 A9 0B 8D 2D 8B
C598: D0 A9 E1 8D 1D D0 A9 06 20
C5A0: 8D 1C D0 8D A7 02 8D B1 91
C5A8: 02 A9 93 20 D2 FF 20 5E 59
C5B0: C7 20 4B C7 A9 F5 8D F8 D1
C5B8: 07 A9 01 8D 25 D0 A9 34 CB
C5C0: 8D 10 D0 A9 00 8D AB 02 14
C5C8: A9 14 8D B6 02 A2 0F BD 3C
C5D0: B1 C8 9D 00 D0 CA 10 F7 8C
C5D8: A9 FD 8D 15 D0 20 28 C6 03
C5E0: A9 06 8D 18 D4 A9 01 8D 43
C5E8: 01 D4 A9 4B 8D 00 D4 A9 BF
C5F0: 81 8D 05 D4 A9 81 8D 06 98
C5F8: D4 A9 21 8D 04 D4 A9 07 AF
C600: 8D 08 D4 A9 06 8D 07 D4 83
C608: A9 81 8D 0C D4 A9 81 8D 5A
C610: 0D D4 A9 64 8D 0A D4 A9 16
C618: 64 8D 09 D4 A9 41 8D 0B 6B
C620: D4 AD 1E D0 AD 1F D0 60 8F
C628: A2 18 A9 00 9D 00 D4 CA C9
C630: D0 FA 60 A9 09 8D 0F D4 80
C638: A9 05 8D 0E D4 A9 09 8D 97
C640: 13 D4 A9 F0 8D 12 D4 A9 E0
C648: 81 8D 12 D4 A9 F0 85 A2 01
C650: A5 A2 D0 FC A9 80 8D 12 30
C658: D4 60 A9 39 8D 0F D4 A9 8B
C660: 63 8D 0E D4 A9 09 8D 13 87
C668: D4 A9 0A 8D 14 D4 A9 11 22
C670: 8D 12 D4 60 A9 08 8D 13 97
C678: D4 A9 FF 8D 14 D4 A9 15 2C
C680: 8D 12 D4 A2 28 A0 28 8C 15
C688: 0E D4 88 D0 FA A9 FC 85 EB
C690: A2 A5 A2 D0 FC 8E 0F D4 BB
C698: CA D0 EA A9 00 8D 12 D4 3D
C6A0: 60 AD 16 D0 29 F8 18 6D 3D
C6A8: A7 02 8D 16 D0 AE A7 02 1F
C6B0: CA 10 05 20 BC C6 A2 07 DD
C6B8: 8E A7 02 60 AD 11 D0 10 F0
C6C0: FB A9 E0 85 FB A9 05 85 FC
C6C8: FC A2 08 A0 00 C8 B1 FB 87
C6D0: 88 91 FB C8 C0 27 D0 F5 5E
C6D8: A0 00 B1 FB A0 27 91 FB 7C
C6E0: 18 A5 FB 69 28 85 FB A5 53
C6E8: FC 69 00 85 FC A0 00 CA 3D
C6F0: D0 DB 60 78 A9 12 8D 14 D3
C6F8: 03 A9 C7 6D
```

SWOOP
FROM PAGE 53

Starting address in hex: C000
Ending address in hex: C6DE
SYS to start: 49152

***Flankspeed* required for entry! See page 106.**

```
C000: A0 00 B9 9F C5 99 80 3E 18
C008: C8 D0 F7 B9 9F C6 99 80 D3
C010: 3F C8 10 F7 A0 40 A9 00 AA
C018: 99 C0 3F 88 10 FA A0 18 FD
C020: A9 C3 99 E5 3F 88 88 88 E5
C028: 10 F8 A9 00 8D 7E C5 A9 56
C030: C9 8D 01 D0 78 A9 7F 8D 88
C038: 0D DC A9 01 8D 1A D0 A9 EE
C040: DF 8D 12 D0 A9 1B 8D 11 F3
C048: D0 A9 FB 8D 14 03 A9 C3 D0
C050: 8D 15 03 58 A0 00 B9 86 2F
C058: C5 99 00 D4 C8 C0 19 90 BF
C060: F5 A9 CA A0 C4 20 1E AB 1A
C068: 18 A2 16 A0 03 20 F0 FF ED
C070: A9 CE A0 C4 20 1E AB 20 58
C078: 58 C3 A9 33 8D DF 07 A9 8F
C080: 14 8D 85 C5 A9 B0 8D 2B 80
C088: C5 8D 00 D0 A9 00 8D 2C 10
C090: C5 8D 10 D0 18 A2 18 A0 38
C098: 09 20 F0 FF 38 A9 15 ED 97
C0A0: 85 C5 AA A9 00 20 CD BD EB
C0A8: A9 20 20 D2 FF A9 FD 8D 9A
C0B0: F8 07 8D 84 C5 A9 FF 8D BF
C0B8: FF 07 A9 0D 8D 27 D0 A0 9C
C0C0: 05 AD 1B D4 29 1F 99 2D 72
C0C8: C5 88 10 F5 A2 07 AD 15 89
C0D0: D0 49 01 8D 15 D0 A5 A2 A7
C0D8: 18 69 0A C5 A2 D0 FC CA 65
C0E0: D0 EC 8E 84 C5 A9 7F 8D 2E
C0E8: 15 D0 A9 37 85 02 A2 05 DE
C0F0: BD 2D C5 C9 37 B0 03 FE 55
C0F8: 2D C5 CA 10 F3 20 0D C3 AB
C100: A5 A2 C5 A2 F0 FC C6 02 67
C108: D0 E4 AD 1E D0 AD 1F D0 F7
C110: A2 05 AD 8D 02 8D 84 C5 CC
C118: D0 F8 BD 2D C5 F0 43 18 DE
C120: 7D 42 C5 C9 D0 B0 1D C9 D7
C128: 34 90 19 9D 2D C5 18 BD 6C
C130: 34 C5 7D 48 C5 A8 BD 3B 57
C138: C5 7D 4E C5 85 02 D0 0A F1
C140: C0 1C B0 0A 20 39 C3 4C 41
C148: 62 C1 C0 40 B0 F6 AD 1B DD
C150: D4 29 3F F0 EF 98 9D 34 D8
C158: C5 A5 02 F0 02 A9 01 9D 01
C160: 3B C5 CA 10 AD AD 33 C5 90
C168: F0 1F 18 69 04 C9 CB 90 24
C170: 0A AD 15 D0 29 7F 8D 15 59
C178: D0 A9 00 8D 33 C5 AD 1B 42
C180: D4 09 08 8D 2E D0 4C CF 0F
C188: C1 A2 05 38 BD 34 C5 ED CF
C190: 2B C5 85 02 BD 3B C5 ED B5
C198: 2C C5 05 02 F0 06 CA 10 63
C1A0: EA 4C CF C1 AD 15 D0 3D 3A
C1A8: 64 C5 F0 23 BD 2D C5 C9 61
C1B0: AA B0 1C 8D 33 C5 BD 34 A0
C1B8: C5 8D 3A C5 BD 3B C5 8D 58
C1C0: 41 C5 A9 FF 8D FF 07 AD B3
C1C8: 15 D0 09 80 8D 15 D0 AD 59
C1D0: 00 DC 29 10 D0 41 AD 7E 25
C1D8: C5 D0 3C 38 AD 2B C5 E9 6C
C1E0: 1C AA 85 02 AD 2C C5 E9 B8
C1E8: 00 4A 66 02 46 02 46 02 2C
C1F0: 18 A5 02 69 FA 85 FB 85 1C
C1F8: FD A9 06 69 00 85 FC 18 AA
C200: 69 D4 85 FE 8A 29 07 4A C7
C208: AA BD 7F C5 8D 83 C5 A9 36
C210: 15 8D 04 D4 EE 7E C5 AD 6C
C218: 1F D0 29 FE F0 62 85 02 0B
C220: 8D 84 C5 A9 81 8D 0B D4 90
C228: A0 06 06 02 90 2F A9 FE 3F
C230: 99 F9 07 84 04 20 85 C3 BC
C238: A4 04 A2 03 AD 1B D4 29 4D
C240: 07 F0 F9 99 28 D0 A5 A2 0D
C248: C5 A2 F0 FC CA D0 ED AD D5
C250: 15 D0 39 6B C5 8D 15 D0 14
C258: A9 00 99 2D C5 88 10 CA F1
C260: A0 00 AD 7E C5 F0 0B C9 B8
C268: 14 B0 07 8C 7E C5 A9 20 CE
C270: 91 FB 8C 84 C5 8C 04 D4 3A
C278: A9 80 8D 0B D4 AD 1F D0 AD
C280: AD 15 D0 29 FE D0 13 38 58
C288: AD 85 C5 E9 01 C9 08 90 CE
C290: 03 8D 85 C5 20 BB C3 4C 58
C298: 94 C0 AE 85 C5 A0 00 C8 51
C2A0: D0 FD CA D0 FA AD 1E D0 A2
C2A8: 29 01 F0 5B 8D 84 C5 A9 A0
C2B0: FE 8D F8 07 A9 81 8D 0B 01
C2B8: D4 A2 03 AD 1B D4 29 07 01
C2C0: F0 F9 8D 27 D0 A5 A2 C5 3F
C2C8: A2 F0 FC CA D0 ED A9 80 0D
C2D0: 8D 0B D4 AD 7E C5 F0 09 2A
C2D8: C9 13 B0 05 A9 14 8D 7E 35
C2E0: C5 A2 00 8E 84 C5 AD 15 E4
C2E8: D0 29 7E 8D 15 D0 20 BB B0
C2F0: C3 8E 15 D0 CE DF 07 AD 8C
C2F8: DF 07 29 0F D0 06 20 C5 D4
C300: C3 4C 77 C0 4C 84 C0 20 F9
C308: 0D C3 4C 10 C1 A2 06 8A 2A
C310: 0A A8 BD 2D C5 99 03 D0 E0
C318: BD 34 C5 99 02 D0 BD 3B 35
C320: C5 F0 09 AD 10 D0 1D 64 EF
C328: C5 4C 32 C3 AD 10 D0 3D FB
```

All the programs in this issue of *Ahoy!* are available on disk or cassette. See page 60.

Entry of hexadecimal program listings in *Ahoy!* requires the use of our *Flankspeed* machine language entry program. See the documentation and listing for *Flankspeed* on page 106.

```
C330: 6B C5 8D 10 D0 CA 10 D7 82
C338: 60 AD 1B D4 29 07 A8 B9 C8
C340: 54 C5 9D 42 C5 AD 1B D4 9D
C348: 29 07 A8 B9 54 C5 9D 48 DA
C350: C5 B9 5C C5 9D 4E C5 60 04
C358: A0 00 B9 79 07 D9 8F 07 A3
C360: F0 05 B0 0B 4C 7A C3 C8 65
C368: C0 06 90 EE 4C 7A C3 A0 D9
C370: 05 B9 79 07 99 8F 07 88 68
C378: 10 F7 A0 05 A9 30 99 79 13
C380: 07 88 10 FA 60 AD 7B 07 AB
C388: 48 38 A9 15 ED 85 C5 AA AB
C390: A0 04 38 B9 79 07 69 00 11
C398: C9 3A 90 02 A9 30 99 79 1C
C3A0: 07 88 10 EF CA D0 E9 68 1E
C3A8: CD 7B 07 F0 0D 18 AD DF 9C
C3B0: 07 69 01 C9 3A B0 03 8D 67
C3B8: DF 07 60 A5 A2 18 69 64 2E
C3C0: C5 A2 D0 FC 60 A2 0A A0 A4
C3C8: 0E 18 20 F0 FF A9 1B A0 65
C3D0: C5 20 1E AB A0 0C B9 9E 85
C3D8: D9 49 05 99 9E D9 88 10 AB
C3E0: F5 A5 A2 18 69 14 C5 A2 1D
C3E8: D0 FC A5 CB C9 40 F0 E4 08
C3F0: A0 0C A9 20 99 9E 05 88 2D
C3F8: 10 FA 60 A9 01 8D 19 D0 86
C400: A2 0B A0 0B AD 12 D0 10 F9
C408: 03 4C B4 C4 AD 84 C5 F0 B9
C410: 03 4C B0 C4 EE 72 C5 AD A9
C418: 72 C5 29 07 D0 14 EE 79 CD
C420: C5 AD 79 C5 29 03 AA BD 67
C428: 7A C5 A0 05 99 F9 07 88 31
C430: 10 FA AD 00 DC 29 0C 49 44
C438: 0C 4A 4A A8 18 AD 2B C5 38
C440: 79 73 C5 AA AD 2C C5 79 B6
C448: 76 C5 F0 07 E0 40 B0 1B 69
C450: 4C 57 C4 E0 1C 90 14 8E E8
C458: 2B C5 8E 00 D0 8D 2C C5 28
C460: AD 10 D0 29 FE 0D 2C C5 16
C468: 8D 10 D0 AD 7E C5 F0 40 F9
C470: A0 00 A9 20 91 FB 38 A5 46
C478: FB E9 28 85 FB 85 FD A5 31
C480: FC E9 00 85 FC 18 69 D4 40
C488: 85 FE AD 7E C5 C9 14 90 6D
C490: 0B 8C 7E C5 A9 14 8D 04 BB
C498: D4 4C B0 C4 A9 01 91 FD 69
C4A0: AD 83 C5 91 FB EE 7E C5 58
C4A8: AD 7E C5 0A 0A 8D 01 D4 12
C4B0: A2 DF A0 00 8E 12 D0 8C D1
C4B8: 21 D0 8C 20 D0 AD 0D DC BF
C4C0: 29 01 F0 03 4C 31 EA 4C 93
C4C8: BC FE 8E 08 93 00 9E 53 A0
C4D0: 43 4F 52 45 20 05 30 30 80
C4D8: 30 30 30 30 20 20 20 20 1A
C4E0: 20 20 20 20 20 20 9F 48 89
C4E8: 49 47 48 20 20 05 30 30 67
C4F0: 30 30 30 30 0D 0D 20 20 0C
C4F8: 20 96 4C 45 56 45 4C 20 49
C500: 05 31 20 20 20 20 20 20 F6
C508: 20 20 20 20 20 20 20 20 09
C510: 20 99 4C 49 56 45 53 20 6E
C518: 05 33 00 1C 50 52 45 53 A7
C520: 53 20 41 4E 59 20 4B 45 2D
C528: 59 05 00 00 00 00 00 00 86
C530: 00 00 00 00 2D 5A 9B AA FD
C538: D7 F0 1E 00 00 00 00 00 1F
C540: 00 01 01 01 FF FF 01 01 45
C548: 01 00 01 00 01 00 00 00 4B
C550: 00 00 00 00 01 FF 02 FE 52
C558: 03 FD 01 FE 00 FF 00 FF 59
C560: 00 FF 00 FF 02 04 08 10 7E
C568: 20 40 80 FD FB F7 EF DF 0B
C570: BF 7F FF 00 FE 02 00 FF B0
C578: 00 00 FA FB FC FB 00 74 DC
C580: 54 42 6A 00 00 00 00 00 81
C588: 00 00 00 00 F0 00 08 00 81
C590: 00 00 00 F9 00 FF 00 00 8A
C598: 80 00 00 00 28 01 9F 00 E1
C5A0: 00 00 00 00 00 00 00 00 A0
C5A8: 20 00 04 10 00 08 18 00 FC
C5B0: 18 18 00 18 0C 81 30 0E C4
C5B8: 42 70 07 66 E0 07 99 E0 3B
C5C0: 03 FF C0 00 FF 00 00 3C C0
C5C8: 00 00 00 00 00 00 00 00 C8
C5D0: 00 00 00 00 00 00 00 00 D0
C5D8: 00 00 00 00 00 00 00 00 D8
C5E0: 00 00 00 00 00 00 00 00 E0
C5E8: 00 00 00 00 00 00 00 00 E8
C5F0: 00 00 81 00 00 42 00 0E C2
C5F8: 66 70 3F 99 FC 63 FF C6 CF
C600: 80 FF 01 00 3C 00 00 00 BD
C608: 00 00 00 00 00 00 00 00 08
C610: 00 00 00 00 00 00 00 00 10
C618: 00 00 00 00 00 00 00 00 18
C620: 00 00 00 00 00 00 00 00 20
C628: 00 00 00 00 81 00 00 42 EB
C630: 00 00 66 00 03 99 C0 0F 03
C638: FF F0 1E FF 78 18 3C 18 2C
C640: 30 00 0C 30 00 0C 20 00 D8
C648: 04 20 00 04 00 00 00 00 70
C650: 00 00 00 00 00 00 00 00 50
C658: 00 00 00 00 00 00 00 00 58
C660: 00 00 00 3C 00 00 3C 00 D8
C668: 00 3C 00 00 3C 00 00 3C 1D
C670: 00 00 7E 00 00 3C 00 00 2B
C678: 7E 00 00 7E 00 00 FF 00 75
C680: 08 7E 10 18 FF 18 39 FF 80
C688: 9C 3B FF DC 3F FF FC 3E B7
C690: 7E 7C 3C 3C 3C 38 00 1C 94
C698: 38 00 1C 38 00 1C 00 00 41
C6A0: 00 00 00 24 00 01 99 80 DF
C6A8: 00 42 00 10 66 08 0E 24 9B
```

```
C6B0: 70 43 00 C2 06 00 60 E0 6E
C6B8: 22 07 3B 00 DC 01 00 80 7B
C6C0: 30 08 0C 62 81 46 0C 24 5F
C6C8: 30 30 66 0C 01 C3 80 17 F7
C6D0: 18 E8 04 24 20 00 66 00 80
C6D8: 00 00 00 00 00 00 00 D8
```

CHRONO-WEDGE
FROM PAGE 61

Starting address in hex: C000
Ending address in hex: CD43
SYS to start: 49152

***Flankspeed* required for entry! See page 106.**

```
C000: 20 6E C9 20 93 C6 A2 02 77
C008: BD B3 C9 95 7C CA 10 F8 29
C010: A5 BA C9 08 B0 02 A9 08 A6
C018: 8D C7 CC 09 30 8D C6 C9 91
C020: A9 00 8D BA CC 4C 47 C1 34
C028: 8D BC CC 8E BB CC BA BD CE
C030: 01 01 C9 8C D0 18 BD 02 31
C038: 01 C9 A4 D0 11 A2 00 86 B2
C040: 90 AD BC CC DD A7 C9 F0 48
C048: 16 E8 E0 06 90 F6 AD BC 20
C050: CC 18 AE BB CC C9 3A B0 21
C058: 03 4C 80 00 4C 8A 00 8D 8C
C060: BE CC 8E B7 CC AD B6 C9 2D
C068: 85 BA 8D C7 CC 20 72 C5 23
C070: AE B7 CC 8A 0A AA BD 12 B2
C078: CA 48 BD 11 CA 48 60 A9 77
C080: A7 A0 CC 20 65 C6 A9 C9 55
C088: A0 CC 20 65 C6 A2 50 BD F2
C090: C9 CC 9D CB CC CA 10 F7 30
C098: 20 9F C6 4C 47 C1 AD 01 23
C0A0: 02 C9 24 D0 03 4C 1B C2 8E
C0A8: C9 54 D0 06 20 93 C6 4C 64
C0B0: 47 C1 C9 50 D0 1B AD 02 6F
C0B8: 02 C9 34 F0 0D C9 35 F0 A6
C0C0: 09 A9 B1 A0 CB 20 65 C6 DD
C0C8: F0 E5 25 0F 8D BA CC D0 B9
C0D0: DE C9 44 D0 1C AD 02 02 5C
C0D8: C9 38 F0 0F C9 39 F0 0B D9
C0E0: A9 DA A0 CB 20 65 C6 F0 0F
C0E8: C6 D0 C4 8D C7 CC 4C 18 CB
C0F0: C0 C9 51 D0 03 4C 4C C6 FF
C0F8: C9 3F D0 2F A9 5A A0 CC 73
C100: 20 65 C6 F0 AA 20 65 C6 34
C108: F0 3D AD C7 CC 85 BA 20 D8
C110: B1 FF A9 F0 20 B9 ED 24 48
C118: 90 10 0D 20 AE FF A9 BF FD
C120: A0 C9 20 65 C6 4C B6 C5 9F
C128: 4C AE FF A5 BA 20 B1 FF 55
C130: A9 6F 85 B9 20 93 FF A0 DC
C138: 00 B9 01 02 F0 06 20 A8 B4
C140: FF C8 10 F5 20 AE FF 20 FD
C148: 0A C1 A9 BF A0 C9 20 65 6D
C150: C6 AD C7 CC 20 B4 FF A9 D7
C158: 6F 85 B9 20 96 FF A9 00 67
C160: 8D B4 CC 8D B5 CC 20 A5 45
C168: FF AA EE B4 CC AD B4 CC B2
C170: C9 02 D0 03 8E B5 CC 8A AB
C178: 20 D2 FF C9 0D D0 E7 20 1B
C180: AB FF AD B3 CC F0 01 60 AC
C188: 4C 4F C6 A2 00 8E BD CC A6
C190: 20 69 C3 20 98 C8 A2 07 09
C198: BD 74 CA 8E B7 CC AA 8E E1
C1A0: BB CC 20 C8 C1 B0 0C AD 3E
C1A8: BB CC AE B7 CC 20 D9 C1 20
C1B0: 90 15 18 AE B7 CC CA 10 7C
C1B8: DF AD 00 CE D0 D2 A9 66 C8
C1C0: A0 CB 20 65 C6 F0 80 60 4B
C1C8: BD 00 CE C9 05 90 08 29 E5
C1D0: 0F C9 00 F0 02 18 60 38 4D
C1D8: 60 8E B7 CC AA 8E BB CC 0E
C1E0: 20 00 C2 CC C8 CC D0 16 0D
C1E8: AE BB CC A0 00 BD 03 CE 50
C1F0: D9 CB CC D0 09 E8 C8 CC BB
C1F8: C8 CC D0 F1 18 60 38 60 62
C200: A0 00 BD 03 CE F0 08 C9 F2
C208: A0 F0 04 E8 C8 D0 F3 60 74
C210: A2 26 A9 2D 20 D2 FF CA 6D
C218: 10 F8 60 A2 00 8E BD CC 3D
C220: BD 1D CA 30 06 9D 02 CA 66
C228: E8 D0 F5 A2 00 BD 03 02 3D
C230: F0 03 E8 D0 F8 8E CA C9 F9
C238: A9 60 8D 7B C9 20 98 C8 96
C240: AD 02 CE C9 41 F0 04 C9 88
C248: 42 D0 75 AD BA CC F0 03 F9
C250: 20 75 C9 20 89 C6 20 10 50
C258: C2 A9 B7 A0 C9 20 65 C6 33
C260: A2 90 BD 00 CE 29 7F 20 E8
C268: D2 FF E8 E0 A2 D0 F3 A9 16
C270: 2C 20 D2 FF BD 00 CE 20 3C
C278: D2 FF BD 01 CE 20 D2 FF CB
C280: AD A5 CE 8D 48 CA AD A6 97
C288: CE 8D 49 CA A9 3E A0 CA 4C
C290: 20 65 C6 A9 60 A0 CA 20 72
C298: 65 C6 20 10 C2 20 89 C6 28
C2A0: 20 69 C3 20 98 C8 20 96 26
C2A8: C3 AD 00 CE D0 F2 A2 01 50
C2B0: 8E 3D CA CA 8E C5 CC 20 53
C2B8: 89 C6 20 CC FF 20 D0 C2 A9
C2C0: A9 00 8D C5 CC 8D BA CC 9F
C2C8: A9 09 8D 13 C9 4C 47 C1 3B
C2D0: A9 AE 85 BB A9 C9 85 BC 20
C2D8: A9 05 85 B7 AD C7 CC 85 8C
C2E0: BA A9 60 85 B9 20 D5 F3 CE
C2E8: A5 BA 20 B4 FF A5 B9 20 9D
C2F0: 96 FF A9 00 85 90 A0 22 0A
C2F8: 20 A5 FF 88 10 FA 48 20 BA
C300: A5 FF 8D D4 CC 68 8D D3 9E
C308: CC A2 00 20 A5 FF F0 06 34
C310: 9D D5 CC E8 D0 F5 20 3F 5F
C318: C3 AD BA CC F0 03 20 75 9A
C320: C9 AD D4 CC AE D3 CC 20 A8
C328: 0C C9 A9 20 20 D2 FF A2 5D
```

```
C330: 00 BD D5 CC 20 D2 FF E8 6C
C338: E0 0C 90 F5 20 89 C6 20 3C
C340: 42 F6 4C E7 FF 20 E1 FF AF
C348: F0 10 20 E4 FF F0 19 20 78
C350: E4 FF F0 FB C9 20 F0 F7 F4
C358: D0 EB 20 E7 FF 4C C0 C2 EC
C360: A9 20 20 D2 FF CA D0 F8 B1
C368: 60 A2 02 BD 1D CA 9D 02 B2
C370: CA CA 10 F7 AE BD CC A0 E7
C378: 00 BD 22 CA 29 7F 99 05 6A
C380: CA C8 E8 BD 22 CA 30 05 DC
C388: 99 05 CA D0 F4 A9 0D 99 08
C390: 05 CA 8E BD CC 60 A0 07 81
C398: B9 74 CA 8C C6 CC AA 20 7C
C3A0: C6 C3 A9 00 8D C5 CC 20 15
C3A8: 45 C3 AC C6 CC 88 10 E8 73
C3B0: 60 48 AD C5 CC D0 0D A9 21
C3B8: 20 20 D2 FF A9 22 8D C5 EA
C3C0: CC 20 D2 FF 68 60 8E C0 98
C3C8: CC 20 54 C4 B0 27 AE C0 16
C3D0: CC BD 00 CE F0 EF A0 00 AB
C3D8: BD 03 CE F0 19 C9 A0 F0 CD
C3E0: 15 20 B1 C3 C8 C0 11 90 B6
C3E8: 03 88 D0 0A 20 D2 FF EE 31
C3F0: BF CC E8 D0 E3 60 AD BF E8
C3F8: CC F0 FA A9 22 20 D2 FF 70
C400: A9 00 8D BF CC 84 B7 8C 8C
C408: C8 CC A9 11 E5 B7 F0 04 EA
C410: AA 20 60 C3 AE C0 CC BD F8
C418: 00 CE 20 93 C4 AC C0 CC 99
C420: B9 1C CE AA B9 1D CE A0 B5
C428: 20 8C C4 CC A0 05 8C 13 AB
C430: C9 20 0C C9 A9 20 20 D2 AC
C438: FF A0 00 8C B8 CC 8C 3D B4
C440: CA A2 30 8E C4 CC 8D 3D C8
C448: CA 20 CD C6 A9 20 8D C4 E3
C450: CC 4C 89 C6 AD 02 02 C9 35
C458: 3A D0 19 A0 00 AE C0 CC 59
C460: B9 03 02 F0 23 C9 3F F0 2D
C468: 0D C9 2A F0 07 DD 03 CE 11
C470: F0 0D 38 60 18 60 BD 03 40
C478: CE F0 F7 C9 A0 F0 F3 E8 68
C480: C8 CC CA C9 90 DA F0 D8 DF
C488: BD 03 CE F0 E7 C9 A0 F0 4C
C490: E3 38 60 48 A8 F0 0C 29 24
C498: 80 F0 0B 98 29 40 F0 03 0B
C4A0: A9 3E 2C A9 20 2C A9 2A 7E
C4A8: 20 D2 FF 68 29 0F 18 F0 45
C4B0: 10 C9 01 F0 10 C9 02 F0 49
C4B8: 10 C9 03 F0 10 A9 A1 B0 92
C4C0: 0E A9 A6 90 0A A9 B0 B0 C4
C4C8: 06 A9 AB B0 02 A9 B5 A0 D6
C4D0: CA 4C 65 C6 20 0A C1 A6 A6
C4D8: 2B A4 2C 86 C3 84 C4 AD 16
C4E0: BE CC C9 25 D0 03 A9 01 D9
C4E8: 2C A9 00 85 B9 A9 00 20 C7
C4F0: D5 FF AD BE CC C9 25 F0 DF
C4F8: 3B 86 AE 84 AF A5 AF 85 78
C500: 2E A5 AE 85 2D 20 59 A6 55
C508: 20 33 A5 AD BE CC C9 2F 33
C510: F0 22 A9 01 8D B3 CC 20 FB
C518: 47 C1 A9 00 8D B3 CC AD 86
C520: B5 CC 29 0F D0 11 A9 00 66
C528: 20 90 FF 20 3B C6 20 8E A9
C530: A6 4C AE A7 4C 47 C1 4C 1B
C538: 4F C6 20 0A C1 AD C9 CC 7E
C540: C9 30 F0 0A A9 0D A0 CC 59
C548: 20 65 C6 4C 47 C1 20 59 63
C550: E1 A5 90 F0 07 C9 40 F0 5B
C558: 03 4C 68 C5 CE C8 CC CE 09
C560: C8 CC 20 9F C6 4C 47 C1 D1
C568: A9 94 A0 C9 20 65 C6 4C A9
C570: 4A C1 A0 00 A2 01 BD 00 7E
C578: 02 F0 28 C9 3A D0 0B E8 5C
C580: BD 00 02 D0 05 CA 8A A8 14
C588: D0 19 C9 22 F0 05 9D C8 BA
C590: CC D0 EC E8 BD 00 02 F0 B4
C598: 0A C9 22 F0 06 99 C9 CC B5
C5A0: C8 D0 F0 84 B7 8C C8 CC 89
C5A8: A9 C9 85 BB A9 CC 85 BC 16
C5B0: 60 A9 0D 4C A8 FF 20 E7 C4
C5B8: FF A9 7C A0 CA 4C 05 C1 5D
C5C0: A9 0F AE C7 CC A0 6F 20 EC
C5C8: BA FF A9 00 20 BD FF 20 2B
C5D0: C0 FF AD C7 CC 20 B1 FF A5
C5D8: A5 B9 20 93 FF A9 02 AE 46
C5E0: C7 CC A0 02 20 BA FF A9 9C
C5E8: 01 A2 AD A0 C9 20 BD FF 82
C5F0: 20 C0 FF A2 0F 20 C9 FF 6D
C5F8: A2 07 BD 09 CA 20 CA F1 11
C600: CA 10 F7 20 CC FF A2 0F 71
C608: 20 C9 FF A9 32 8D FC C9 22
C610: A2 00 BD FB C9 30 06 20 8C
C618: CA F1 E8 D0 F5 A2 02 20 49
C620: C9 FF A2 00 BD 00 CF 20 3A
C628: CA F1 E8 D0 F7 20 CC FF 83
C630: A9 02 20 C3 FF 20 00 C9 A9
C638: 4C 4A C1 A2 02 BD AB E3 82
C640: 95 7C CA 10 F8 A9 A1 A0 12
C648: CC 4C 65 C6 20 3B C6 A9 59
C650: 00 8D B6 CC 9D 00 02 9D 9E
C658: C9 CC E8 E0 59 90 F5 20 B8
C660: 59 A6 4C 86 E3 84 15 85 36
C668: 14 A0 00 B1 14 F0 21 C9 BE
C670: 8D F0 18 C9 C4 F0 14 C9 64
C678: D9 F0 10 C9 D4 F0 0C C9 B8
C680: FF F0 0D 20 D2 FF C8 D0 0B
C688: E2 A9 0D 29 7F 20 D2 FF BD
C690: A9 00 60 20 35 C7 20 0C E3
C698: C8 20 B1 C7 4C 4D C8 20 7D
C6A0: 8B C1 AE BB CC AD 0B DC BA
C6A8: 20 C9 C6 E8 AD 0A DC 20 F6
C6B0: C9 C6 E8 A0 00 B9 D0 C9 1F
C6B8: 20 C9 C6 C8 E8 B9 D0 C9 6F
C6C0: 20 C9 C6 AD 08 DC 4C C0 11
C6C8: C5 9D 16 CF 60 AC C0 CC AC
```

```
C6D0: B9 16 CE F0 27 10 04 18 B3
C6D8: 20 91 C8 A2 3A 20 03 C7 1B
C6E0: C8 A2 20 B9 16 CE 20 03 2E
C6E8: C7 A2 2F C8 B9 16 CE 20 0A
C6F0: 03 C7 C8 A2 00 B9 16 CE C5
C6F8: 20 03 C7 60 A9 01 A0 CC 5C
C700: 4C 65 C6 48 20 10 C7 68 21
C708: 20 14 C7 8A F0 ED D0 08 46
C710: 4A 4A 4A 4A 29 0F 09 30 AA
C718: 4C D2 FF AD 0B DC 10 04 E0
C720: 18 20 91 C8 A2 3A 20 03 B2
C728: C7 A2 20 AD 0A DC 20 03 6A
C730: C7 AD 08 DC 60 A9 D6 A0 0C
C738: C9 20 65 C6 20 1B C7 A0 F1
C740: CA A9 BA 20 65 C6 20 CF AB
C748: FF C9 0D D0 03 68 68 60 24
C750: C9 41 F0 06 C9 50 F0 05 62
C758: D0 DB A9 00 2C A9 80 8D 92
C760: CF C9 A0 CA A9 D5 20 65 6A
C768: C6 20 CF FF F0 FB C9 30 06
C770: 90 F0 C9 33 B0 EC 8D CB E5
C778: C9 20 CF FF F0 FB C9 30 19
C780: 90 E0 C9 3A B0 DC 8D CC DD
C788: C9 A0 CA A9 EF 20 65 C6 A3
C790: 20 CF FF F0 FB C9 30 90 F7
C798: F0 C9 36 B0 EC 8D CD C9 4C
C7A0: 20 CF FF F0 FB C9 30 90 08
C7A8: E0 C9 3A B0 DC 8D CE C9 41
C7B0: 60 A0 CB A9 09 20 65 C6 7C
C7B8: 20 CF FF F0 FB C9 30 90 20
C7C0: 4A C9 32 B0 46 8D D2 C9 28
C7C8: 8D E1 C9 20 CF FF F0 FB DE
C7D0: C9 30 90 DD C9 33 B0 D9 C0
C7D8: 8D D3 C9 8D E2 C9 A0 CB AA
C7E0: A9 23 20 65 C6 20 CF FF E9
C7E8: F0 FB C9 30 90 F0 C9 34 4F
C7F0: B0 EC 8D D4 C9 8D E4 C9 F6
C7F8: 20 CF FF F0 FB C9 30 90 60
C800: DD C9 3A B0 D9 8D D5 C9 99
C808: 8D E5 C9 60 18 AD CB C9 01
C810: 20 8D C8 18 20 87 C8 8D 9C
C818: BE CC AD CC C9 20 8D C8 5E
C820: 18 6D BE CC 6D CF C9 8D C5
C828: 0B DC AD CD C9 20 8D C8 CB
C830: 18 20 87 C8 8D BE CC AD 7F
C838: CE C9 20 8D C8 18 6D BE 8B
C840: CC 8D 0A DC A9 00 8D 09 C1
C848: DC 8D 08 DC 60 18 AD D2 90
C850: C9 20 8D C8 18 20 87 C8 19
C858: 8D BE CC AD D3 C9 20 8D 6A
C860: C8 18 6D BE CC 8D D0 C9 62
C868: AD D4 C9 20 8D C8 18 20 63
C870: 87 C8 8D BE CC AD D5 C9 27
C878: 20 8D C8 18 6D BE CC 8D 8D
C880: D1 C9 A9 93 4C D2 FF 18 90
C888: 0A 0A 0A 0A 60 38 E9 30 63
C890: 60 29 7F F8 69 12 D8 60 47
C898: 20 E7 FF 20 0A C1 A9 0F 45
C8A0: AE C7 CC A8 20 BA FF A9 11
C8A8: 00 20 BD FF 20 C0 FF A9 11
C8B0: 02 AE C7 CC A8 20 BA FF 79
C8B8: A9 01 A2 AD A0 C9 20 BD FB
C8C0: FF 20 C0 FF A2 0F 20 C9 3D
C8C8: FF A9 31 8D FC C9 A2 00 9A
C8D0: BD FB C9 C9 0D F0 06 20 42
C8D8: CA F1 E8 D0 F3 20 CA F1 20
C8E0: 20 CC FF A2 02 20 C6 FF 59
C8E8: A2 00 20 A5 FF 9D 00 CE BD
C8F0: 9D 00 CF E8 E0 00 D0 F2 EB
C8F8: 8E BC CC A9 02 20 C3 FF A0
C900: 20 E7 FF AD BA CC F0 03 31
C908: 4C 75 C9 60 8D C2 CC 8E 9F
C910: C1 CC A2 09 8E C3 CC A0 0A
C918: B0 AD C1 CC DD 9C C9 AD F6
C920: C2 CC FD 9D C9 90 0F 8D 42
C928: C2 CC AD C1 CC FD 9C C9 58
C930: 8D C1 CC C8 D0 E3 98 CA 2D
C938: F0 11 C9 B0 F0 03 8D C3 F9
C940: CC 2C C3 CC 30 05 AD C4 71
C948: CC F0 1F 29 7F 8E B9 CC E2
C950: AE 3D CA D0 0F AE B8 CC 1B
C958: 9D 23 CD E8 8E B8 CC AE 92
C960: B9 CC 10 06 AE B9 CC 20 52
C968: D2 FF CA 10 AA 60 A0 CB 8D
C970: A9 3D 4C 65 C6 A9 82 AE AA
C978: BA CC A0 60 20 BA FF A9 85
C980: 00 20 BD FF 20 C0 FF A2 E1
C988: 82 20 C9 FF C9 05 D0 03 97
C990: 4C B6 C5 60 41 42 4F 52 DE
C998: 54 45 44 0D 00 01 00 0A 8E
C9A0: 00 64 00 E8 03 10 27 5E 86
C9A8: 2F 25 40 5F 21 23 24 30 35
C9B0: 3A 3C 3E 4C 28 C0 08 0D AF
C9B8: 44 49 53 4B 3A 20 FF 0D 4C
C9C0: 44 49 53 4B 2D 28 38 29 A3
C9C8: 3A FF 00 30 33 30 36 00 CC
C9D0: 01 17 30 31 31 37 0D 54 14
C9D8: 4F 44 41 59 20 49 53 3A FD
C9E0: 20 30 31 2F 31 37 20 20 3A
C9E8: 20 20 20 20 43 55 52 52 A6
C9F0: 45 4E 54 20 54 49 4D 45 29
C9F8: 3A 20 00 55 31 3A 32 20 66
CA00: 30 20 31 38 20 31 31 0D 49
CA08: FF 0D 30 20 32 3A 50 2D 4F
CA10: 42 D3 C4 D3 C4 D3 C4 9D B9
CA18: C0 39 C5 7E C0 31 38 20 A0
CA20: 30 0D B1 B4 B7 B1 30 B1 0F
CA28: 33 B1 36 B2 B5 B8 B1 31 47
CA30: B1 34 B1 37 B3 B6 B9 B1 D4
CA38: 32 B1 35 B1 38 01 20 44 A0
CA40: 4F 53 20 56 45 52 2E 3D 5C
CA48: 20 20 0D 46 49 4C 45 4E 05
CA50: 41 4D 45 20 20 20 20 20 C4
CA58: 20 20 20 20 20 20 20 00 39
CA60: 54 59 50 20 42 4C 4B 20 78
CA68: 4C 41 53 54 20 55 50 44 A7
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 105 and 106 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
CA70: 41 54 45 8D E2 C2 A2 82 A3
CA78: 62 42 22 02 0D 44 45 56 2E
CA80: 49 43 45 20 49 53 20 4F 7E
CA88: 46 46 4C 49 4E 45 20 4F AD
CA90: 52 20 49 53 20 4E 4F 54 B1
CA98: 20 50 52 45 53 45 4E 54 DB
CAA0: 00 52 45 4C 20 FF 44 45 2E
CAA8: 4C 20 FF 50 52 47 20 FF 1F
CAB0: 53 45 51 20 FF 55 53 52 B5
CAB8: 20 FF 0D 0D 41 4D 20 4F F0
CAC0: 52 20 50 4D 20 20 20 20 51
CAC8: 20 20 20 28 41 2F 50 2F 41
CAD0: 43 52 29 20 00 0D 43 55 55
CAD8: 52 52 45 4E 54 20 48 4F 1D
CAE0: 55 52 53 20 20 20 28 30 94
CAE8: 31 2D 31 32 29 20 00 0D 01
CAF0: 43 55 52 52 45 4E 54 20 36
CAF8: 4D 49 4E 55 54 45 53 20 40
CB00: 28 30 30 2D 35 39 29 20 6D
CB08: 00 0D 43 55 52 52 45 4E E5
CB10: 54 20 4D 4F 4E 54 48 20 2C
CB18: 20 20 28 30 31 2D 31 32 72
CB20: 29 20 00 0D 43 55 52 52 B3
CB28: 45 4E 54 20 44 41 59 20 2F
CB30: 20 20 20 20 28 30 31 2D 67
CB38: 33 31 29 20 00 0D 20 20 33
CB40: 2D 2A 2D 20 41 48 4F 59 17
CB48: 21 27 53 20 43 48 52 4F 31
CB50: 4E 4F 2D 57 45 44 47 45 88
CB58: 20 56 45 52 20 31 2E 30 16
CB60: 20 2D 2A 2D 0D 00 93 53 F8
CB68: 59 53 54 45 4D 20 45 52 B3
CB70: 52 4F 52 3A 20 44 41 54 98
CB78: 45 2D 53 54 41 4D 50 45 B6
CB80: 52 20 43 41 4E 4E 4F 54 B7
CB88: 20 0D 4C 4F 43 41 54 45 6F
CB90: 20 46 49 4C 45 2C 20 50 6E
CB98: 4C 45 41 53 45 20 43 48 AF
CBA0: 45 43 4B 20 44 49 52 45 B9
CBA8: 43 54 4F 52 59 2E 0D 0D 83
CBB0: 00 0D 50 52 49 4E 54 45 91
CBB8: 52 20 4D 55 53 54 20 42 D7
CBC0: 45 20 34 20 4F 52 20 35 71
CBC8: 2C 20 50 4C 45 41 53 45 D0
CBD0: 20 52 45 2D 4B 45 59 2E CD
CBD8: 0D 00 0D 44 52 49 56 45 6E
CBE0: 20 4D 55 53 54 20 42 45 F2
CBE8: 20 38 20 4F 52 20 39 2C 88
CBF0: 20 50 4C 45 41 53 45 20 EC
CBF8: 52 45 2D 4B 45 59 2E 0D E2
CC00: 00 2D 2D 3A 2D 2D 20 2D 3C
CC08: 2D 2F 2D 2D 00 0D 20 20 0C
CC10: 20 20 20 46 49 4C 45 4E DF
CC18: 41 4D 45 20 4D 55 53 54 56
CC20: 20 49 4E 43 4C 55 44 45 46
CC28: 20 22 30 3A 22 20 46 4F AC
CC30: 52 0D 20 20 44 41 54 45 EE
CC38: 20 53 54 41 4D 50 45 52 76
CC40: 20 54 4F 20 57 4F 52 4B 68
CC48: 2C 20 50 4C 45 41 53 45 50
CC50: 20 52 45 2D 4B 45 59 2E 4D
CC58: 0D 00 0D 20 20 43 48 52 90
CC60: 4F 4E 4F 2D 57 45 44 47 A2
CC68: 45 20 56 31 2E 30 20 28 FB
CC70: 43 29 20 43 4F 50 59 52 8B
CC78: 49 47 48 54 20 31 39 38 68
CC80: 35 0D 0D 20 20 20 20 20 70
CC88: 20 20 20 20 20 20 20 44 AD
CC90: 45 4E 4E 59 20 4D 55 53 E1
CC98: 43 41 54 45 4C 4C 49 0D A5
CCA0: 00 0D 42 59 45 0D 00 12 AD
CCA8: 53 54 41 4D 50 49 4E 47 0E
CCB0: 3A 20 00 00 0D 30 00 00 48
CCB8: 00 00 00 00 3A 00 10 00 03
CCC0: 00 00 00 00 00 00 00 08 C8
CCC8: 00 00 00 00 00 00 00 00 C8
CCD0: 00 00 00 00 00 00 00 00 D0
CCD8: 00 00 00 00 00 00 00 00 D8
CCE0: 00 00 00 00 00 00 00 00 E0
CCE8: 00 00 00 00 00 00 00 00 E8
CCF0: 00 00 00 00 00 00 00 00 F0
CCF8: 00 00 00 00 00 00 00 00 F8
CD00: 00 00 00 00 00 00 00 00 00
CD08: 00 00 00 00 00 00 00 00 08
CD10: 00 00 00 00 00 00 00 00 10
CD18: 00 00 00 00 00 00 00 00 18
CD20: 00 00 00 00 00 00 00 00 20
CD28: 00 00 00 00 00 00 00 00 28
CD30: 00 00 00 00 00 00 00 FF 30
CD38: FF FF FF FF FF FF FF FF 38
CD40: 00 00 00 00 40
```

NOTEMAKER
FROM PAGE 51

```
·10 REM DISKETTE NOTEMAKER                                  KI
·15 REM FOR AHOY! MAGAZINE                                  DM
·20 REM BY TIM GERCHMEZ                                     GN
·25 POKE53281,0:POKE53280,15:POKE646,1                      BF
·30 PRINT"[CLEAR][DOWN][8" "]DISKETTE NOT
 E CREATOR[DOWN][DOWN]"                                     NO
·35 PRINT"YOU CAN USE THE WHOLE SCREEN AS
  A "                                                       LG
·40 PRINT"SCRIBBLING PAD. (EXCEPT THE TOP
 MOST "                                                     IN
·45 PRINT"AND BOTTOMMOST LINE)."                            KG
·50 PRINT:PRINT"PRESS RETURN WHEN DONE."                    PP
·55 PRINT"PRESS SHIFT-RETURN IF YOU WANT
 TO"                                                        DI
·60 PRINT"ADVANCE THE CURSOR TO THE NEXT
 LINE."                                                     OG
```

```
·62 PRINT:PRINT"THE CURSOR COLOR IN USE W
 HEN RETURN IS"                          DG
·63 PRINT"PRESSED WILL BE THE ONE USED WI
 TH THE":PRINT"NOTEFILE."                MJ
·64 PRINT:PRINT"YOU LOAD THE NOTEFILE WIT
 H A ,8,1"                               DM
·65 PRINT:INPUT"FILENAME";FI$:FI$=FI$+".N
 ":PRINT:PRINT"PRESS RETURN TO START."   HO
·70 POKE198,0                            KB
·75 GETA$:IFA$=""THEN75                  GH
·80 PRINT"[CLEAR]";:SYS65487             GO
·85 OPEN15,8,15,"I0":PRINT#15,"S0:"FI$   MM
·90 CLOSE15:OPEN2,8,2,FI$+",P,W"         JD
·95 PRINT#2,CHR$(0)CHR$(1);              OF
·100 FORT=256TO514:CT=T:PRINT#2,CHR$(2);:
 NEXT                                    LE
·105 READA:IFA=256THEN115                HJ
·110 CT=CT+1:PRINT#2,CHR$(A);:GOTO105    HM
·115 CT=CT+1:IFCT<2040THENPRINT#2,CHR$(PE
 EK(CT));:GOTO115                        JK
·120 CLOSE2                              NC
·125 PRINT"[CLEAR][DOWN][DOWN]DONE."     PD
·130 PRINT:PRINT"ANOTHER FILE (Y/N)":POKE
 198,0                                   LB
·135 GETA$:IFA$=""THEN135                HO
·140 IFA$="Y"THENRUN                     IK
·145 IFA$="N"THENSYS2048                 FF
·150 GOTO 135                            CH
·1000 DATA160,0,173,134,2,153,0,216,153,0
 ,217,153,0,218,153,0,219                PM
·1005 DATA200,208,241,169,19,32,210,255,7
 6,116,164,256                           LB
```

FLANKSPEED LISTING CREATOR
FROM PAGE 51

```
·5 POKE53280,12:POKE53281,11             LL
·6 PRINT"[CLEAR][c 8][RVSON][7" "]FLANKSP
 EED LISTING CREATOR[7" "]";             JD
·10 PRINT"[RVSON] CONVERTS MEMORY TO FLAN
 KSPEED LISTING  "                       CN
·15 PRINT"[9" "]CREATED BY G. F. WHEAT[9"
  "]"                                    AD
·20 PRINT"[RVSON][3" "]COPR. 1986, ION IN
 TERNATIONAL INC.[3" "]"                 CJ
·40 PRINT:INPUT"PROGRAM NAME";C$         GH
·50 PRINT:INPUT"SYS TO START IN DECIMAL";
 D$                                      KI
·80 B$="STARTING BYTE OF ML IN HEX":GOSUB
 2010:AD=B                               IK
·85 GOSUB2520:IFB=0THEN80                FO
·90 B$="LAST BYTE OF ML IN HEX":GOSUB2010
 :EN=B                                   EM
·95 GOSUB2510:IFB=0THEN80                FP
·100 OPEN4,4:PRINT#4,"PROG NAME: ";C$:PRI
 NT#4                                    HF
·110 AC=AD:GOSUB3010:PRINT#4,"STARTING AD
 DRESS: "B$:PRINT#4                      IB
·120 AC=EN:GOSUB3010:PRINT#4,"ENDING ADDR
 ESS: "B$:PRINT#4                        PJ
·125 PRINT#4,"SYS TO START: "D$:PRINT#4   IA
·130 AC=AD:GOSUB3010:C$=B$+": ":CK=AD-(IN
 T(AD/256)*256)                          KC
·140 FORB=0TO7:AC=PEEK(AD+B):CK=CK+AC:IFC
 K>255THENCK=CK-255                      IN
·150 GOSUB3030:C$=C$+B$+" ":IFAD+B=ENTHEN
 500                                     HE
·160 NEXTB:AC=CK:GOSUB3030:C$=C$+B$      PI
·170 PRINT#4,C$:AD=AD+8:GOTO130          ME
·500 AC=CK:GOSUB3030:C$=C$+B$:PRINT#4,C$ PC
·600 CLOSE4:END                          JD
·2000 REM GET FOUR DIGIT HEX             PC
·2010 PRINT:PRINTB$;:INPUTT$             GM
·2020 IFLEN(T$)<>4THEN2010               EB
·2040 FORA=1TO4:A$=MID$(T$,A,1):GOSUB2060
 :IFT(A)=16THEN2010                      KH
·2050 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)
 *16)+T(4):RETURN                        GF
·2060 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-5
 5:RETURN                                EH
·2070 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-4
 8:RETURN                                KP
·2080 T(A)=16:RETURN                     NP
·2500 REM ADRESS CHECK                   LI
·2510 IFAD>ENTHENB=0:RETURN              PK
·2520 IFB<2560R(B>40960ANDB<49152)ORB>532
 47THENB=0                               AP
·2530 RETURN                             IM
·3000 REM ADDRESS TO HEX                 EB
·3010 B$="":A=4096:GOSUB3070             PJ
·3020 A=256:GOSUB3070:GOTO3035           NB
·3030 B$=""                              FO
·3035 A=16:GOSUB3070                     PN
·3040 A=1:GOTO3070                       LN
·3060 RETURN                             IM
·3070 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):
 GOTO3090                                CJ
·3080 A$=CHR$(T+48)                      JP
·3090 B$=B$+A$:AC=AC-A*T:RETURN          EG
```

SCUTTLEBUTT

Continued from page 96

in memory, uses every C-128 key, and includes a function key-accessible page up/page down feature. A video display of 80, 160, or 320 columns can be selected.

HomePak ($49.95) now includes punter protocol for communication with Commodore bulletin boards.

The Consultant ($59.95) for the 128 utilizes the 128's extra keys to perform certain functions, and can be automatically booted on power-up.

Batteries Included, 416-881-9941 (see address list, this page).

TAX/MONEY MANAGERS

The 1986 edition of *J.K. Lasser's Your Income Tax* ($69.95) for the C-64 includes 28 forms and schedules, an "interactive interview" that selects the proper ones for the user, and a copy of the guidebook of the same name. Included is a worksheet feature for performing computations, taking notes, and listing information.

The program allows individuals to move back and forth between 1040 line numbers and related forms and schedules. All screen-displayed forms and schedules match the official IRS forms; the program also prints out IRS-accepted forms and schedules.

J.K. Lasser continues nosing into your personal financial affairs with *J.K. Lasser's Your Money Manager* ($69.95), a home accounting/small business tool providing a check writer, financial statements, balance sheets, budget reports, and other essentials for financial planning.

Simon & Schuster Computer Software, 212-245-6400 (see address list, this page).

COMPANIES MENTIONED IN SCUTTLEBUTT

AC3L Software
P.O. Box 7
New Derry, PA 15671

Accolade Inc.
20863 Stevens Creek Boulevard
Cupertino, CA 95014
Phone: 408-446-5757

Access Software
2561 South 1560 West
Woods Cross, UT 84087
Phone: 801-298-9077

American People/Link
3215 North Frontage Road, Suite 1505
Arlington Heights, IL 60004
Phone: 800-524-0100; in IL 312-870-5200

Batteries Included
30 Mural Street
Richmond Hill, Ontario
L4B 1B5 Canada
Phone: 416-881-9941

Berkeley Software
2150 Shattuck Avenue
Berkeley, CA 94704
Phone: 415-644-0883

Brown Boxes, Inc.
26 Concord Road
Bedford, MA 01730
Phone: 617-275-0090

Cardco, Inc.
300 S. Topeka
Wichita, KS 67202

Covox Inc.
675-D Conger Street
Eugene, OR 97402
Phone: 503-342-1271

Epyx, Inc.
1043 Kiel Court
Sunnyvale, CA 94089
Phone: 408-745-0700

Ergo Systems, Inc.
303-3 Convention Way
Redwood City, CA 94063
Phone: 415-363-5966

Fuji Photo Film U.S.A., Inc.
350 Fifth Avenue
New York, NY 10118
Phone: 212-736-3335

IHT Software
2269 Chestnut Street, Suite 162
San Francisco, CA 94123
Phone: 415-441-1607

Learning Technologies
4255 LBJ, Suite 265
Dallas, TX 75244
Phone: 214-385-2351

Maxtron
1825A Durfee Avenue
South El Monte, CA 91733
Phone: 818-350-5707

Microcomscribe
8982 Stimson Ct.
San Diego, CA 92129

MicroProse Simulation Software
120 Lakefront Drive
Hunt Valley, MD 21030
Phone: 301-667-1151

Multibotics, Inc.
2561 South 1560 West
Woods Cross, UT 84087
Phone: 801-298-9079

Nth Digit Solutions
3243 Arlington Avenue, No. 195
Riverside, CA 92506

Progressive Peripherals and Software
464 Kalamath Street
Denver, CO 80204
Phone: 303-825-4144

Scott, Foresman and Company
1900 East Lake Avenue
Glenview, IL 60025
Phone: 312-729-3000

ShareData, Inc.
7122 Shady Oak Road
Eden Prairie, MN 55344
Phone: 800-328-6061 or 612-829-0409

Simon & Schuster
1230 Avenue of the Americas
New York, NY 10020
Phone: 212-245-6400

Smoky Mountain Software
P.O. Box 1710
Brevard, NC 28712
Phone: 704-885-2516

Springboard Software, Inc.
7808 Creekridge Circle
Minneapolis, MN 55435

SubLOGIC Corporation
713 Edgebrook Drive
Champaign, IL 61820
Phone: 217-359-8482; orders 800-637-4983 except in IL, AK, HI

Xetec, Inc.
2804 Arnold Rd.
Salina, KS 67401
Phone: 913-827-0685

...COMING IN THE MAY ISSUE OF *AHOY!* (ON SALE APRIL 8)...

GOING FOR THE GOLD
STRATEGY GUIDE TO EPYX'S OLYMPIC SERIES

THE GREAT ESCAPE
USING ESCAPE CODES ON THE C-128

THE MUSICAL COMMODORE
ASSEMBLY LANGUAGE SOUND TECHNIQUES

THERE'S NO BEATING MASTERTRONIC

NEW

A superb sports simulation program with stunning 3-D graphics. A one or two player game that enables you to either compete head to head or against the computer.

Features include a self demo mode for instructions, a giant video screen for close-ups of the action in detail. Exciting to play with real baseball features, yet a detailed knowledge of the game is not essential: Steal Options, Curved balls, Pitchers Cheers leaders are all part of the incredible features that make this game an essential part of every collection.

C64/128
DISC

NEW

Join in the fastest moving game in the world, when you play 5-a-side soccer, an action packed sports simulation from Mastertronic.

Each game lasts ten minutes, with full crowd atmosphere and an announcer to keep you up to date with the action on the field. The tackles bite hard and the players twist, turn and shout as the crowd roar their approval.

You'll know you've been in a game when you play Mastertronic 5-a-side soccer!!

FEATURES INCLUDE:–
3D animated graphics one or two player game.
Incredible voice synthesis.

C64/128
DISC

THE MASTERTRONIC PLEDGE
We promise to bring you two outstanding added dimension programs every month.

BE A 'SPORTS HERO' WITH MASTERTRONIC

These superb sports simulations make you part of the team in two of the most action packed games to hit your **Commodore C64/128** screens.

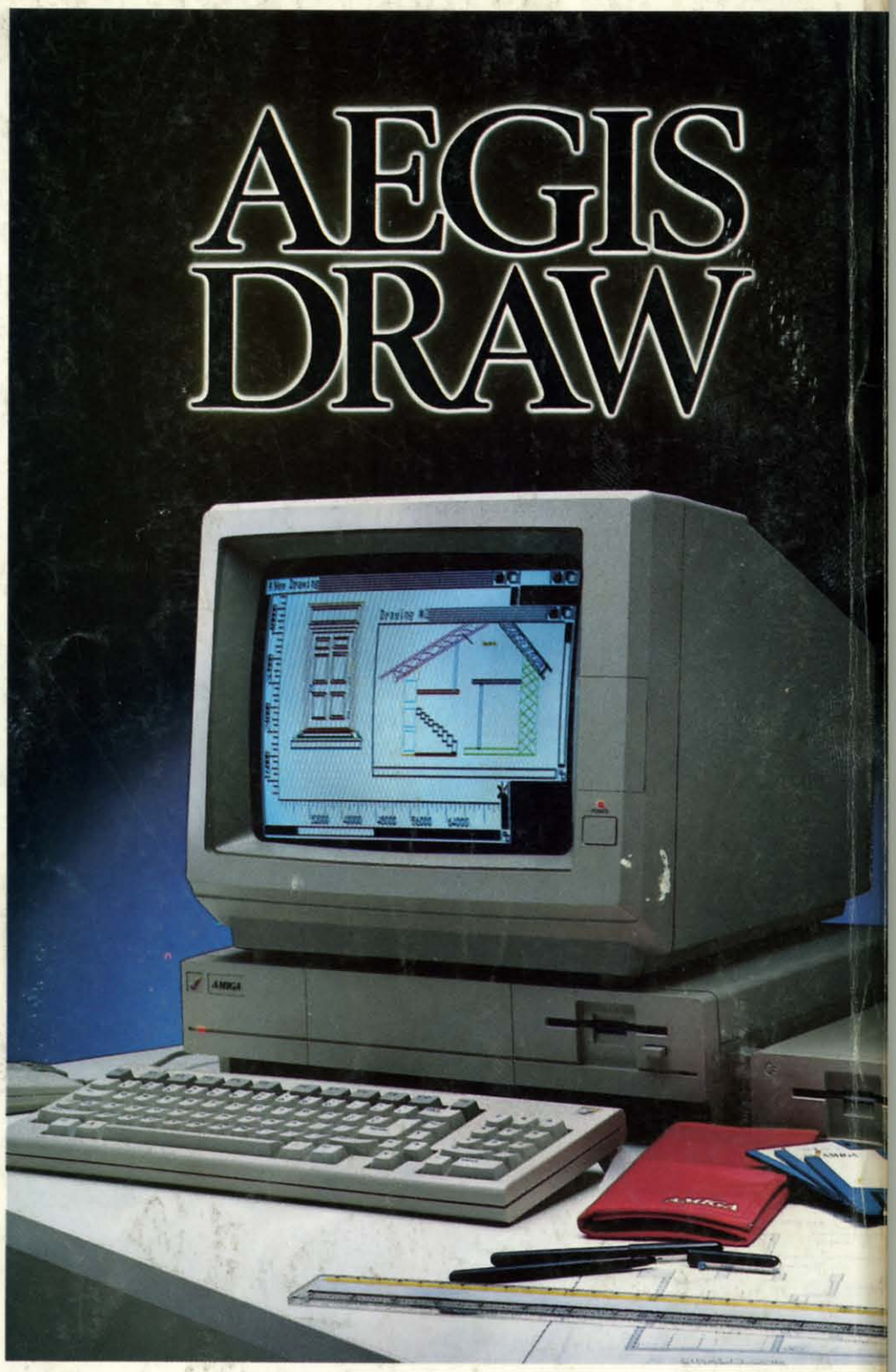
Astounding value at **$9.99** each, these programs will become a star part of your collection!

Reader Service No. 287

Mastertronic International Inc. 7311B Grove Road, Frederick, Maryland 21701 Tel: (301) 695 8877

AEGIS DRAW

A Drafting and CAD Tool for the Amiga™



Aegis Development, Inc. brings creativity to your fingertips! Use *Aegis Draw*™ to create accurate and detailed drawings of anything your mind can imagine and then transfer those images to plotters, printers, and other output devices. *Aegis Draw* was designed specifically for the Amiga and takes advantage of all the unique and powerful graphics capabilities that make this computer so special. You may work on several drawings at the same time using different windows. You may zoom in on an image or open a new window to observe detail while keeping the overall view of the drawing. Accuracy for a drawing is almost unlimited with accuracy far greater than 2,000,000,000 points! Flexible? Sure! Mark an image and store it, delete it, scale it, rotate it, whatever! *Aegis Draw* puts you in charge!

Aegis Draw also supports layering of a drawing— you may break up a drawing into various components allowing all or selected pieces of the layers to appear. A house plan can be broken into electrical, plumbing, and structural layers. The layers can appear in different colors, overriding the colors of the individual graphic elements.

Mouse, Keyboard, or Tablet input with pull down menus is provided. *Aegis Draw* allows you to set the physical scale for the output device, and create scaled drawings for architecture, electrical or structured engineering, and related CAD documents. Plotting can occur in background mode allowing you to keep working on other drawings. Plotters from HP, Epson, Comrex, and others are supported.

Mistakes? Accidental deletion can be reversed using the UNDO function. Expand your creativity by passing your *Aegis Draw* image into a paint system such as *Aegis Images*™ to add flare and solid image fills. All Aegis products use the Amiga standard IFF file format for easy data file exchange.

So, if you are serious about your Amiga computer, don't you think you owe it to yourself to get the most out of it? With *Aegis Draw,* your investment can last a lifetime. See your dealer today and ask for a complete demonstration!

P.S. Don't let your friends use *Aegis Draw*—you'll never get your computer back if you do!

For the dealer nearest you, call 1-213-306-0735

Reader Service No. 131

Remember—Software piracy is a crime!

AEGIS DEVELOPMENT

2210 Wilshire Blvd., Suite 277
Santa Monica, CA 90403

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. ☐ masters 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 _____

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

1. ☐ yes 2. ☐ no

Name_____

Street_____

City_____ **State**_____ **Zip**_____

PLACE
STAMP
HERE

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