REVIEWS

THE MUSIC SHOP FOR MIDI

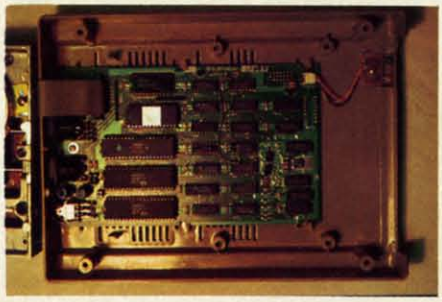
Passport Designs, Inc.
Commodore 64
Disk; $99.95

The Music Shop has proven to be one of the better music packages for the Commodore 64. Passport has taken this program one step further, adapting it for use with MIDI-equipped synthesizers. To use *The Music Shop for MIDI* you will need the Passport MIDI Interface Card, available separately. Additionally, the program will not work unless it is hooked up to a synthesizer. The MIDI version will use files created by the non-MIDI version.

The MIDI interface is about the size of a typical game cartridge, and plugs into the cartridge port. It has three short wires with DIN sockets on the ends. These are for MIDI-in, MIDI-out, and drum sync. The MIDI-in and out go in daisy chain fashion to any MIDI keyboards and drum machines you may have. The drum plug provides the clock signals necessary to sync any non-MIDI drum machines.

All features of the program are unchanged from the earlier version (see review last issue). The only changes are additional menus for the MIDI features, falling under the TOOLS MENU. The first feature is the MIDI ON/OFF function. When this is in the on position, notes may be entered onto the staff using your synthesizer. This is *not* real time entry of the music. If you currently have a quarter note selected for entry and you hold a G on the synthesizer, a quarter note G will appear on the staff. If you hold a chord, that chord will appear. To change to a different note you will have to resort to the computer keyboard. There is no provision within the program to do any real-time recording. The sole intention of this program, as of the original, is to allow you to enter music on a staff and then have the computer play it back. The MIDI enhancement only makes it sound better.

Useful, or necessary, for operation of* The Music Shop for MIDI: *64, monitor, printer, drive, joystick, keyboard. Not shown: MIDI Interface Card. READER SERVICE NO. 280



Motherboard of Enhancer 2000 drive. READER SERVICE NO. 281

The other MIDI menu is the SETUP MENU. This one facilitates the distribution of the music on up to four synthesizers. Depending on the staff that you are using, you may be able to use either two or four synthesizers. With one type of staff you have a separate staff for each synthesizer. With others you assign a note to a specific synthesizer by pointing its stem either up or down. There are eight presettable SETUP MIDI menus, allowing you to change your synthesizer/instrument definitions numerous times within a composition. Depending on what type of synthesizer you are using, you may not be able to take full advantage of the multiple instruments. Some synthesizers are only capable of producing one instrument at a time. I tested this program using a CASIO CZ-101. This is capable of producing four instruments at once, and worked flawlessly with *The Music Shop*.

The Music Shop with MIDI is an excellent adaptation of the earlier version. If your only interests are entering music on a staff and having your compositions played back in multiple instruments, this program will serve your needs well. If, on the other hand, you require real time entry features, you will not be able to use this package.

Passport Designs, Inc., 625 Miramontes St., Half Moon Bay, CA 94019 (phone: 415-726-0280).

—David Barron

ENHANCER 2000 DISK DRIVE

The Comtel Group, Inc.
Price: $219.95

The first thing you'll notice about the Enhancer 2000—especially if you're short of desktop space—is its small size. At 10 X 7.3 X 2.5", its total volume is just under half of the 1541. The main reason for the drive's compactness is its externally located power supply.

Regarding compatibility with the 1541, most of the software tested worked perfectly, with the exception of some recent releases (including *Beach-Head II* and *The Music Shop*). Also, the drive did not work with the Epyx *Fast Load* cartridge. The Comtel Group informed us that while they were aware that the production model tested for this review was not 100% compatible with the 1541, newer models would be.

www.commodore.ca

HAS EVERYTHING!

SUBSCRIBE TO AHOY!

☐ Twelve Issues for $19.95 ($26.95 Canada and elsewhere)
☐ Twenty-four Issues for $37.95 ($49.95 Canada and elsewhere)

Name_____

Address_____

City_____ State_____ Zip_____

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

www.commodore.ca

Formatting time is about 22 seconds. The aforementioned newer models, the Comtel Group assures us, will be faster by 30%.

The drive door is a hinge type reminiscent of the older 1541's. It seemed rather fragile, foreboding jamming problems after long periods of use. Three indicator lights are located on the front: a green power light and an amber error indicator below the disk insertion slot, and a red drive in use light above the slot. On the back are the dual serial bus, power socket, and power switch.

Inside, the drive has two circuit boards, with the main board located below the drive mechanism. Four of the chips on the main board have sockets, so if it does become necessary to replace them, it would be fairly simple to do so. The drive mechanism itself is a JVC direct drive motor.

I found changing the device number of the drive quite easy. The cover must first be removed to expose the main board. The jumper block is located on the rear left edge of the board. Using a pair of long nose pliers or tweezers, one or both or the jumpers can easily be removed to give you the device number needed, from 8 through 11. If later you decide to change the device number back to 8, just place the jumpers back accordingly.

A bonus disk included with the drive contained 12 programs. Besides the usual array of diagnostic programs that come with a new drive were several useful utilities, such as *Copy-All64* (disk backup utility), *Disk Log-Printer* (gives you a printout of your disk directory), *Disk Doctor* (allows you to edit tracks and sectors), and *Change Disk* (for changing the device number of the drive via software). Two games, a mortgage program, and a computer quiz were also included.

The manual is similar to the *1541 User's Manual,* but contains slightly less documentation. Explanations are given of all available DOS commands.

The manufacturer includes a limited one-year warranty, and for an additional $35.00 you can extend it to a second year. Considering the lifespan of most 1541-compatible drives, I would recommend spending the extra $35.00 for the extended warranty.

I found the Enhancer 2000 to be an efficient workhorse with a very quiet operation. After four hours of continuous functioning in 90 degree temperature, it performing flawlessly. Though the drive's list price is $215.00, it is expected to sell in the $169-$189 range.

The Comtel Group, 1651 East Edinger, Suite 209, Santa Ana, CA 92705 (phone: 714-953-6165).

—Michael Davila

VIEWTRON

Viewdata Corporation of America, Inc.

"Viewtron? What's Viewtron?" It seems that every time I mention this new videotex service the response is the same. If you haven't heard of it by now, grab hold of your socks while I tell you what you've been missing!

Videotex is a type of system that allows information to be transferred between the host and the user as full color graphics. Through the use of its own special software, Viewtron offers the user access to a wealth of online information with every single page in full color.

What's offered besides pretty pictures? How about reviews and ratings of more than 600 pieces of Commodore software and hardware, with new reviews added every two weeks? It's not a one way street, either. They want to know what *you* think about the items mentioned. Plus, nearly every item reviewed can be ordered while you are online from a major mail order firm, at a discount.

If being able to order while online whets your appetite for more, you'll love the next feature. Viewtron has a special electronic auction that is open to all users. Many major brands are "put on the block" every day. The prices keep dropping every 15 minutes until the items are sold. It doesn't matter how low a price goes; the item stays until it's bought, even if it ends up selling for $1.00.

Ready for the communications part of the system? Then move over to the online CB simulator and talk to other

READER SERVICE INDEX

Page No.	Company	Svc. No.
48	Abacus Software	142
38	Abacus Software	141
116	Abby's Discount Software	278
C-4	Access Software, Inc.	286
26	Ace Software	260
10	Accolade, Inc.	299
49	Activision	266
49	Activision	267
56	Activision	276
12	Alsoft	121
92	APCAD	261
109	Aquarian Software	136
10	Artworx Software Co., Inc.	300
107	Big L. Electronics	137
15	C.O.M.B. Co.	248
74	C.O.M.B. Co.	258
83	C.O.M.B. Co.	257
12	CBS Software	116
54	CBS Software	274
74	Central Point Software	115
119	Cheatsheet Products, Inc.	279
103	Comal Users Group, USA, Ltd.	–
20&21	CompuServe	125
4&5	Computer Centers of America	127
C-3	Computer Devices International	138
94	Computer Information Ltd.	256
24	Computer Management Corp.	264
25	CVC Online	263
41	Datamost Inc.	295
7	Digital Vision, Inc.	130
53	Electronic Arts	273
49	Epyx	268
55	Epyx	275
112	Floppy House Software	247
13	HBH Corporation	117
14	HBH Corporation	118
46	Hesware	106
84	Inkwell Systems	297
8	Integrated-Software Systems	253
44	IRQ, Inc.	107
101	Jason-Ranheim	131
113	Jason-Ranheim	112
11	John Henry Software	102
59	John Henry Software	288
42&43	Lyco Computer	128
16-17	Mastertronic International Inc.	265
31	Megasoft Ltd.	252
47	Megasoft Ltd.	251
36	Micro-W Distributing, Inc.	289
27	Microcomscribe	292
34	MicroProse Simulation Software	246
10	MicroProse Simulation Software	101
120	Microvations Houseworks	113
22	Mimic Systems Inc.	294
70	N Systems	123
71	Navarone Industries, Inc.	250
93	Ohio Computer Services, Inc.	262
78&79	Okidata	126
81	Passport Designs, Inc.	280
99	PDS Sports	134
11	Peri-Comp, Inc.	103
72	Peripheral Systems of America	133
11	Post Technologies, Inc.	120
113	Professor Jones	108
62-69,96	Protecto Enterprizes	293
90	Quorum International, Unltd.	122
23	R.J. Brachman Assoc., Inc.	259
45	Roger Wagner Publishing, Inc.	109
9	Sakata U.S.A. Corporation	298
117	Schnedler Systems	–
29	Skyles Electric Works	272
87	Solid State Software	104
6	Starpoint Software	140
50	Strategic Simulations Inc.	269
C-2	SUBLogic Corporation	249
145	TPUG Inc.	124
86	T&D Subscription Software	254
50	Telarium	271
50	Telarium	270
57	The Avalon Hill Game Co.	277
80	The Comtel Group, Inc.	135
81	The Comtel Group, Inc.	281
89	Ultrabyte	114
86	Universal Marketing	255
95	Value-Soft	290
114&115	Viewtron	129
84	Visp Soft	296
11	VMC Software	119
110	Wedgwood Rental	111
111	White House Computer	132
104	World of Commodore III	287
88	Xetec Inc.	110

The publisher cannot assume responsibility for errors in the above listing.

www.commodore.ca

NEW For C-64

PERSONAL DISK DIARY

Featuring:

★ Day to Day Entry
★ Word Search
★ 30 Pages of Notes
★ Full Calendar For Any Year
★ Password Security
★ Address Book
★ Printer Optional

***Only* $22.00**
(Postage & Packaging Included)

VISP Soft

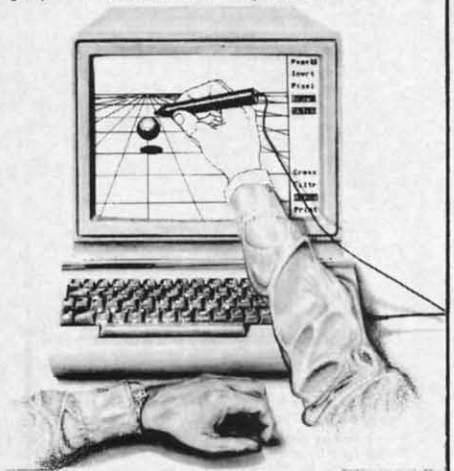
Send Check or Money Order To:
VISP Soft
P.O. Box 308
Westfield, MA 01086

Disk Drive Required

Reader Service No. 296

EXTEND YOUR REACH

Flexidraw, the light pen system for Commodore users who want easy Computer Aided Design with professional results. Whether drawing schematics or practicing your artistic talents, FLEXIDRAW will give you over 65 graphic features including two full screens, 16 colors, sprite animation, and a library of technical symbols. The FLEXIDRAW system; quality light pen, disk, and manual, only $149.95.



From the makers of Penware:
Flexidraw · the Graphics Integrator · Flexifont

Inkwell Systems
P.O. Box 85152 MB290
San Diego, CA 92138
(619) 268-8792

Reader Service No. 297

Viewtron subscribers. Live and in color! With subscribers from around the country, you're sure to always have someone to chat with.

Need help or advice about something? Query one of the many qualified experts in the "Ask Our Expert" section. Consumer advice, pet care, fashion and beauty, even plant care. And if your worries are getting you down you can shift to the travel section and book yourself a seat on a flight to any point on the globe. You can even make reservations for a trip by hot air balloon or relaxing ocean cruise.

After coming back from your vacation you might find yourself wondering about how the rest of the world fared while you were away. Flip on the computer and dial up Viewtron! World, national, and local news are available. (Viewtron's parent company is Knight-Ridder Newspapers, Inc., so you know you'll always have access to late-breaking headlines.) Sports news is of primary importance to many, and whether your sport is baseball, soccer, or auto racing, you can get full and accurate reports about any of them.

Perhaps, like many of us, you're watching your money carefully these days. By offering several "bank at home" services, Viewtron makes it simple. You can also get up-to-the-minute reports on the stock market, commodities, bonds, gold, and worldwide currencies. Information about the real estate market is also to be had.

Like a little entertainment after all this? Scan over to the games section. Many different games and quizzes are online, along with a self-test section that can help you learn more about the type of person you are. And with trivia all the rage nowadays, Viewtron was not about to be left out; its trivia section has questions geared to experts and beginners. The silver screen is not forgotten either. (That's movies to you youngsters.) Movie news, movie trivia, movie reviews, and even an Academy Awards quiz can be found.

Are you a soap opera buff? My wife is a devout fan of several, but finds it hard to watch both the shows and our children during the day. She generally tapes the shows and watches them after the kids are asleep, but what to do when the VCR chews up the last 15 minutes? Viewtron to the rescue! With a complete summary of each show posted daily we never have to worry about missing our soaps.

Some of you may be asking, "Gee, B.W., this sounds great, but how's the service? What if I have a problem?" Let me reassure you. Viewtron has the best customer service department it has ever been my pleasure to call! I cannot believe how friendly and willing to help they are. There are no delays or forgotten promises to "call you back" here. These people are professionals and it shows. It seemed to me that everyone I dealt with at Viewtron had a real and honest desire to help me in any way possible.

What does access to this amazing world of information cost? $9.95! That price includes a disk with the special terminal software needed to use the system, one free hour of service, an ID and password, and a user manual. Subscribers pay only for their use of the service. There are no monthly fees or minimum use charges. Connection rates are $.09 per minute weekdays after 6 p.m. and weekends, and $.22 per minute weekdays. There is no extra charge for use at 1200 baud. (Prices may vary in some cities.)

Viewdata Corporation of America, Inc., 1111 Lincoln Road, 7th Floor, Miami Beach, FL 33139 (phone: 305-674-1444). *—B.W. Behling*

MICROLINE 192 PRINTER

Okidata
$499.00

I have been using Okidata products as far back as the Microline 82A. They have established a reputation as a company that manufactures fast, dependable printers. The 82A was a 120 cps workhorse, and the 92A basically a "super" 82A; but the 192 is a quantum leap forward.

The housing for the 82A and 92A was far from sleek. They were big, heavy, and relatively noisy. The 192,

www.commodore.ca

on the other hand, is slim, trim, and quiet. Housed in a tan plastic case (14.2"W X 10.8"D X 3.2"H), it weighs in at a mere 8.8 lbs. The first impression that this printer gives is that of a toy. I lost count of the people who saw it and asked, "Is it thermal?" Once this tiny powerhouse is activated, though, all skepticism is overcome.

Two primary text modes are available: Data Processing (DP) and Correspondence Quality (CQ). Print speeds are 160 cps and 33 cps respectively. If you were to race the 192 against another 160 cps printer, the 192 would win. This is because the 192 kicks into a 200 cps mode when it encounters spaces. If you were printing out two columns, the printer would output the data at 160 cps and speed up to 200 cps as it moved on to the next column. The CQ mode is one of the finest I have ever seen. I would consider it perfectly acceptable for all applications short of a resume. When in this mode, the printer uses a two-pass technique to print in a 17 X 17 matrix (DP mode prints in a 9 X 9 matrix). Additionally, in emphasized or enhanced modes the speed is 80 cps.

No fewer than 13 character sets are included, among them ASCII, no-slash zero ASCII, international sets, and an IBM character set capable of producing IBM graphics characters.

On ground equally high as the text modes are two graphics modes: APA, and Block. The Block graphics are standard TRS-80 graphic characters, but the real power lies in the APA graphics. This is the dot-addressable graphics mode. Available are six resolutions: 60 X 72, 72 X 72, 120 X 144, 144 X 144, 240 X 144, and 288 X 144 (all resolutions given in dots per inch). This variety is sufficient for almost any graphic application.

Besides the raw power of the printer hardware, this printer is a pleasure to use for two other reasons, namely the manual and the menu select mode. The manual is very well written. It will taken even the beginner through the setup and use of the printer. The menu select mode is a system by which the front panel

MUST LIQUIDATE!
LIMITED INVENTORY OF ULTRA FAMOUS
64K COMPUTERS
AT FAR BELOW DEALER COST!

Monitor is not available. See footnote.*

Millions sold at regular prices . . . and STILL being sold at regular prices!

Sorry, we're NOT permitted to print the brand name. But if you phone us, all details can be divulged.

Phone Toll-Free: 1-800-328-0609

PLENTY OF SOFTWARE AVAILABLE! Thousands upon thousands of programs are available for it from computer stores across the nation.

FOR BUSINESS! This is an ideal entry level business computer. Use the computer for word processing, data base, accounts payable/receivable, general ledger, payroll, inventory, tax accounting, spreadsheets, mailing lists . . . and much, **MUCH** more!

FOR EDUCATION! Perfect for everyone from Ph.D. candidates to preschool youngsters. A vast number of programs are available to challenge and educate the mind.

FOR HOME! Use for analysis of personal investments, income tax planning, storage of recipes and household data. And of course, there are **hundreds** of exciting, fast-paced arcade games available!

Units sold SINGLY or in any combination you desire.	Original List Price	YOUR COST
64K COMPUTER Item H-572-63553-00	$200.00	**$99**
DISK DRIVE Item H-572-63553-01	$269.00	**$148**
PRINTER Item H-572-63831-00	$200.00	**$118**
We reserve the right to limit quantity purchases. **TOTAL**	$669.00	**$365**

COMPUTER
64K RAM. 20K ROM standard (includes operating systems and BASIC interpreter). Text characters. High resolution graphics. 320 x 200 pixels. Music synthesizer produces 3 independent voices, each with a range of 9 octaves.

Typewriter keyboard with 66 keys for upper and lower case letters, numerals, symbols, reverse characters. 2 cursor control keys. 4 function keys, programmable to 8. Input/output ports for user, serial, ROM cartridge, 2 joystick/paddles, video, cassette drive interface. 16 background, border, character colors.

DISK DRIVE
Intelligent, high-speed. External 5¼" floppy diskette recorder. 2K RAM, 16K ROM. Maximum storage of 170K formatted data; 35 tracks. Uses single sided, single density disk. Serial interface. Second serial port for chaining second drive or printer. Data transfer rate of 400 bytes per second.

DOT MATRIX PRINTER
Bi-directional 6x7 dot matrix impact printer. 60 characters per second. Has upper and lower case letters, numerals and symbols. All PET graphic characters. Standard friction feed. Maximum of 80 columns width, dot addressable. CBM ASCII character codes. Original plus maximum of two copies. Paper width: 4.5" to 8.5". Size: 13"W x 8"D x 3¼"H. Weight: 6½ lbs. Power requirements: 120 volts AC, 60 Hz.

Factory Reconditioned With Factory Warranty.

*NOTE: If you do not have a monitor, you can use your TV as a display unit. The computer comes already to be hooked up to your TV.

Credit card customers can order by phone, 24 hours a day, 7 days a week. VISA MasterCard AMERICAN EXPRESS DINERS CLUB

Toll-Free: 1-800-328-0609

TeleCheck Your check is welcome! No delays in orders paid by check.

C.O.M.B. CO.®
Authorized Liquidator
14605 28TH AVENUE NORTH
MINNEAPOLIS, MINNESOTA 55441-3397

C.O.M.B. CO.® Item H-572
14605 28th Ave. N./Minneapolis, MN 55441-3397

Send the following items: (Indicate quantity of each):
___ 64K Computer(s) Item H-572-63553-00 at $99 each plus $8 each for shipping, handling.
___ Disk Drive(s) Item H-572-63553-01 at $148 each plus $8 each for shipping, handling.
___ Printer(s) Item H-572-63831-00 at $118 each plus $7 each for shipping, handling.
(Minnesota residents add 6% sales tax. Allow 3-4 weeks for delivery. Sorry, no C.O.D. orders.)
☐ My check or money order is enclosed. (No delays in processing orders paid by check, thanks to TeleCheck.)
Charge: ☐ MasterCard® ☐ VISA ☐ Am. Ex. ☐ Diners Club
Acct. No. _____ Exp. _____
PLEASE PRINT CLEARLY
Name _____
Address _____
City _____
State _____ ZIP _____
Phone _____
Sign here _____

Reader Service No. 258

www.commodore.ca

FREE 10 DISKETTES OR 20 C-20 CASSETTES

A subscription to the **'Cassette of the Month'** gets you a tape or disk full of 10 **quality Commodore 64 programs** delivered to you by first class mail every month. The documentation included will help you run great **utilities** like 'Word Processor,' and 'Budget Analyzer,' or enjoy great **games** like 'Frogjump' and 'Caterpillar Cave' **FOR AS LITTLE AS 50 CENTS EACH!**

★ **Limited offer** ★ Subscribe for a year on cassette and receive **20 Free C-20 cassettes** or subscribe for a year on disk and receive **10 Free 5¼ single sided double density diskettes!**

PRICES

	TAPE	DISK
1 YR (12 ISSUES)	60.00	75.00
6 MO (6 ISSUES)	35.00	45.00
Single Copies	7.00	9.00

★ We've been in business for over three years! acquiring
★ Over 4000 satisfied color computer owners
★ Commodore 64 required

Mich. Res. add 4%
Overseas ADD $10 to subscription and $1.00 to single issues.

PERSONAL CHECKS WELCOME!

T & D Subscription Software
P.O. BOX 256-C
HOLLAND, MI 49423
(616) 396-7577

Reader Service No. 254

the 21 Second Back-up

NEW

AN IDEA WHOSE TIME HAS COME!

$79.95

THE COPY PROGRAM
- EXACT COPIES IN ONLY 21 SECONDS
- SIMPLE-FULL BACK-UP IN A SINGLE KEY STROKE
- RELIABLE—WORKS VIRTUALLY EVERY TIME

NASHUA DISKETTES

$10.95 bx

UNIVERSAL

475 MAIN ST FARMINGDALE, NY 11735

(516)753-0110

MASTERCARD, VISA, M.O. OR CHECK
NY ORDERS ADD SALES TAX
Prices are mail order only and do not include shipping & handling

Reader Service No. 255

switches and interactive software within the printer firmware guide you through the printer options. As you select and change things such as print modes and character sets, the printer echoes your selections, making offline parameter changes a breeze.

One last feature is an 8K printer buffer. This is a good size, as it can accept up to medium size documents, freeing up your printer for other work.

The Okidata 192 is an excellent value for the money, and I recommend it for low and high end applications alike.

Okidata, 532 Fellowship Road, Mt. Laurel, NJ 08054 (phone: 609-235-2600). *—David Barron*

THE MACHINE SHOP

FS! Software
Commodore 64
Disk; $40.00 (see below)

Discussions on machine language programming are generally concerned with the creation of programs. Very little is said on the analysis or disassembly of machine code into a more readable form. There are numerous utilities available which support the creation of machine language programs. These range from simple machine language monitors to full blown macro-assemblers. The former will handle code just one instruction at a time. The latter support the creation of multi-kilobyte programs. Although machine language monitors will disassemble code one line at a time, they are totally inadequate for programs longer than a few dozen instructions.

As Commodore users we encounter machine language programs in great abundance. They are frequently used as subroutines to speed up critical operations that BASIC just can't handle. A tool to convert this code into a more readable form would be a great boon to anyone wishing to learn more about the subject. *The Machine Shop* from FS! Software is just such a product. It permits the creation of machine language programs from nearly standard MOS source code, as well as the reverse process.

The Machine Shop for the C-64 is not a new product. It is actually an updated reincarnation of *Develop-64,* one of the first assembler development systems for the C-64 and the VIC 20. Originally published by French Silk Smoothware, the company has been reformed under the name of FS! Software. The company's name derives from Don French, the proprietor, and the Jacquard loom. This device was the first to utilize a punch card program for the weaving of French Silk in the early part of the 19th century.

The package is actually two utilities which can be configured to interact in a most intimate fashion. The first is *Develop-64*, a full-featured Macro Assembler. The second is *Decode-64*, a three-pass symbolic disassembler and cross-referencer. It is the latter program which gives *The Machine Shop* its unique and powerful capabilities.

Decode-64 can analyze a block of data, in RAM or on disk, and generate a fully cross-referenced assembler source code file, complete with labels, for user analysis or for incorporation into a *Develop-64* source program.

The best part of the deal is that *The Machine Shop* does not have to cost you a penny. It is one of those rare "freeware" packages where the user pays for the program after having tried it out. Anyone who already has a copy is encouraged to distribute additional copies (of the disk only) to anyone else who might be interested. This is a complete turnaround from the copy protected original distribution of this package. If you like what you see, a $40 payment to FS! Software will get you the complete documentation for the package. You will also receive with your payment a one-year subscription to *Machine Code*, a new quarterly magazine for machine language programmers. The first issue is expected to be published at about the time you read this (Fall 1985). Note that FS! Software will only supply *The Machine Shop* as a complete package for the established price.

DEVELOP-64

This is the heart of the package. It is a full-featured macro-assembler which closely follows many of the

www.commodore.ca

conventions of the original MOS Technology specifications for 6502 source code. It features an easy to use full screen editor based on the native editor of the C-64. As with the BASIC editor, line numbers are required. However, these are automatically generated. The user references the line numbers only when listing, inserting, or deleting lines. The editor also supports string search and replace, as well as block move and copy. The program performs syntax checking on each line as it is entered. This insures that the resulting source code listing will at least be syntactically, if not logically, correct.

Source code programs, in memory, can be up to 2200 lines long. This in no way limits the size of a complete program. *The Machine Shop* fully supports the disk drive. Any number of disk files can be linked up to the capacity of a single disk. In addition, *Develop-64* supports macros and library files. The former allow passing of up to eight parameters to the subroutines. The latter do not permit parameter passing.

Macros and library files in machine language are similar to subroutines in BASIC. They are standard blocks of code which you may use in many of your programs. Although they are not an essential feature for the writing of machine language programs, they are a great convenience. The macro feature allows the source program to call in these subroutines as required. The final code is assembled with the macro code as an integral part.

Interestingly enough, *The Machine Shop* distribution disk lacks a dedicated machine language monitor (MLM), even though there are a number of very good ones in the public domain. (Most other assembler packages we have seen include one of these MLMs.) *Develop-64* does have most of the MLM features built in. The Debugger function is extremely effective in this regard. This built-in utility lets you display memory 160 bytes at a time. The ASCII representation of the code is listed alongside the hex values. In single step mode the contents of memory

GUARANTEED SOFTWARE

NEW

VIZASTAR for the C128

Vizastar, the integrated spreadsheet, database and graphics program that has the Commodore 64 world raving, is now available for the C128. It boasts 80 columns, and has over 40K of free memory in the spreadsheet. Those who already own Vizastar 64 will be pleased to know that your existing files can be read by Vizastar 128. Also, you can upgrade to the 128 version. Call us for details and pricing.

"The only other comparable product would be Lotus 1-2-3 for the IBM PC; nothing in the C64 world comes even close to the features of Vizastar."

AHOY July 85

"I found Vizastar would do anything Lotus 1-2-3 could, and then some. It's my Commodore choice to become the standard against which the others will be judged."

INFO 64 Magazine, Issue #7

"Vizastar is an exceptional package that rivals the features of programs such as Lotus 1-2-3 and offers C64 owners the kind of integrated software previously only available for higher-priced systems."

RUN Magazine, June 1985

"I scrutinized, tested and experimented with Vizastar extensively, but could find no weaknesses whatsoever. It is the most comprehensive, most flexible, most powerful and easiest to use integrated software package I've worked with."

Commodore Microcomputer, Sept/Oct 1985

"I use an IBM PC at work with Lotus 123. I feel Vizastar is just as good and in someways better than 1-2-3."

Steven Roberson, NC. End User

"I have used Multiplan and Superbase; both are good pieces of software, but are inadequate when compared to Vizastar."

Jim Mathews, WA. End User

"So good, I bought a second C64 and Vizastar for my office. A wild bargain! You've saved me from having to buy IBM and Lotus."

Philip Ressler, MA. End User

NEW

VIZAWRITE CLASSIC for C128

This is the new word processor from Vizastar's author, Kevin Lacy and is the successor to Omniwriter, which he also wrote. All the features of Omniwriter are there, plus many significant enhancements, like auto pagination, on-line help, pull-down menus, full-function calculator and more. Up to 8 'newspaper-style' variable-width columns can help with newsletters.

Three different proportionally-spaced "near letter quality" fonts are also built-in for use with Commodore or Epson compatible printers. You can merge almost any other word processor file directly into Vizawrite, including Paper Clip and Omniwriter. Naturally, it is also compatible with Vizastar. At all times, what you see on the screen is exactly the way it will be printed out. Vizawrite can do mail-merges and has an integrated 30,000 word spelling checker that you can expand yourself.

PROGRAM SPECIFICATIONS

Both Vizawrite and Vizastar are written in 100% machine language and run in the 128's FAST mode, making it lightning fast. They require a C128 with 80 column color or monochrome monitor. Both come with a cartridge, a diskette, a backup, and a reference manual. Vizastar also includes a 50 page tutorial book. Both work with 1541 or 1571 disk drives.

RISK-FREE OFFER

Vizastar 128 is priced at $119.97. Vizawrite's price is $79.97, but as an introductory offer, it is now only $69.97. Vizastar 64 XL8 is now available for $119.97. We are so positive you will be satisfied with our programs that we offer a 15-day money-back guarantee. Try it Risk-Free. Call us today or send a check or money order. VISA/MC accepted.

Calif. residents add 6.5% Sales Tax.
Add P&H: UPS-$4; COD/Canada-$7.

SOLID STATE SOFTWARE

1125 E. Hillsdale Blvd., Suite 104
Foster City, CA 94404-1609
(415) 341-5606

• Dealer Inquiries Welcome •

Trademarks: Lotus 1-2-3/Lotus Development. Commodore 64/Commodore Electronics Ltd. Multiplan/Microsoft

Reader Service No. 104

www.commodore.ca

are disassembled one instruction at a time. The contents of all of the microprocessor's internal registers are continuously displayed and updated. The trace function disassembles code in a continuous stream while maintaining the control of the single step feature. Output can be directed to either the printer or the screen. Up to ten break or go points can be set for the trace mode.

DECODE-64

This program is the highlight of the package. It generates a fully labeled disassembly of a block of code from memory or disk. The output can be sent to either the screen, printer, or disk. In the case of the latter, the resulting source files can be directly used by *Develop-64*. The size of disk files are automatically limited to 2000 lines each. Labels are created following a predefined set of rules. Each label contains the originating address in hexadecimal or decimal. The latter are useful for referencing by BASIC's SYS cells. The labels also identify the calling instruction as to type. For example, a J indicates a JMP and an R indicates a relative branch. External references are also identified, as are all zero page references.

As we mentioned, *Decode-64* is a three-pass symbolic disassembler. This means that each disassembly is created on the third scan of the data. In the process a complete symbol table of all the generated labels is created. A useful adjunct to *Decode-64* is the built-in cross referencer. When activated, this routine scans the resulting source code symbol table. All labels are listed with every line number at which they appear. An asterisk identifies the line number which defines the label.

Not all machine code represents executable instructions. Many parts of a program are actually data such as screen messages, numerical constants, program vectors, and so on. Trying to disassemble these parts of the code will waste processing time at best and produce misleading and incorrect code at worst. *Decode-64* lets you mark off blocks of code as data not for disassembly. Up to eight such areas can be so defined.

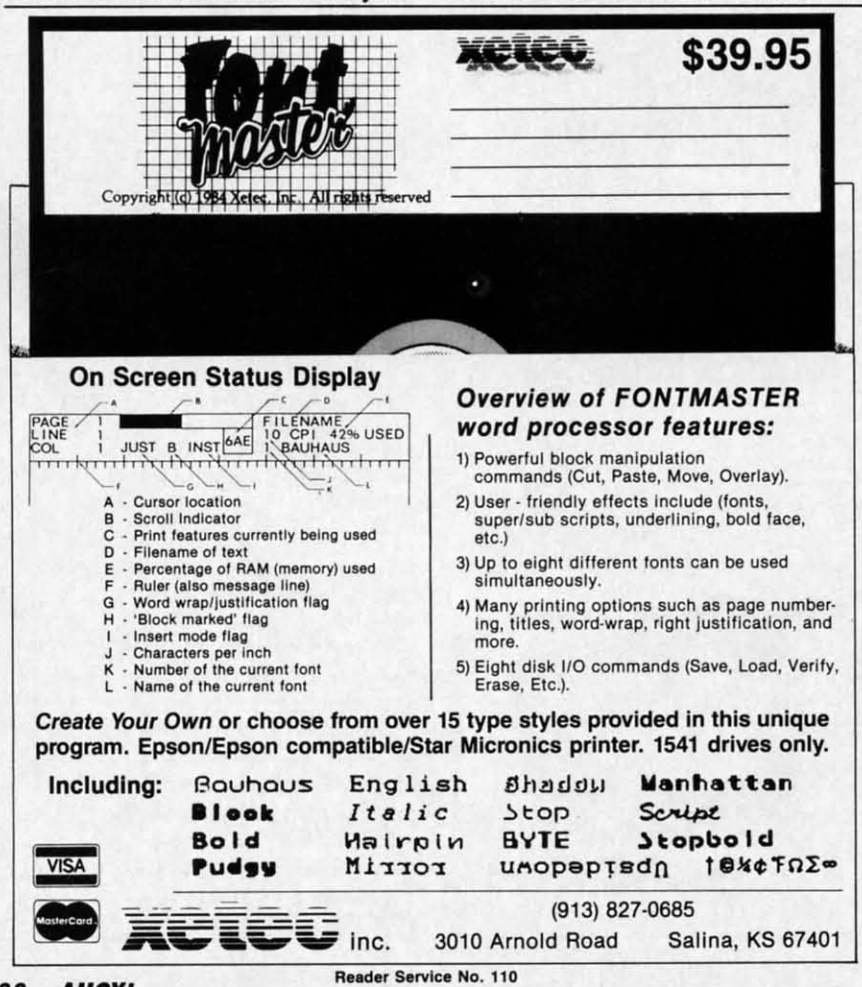
It is up to the user to select the code for this restricted treatment. This can usually be done by a quick disassembly with an MLM. When a lot of illegal opcodes start to show up, this is most likely a data area. A simple memory scan will also serve to display any readable ASCII message blocks. The Debugger portion of *Develop-64* will do this job well.

Both *Decode-64* and the Cross Referencer can be merged with *Develop-64*. The result is a tradeoff where some features are enhanced while others are limited. Most significant is the ability to preassemble a label file which identifies some of the more common machine language calls in the C-64. For example, a complete list of all of the Kernal calls is included on the distribution disk. The symbol table created by this preassembly is kept in memory when *Decode-64* is used on an object program. All calls to the predefined symbols will be properly labeled in the resulting disassembly. The only significant limitation to the merged *Decode* and *Develop* utilities is the inability to mark off data areas in the object code.

THE DOCUMENTATION

We had better go into some detail on this, as it may be the only part of the package you will be paying for. What you get will actually be two books. One of these is a 14-page pamphlet which describes the operation of *Decode-64* in some detail. Although quite readable by most software documentation standards, you will be well advised to run *Decode-64* through its paces as you go through the book. Nothing beats hands-on experience when learning a complex package of this sort.

The other book is a nine chapter, nine appendix volume called *Inside the Commodore 64* by Don French. Don't panic! The instructions for *Develop-64* are only a small part of this, appendices E and F to be exact. The first is a quick run through of its features, while the second takes care of



Reader Service No. 110

www.commodore.ca

the details. Once again, keeping the program and computer at hand will be helpful.

The major part of the book is devoted to a detailed description of the hardware and software features of the C-64 as they relate to machine language programming. The chapters are about equally divided on the instruction set of the 6502/6510 microprocessor and the specific details of the C-64 environment. Emphasis is given to the operation of the C-64 graphics and sound chips. Conspicuous by its absence is a discussion of the C-64 I/O functions of the user or serial ports.

CONCLUSIONS

The Machine Shop in its present incarnation is definitely a best buy. If you have not yet selected an assembler/disassembler for your own use, you will not go wrong with this one. Rumor has it that something of a cult following has developed over the original issue of this package. At least one major text on C-64 assembly language has adopted *Develop-64* as the assembler for all the in text examples. *Assembly Language Programming with the Commodore 64* by Marvin L. DeJong is a 296+ page text on the subject available from Brady Communications (division of Prentice Hall).

FS! Software, P.O. Box 635, Faribault, MN 55021 (phone: 507-332-8122). —*Morton Kevelson*

THE B.E.S.T. GENERAL LEDGER MANAGEMENT SYSTEM

B.E.S.T. Inc.
Commodore 64
Disk; $69.95

If you're looking for accounting software, you're probably planning to do more than prepare your household budget. To set up a computerized accounting system with a minimal financial outlay, a C-64 and *The B.E.S.T. General Ledger Management System* represent a reasonable choice.

Such a system would be suitable for keeping the books of a small business, social organization, or (for you diehards) household. I include only small businesses more for hardware

a Breakthrough in Value!

ULTRABYTE

NEW PRODUCTS - ALL with FREE BACKUP COPIES

ULTRABYTE DISK NIBBLER

Uses proprietary revision of disk operating system (DOS) to make a bit by bit copy. This process, called nibbling, copies the format, data and errors, all at the same time. Errors are not recognized and do not hammer the drive. Built in fast load environment allows copies in 8 minutes with a single 1541 disk drive.

- Easy to use. No separate analysis or error production
- Choose 35, 36, 37 or 38 tracks to copy incl. half tracks
- Makes up to 5 copies of each original — (UNIQUE FEATURE)
- Backs up 99% of all protected software
- Free backup copy - two disks for the price of one

$39.95 FOR 2 COPIES OF THE DISK NIBBLER

plus $4.00 shipping and handling. Mastercard, Visa, Check or M.O.. Foreign orders or COD add $2.00. Calif. add 6.5% ($2.60) sales tax. Additional backup copies are $20.00 plus $3.00 shipping. ATTENTION: Owners of ULTRACOPY whose names are in our file may order a single copy for $20.00 plus $3.00 shipping.

FREE BACKUP DISKS

McMURPHY'S MANSION

New AN ULTRABYTE TEXT ADVENTURE

It seems that an unknown uncle has died in faraway Scotland and left you his vast fortune. $10.000.000 to be exact !!

There is only one small problem. Dear old uncle McMurphy was a bit demented and has hidden your inheritance somewhere in his mansion.

He left you only one small clue, a brass key.

It is now up to you to find your fortune.

- 30 - 60 hours of challenge
- Sealed envelope of hints for beginners
- Instruction manual included
- Solution sheet available

$29.95 for 2 disks
plus $4.00 shipping. COD or Foreign add $2.00. CA tax $1.95

THE DISK SURGEON

New AN ULTRABYTE DISK UTILITY

- Read data from, edit and write to disk
- Analyze disk errors without "hammering"
- Write errors 20, 21, 22, 23, 27 & 29 on disk
- Read data under errors, erase errors
- Decimal, HEX, Octal & Binary conversions
- Format a single track with any ID
- Detailed instructions for beginners
- Much more

INCLUDES THREE PUBLIC DOMAIN COPY PROGRAMS THAT ARE NOT PROTECTED

1) FILE COPY to copy portions of a disk
2) SINGLE TRACK COPY
3) 4 MINUTE COPY of entire disk. Protected disks also require error production

$29.95 for 2 disks
plus $4.00 shipping. COD or Foreign add $2.00. CA tax $1.95

All Products for Commodore 64 & 1541 Disk Drive Only
Write or Call 24 Hour Order Line
ULTRABYTE (818) 796 - 0576
P.O. Box 789 La Canada, CA 91011

DEALERS & DISTRIBUTORS WANTED

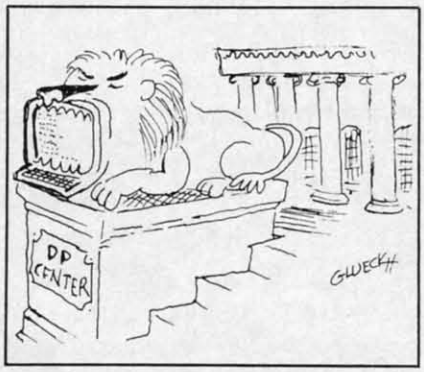
Reader Service No. 114

www.commodore.ca

than software reasons. When you reach thousands of transactions per month, the speed of the C-64 and its drives is just not adequate.

Please notice that I said "keep the books." Neither B.E.S.T.'s software nor any other accounting programs can do true accounting; they all do bookkeeping. When your business and your livelihood depend on predictions that income will exceed expenses, human beings are necessary for two reasons.

First, the books must be interpreted. Some facts about your customers and your vendors just cannot be plugged into a computer program. Second, people must enter financial data into the computer. Accounting software eases the (necessary) pain of keeping the books; it doesn't free you of the task.



That caveat behind us, let's take a look at the power of B.E.S.T.'s *General Ledger.* On a single data disk, you can store 24 months of financial activity. This is a long enough period that you can do some meaningful trend analysis. For each month you can store up to 1500 general ledger transactions entries. As with any general ledger, you must decide the level of detail that each entry will represent. You can, for example, enter each cash sale for the day, or you may choose only to enter total cash. Obviously, the more detail you choose, the more work you will have to do and the better your predictions will be.

You may set up as many as 200 general ledger accounts or types of transactions. Typically, you begin by manually preparing a Chart of Accounts that lists each type of financial transaction. Accounts are usually grouped into categories including assets, liabilities, capital, income, and expense. Types of transactions under expense might include accounts like salaries, payroll tax, insurance, rent, telephone, postage, and supplies.

To let you watch future income and expenses, B.E.S.T.'s *General Ledger* allows 200 accounts receivable entries per month as well as 200 account payable entries. B.E.S.T. has separate programs available for each function. All transactions are entered through the general ledger and may then be automatically posted to the other programs.

Accounts receivable should be maintained separately if you extend a lot of credit to customers. Your general ledger will only show the account and the amount owed. In a separate accounts receivable, you would have more information, like a complete customer list, items purchased, amount owed, due date, and a record of credit payments. You could produce aging reports listing payments 30, 60, or 90 days overdue, and maybe even print invoices. A separate accounts payable package would be similar, but would deal with vendors to whom you owe money rather than customers who owe you.

Other than automatic posting to accounts receivable and payable, the most important feature of B.E.S.T.'s *General Ledger* is its report writer. You can create, edit, list, print, or delete reports with almost any combination or permutation of the data you have entered. Only if you have had to use a report format designed by someone else unfamiliar with your business will you know how important custom design can be.

And B.E.S.T. has built-in flexibility. When you decide to print your report, you will find that several variations are allowed without having to go back and edit the report design. Balance sheets, income statements, comparison reports (year-to-year or quarter-to-quarter), and performance analyses (percent-of-total reports) are all available.

If you'd like, this program will even print your checks automatically. I was especially fond of the "test print" feature that let me check the alignment of the checks in the printer. Although I began with a complaint about the speed of the C-64, you need to know that *General Ledger* can bring any account to your screen for review within 4 seconds. It does so by indexing information on the disk. The price you pay is a 16- to 20-minute wait (with a 1541 drive) while the program formats a new data disk and creates its files.

Odds 'n ends: B.E.S.T.'s *General Ledger* will work with one or two drives. Two means much less disk swapping. Error messages are translated into English. Sorry if you have all those numbers memorized. The program completely controls the keyboard. Neither accidental nor intentional improper keystrokes will crash the program. Last and best, B.E.S.T. allows you to back up the software. Any old 1541 backup utility will do, thanks.

Business Electronics Software and Technology, Inc., P.O. Box 852, McMinnville, OR 97128 (phone: 503-472-9512). ***—Richard Herring***

FREE? DISKETTES

SAVE MONEY! C64 and VIC 20 users can use the diskette flip side, if another 'write enable' notch is correctly made.

TOP NOTCH™ by QUORUM quickly **solves** that **problem.** It's like FREE DISKETTES!

- Stainless Steel Guide
- Easy Leverage Handle
- Clippings Catcher
- Square Notch Cut
- Black Finish

Lifetime Warranty

Get THE BEST! Ask our customers;
U.S. National Bureau of Standards
TRW • IBM • Digital Research • AT&T
Ciba-Geigy Pharmaceuticals • International Paper
U.S. Postal Service • Rockwell International
General Electric Corporate Research & Development
Alcoa • U.S. Naval Weapons Research Center

100% Money Back Satisfaction Guarantee!

TOP NOTCH™ Just $14.95
Add $2.00 s/h • CA Res. add .97¢ tax

ORDER DIRECT FROM QUORUM
1-800-222-2824
In CA: 1-800-222-2812
Call 6 a.m.-6 p.m. PST Dealers Invited

QUORUM INTERNATIONAL, Unltd.
INDUSTRIAL STATION P O BOX 2134-AH
OAKLAND, CA 94614

Reader Service No. 122

www.commodore.ca

COMMODORE ROOTS

GETTING GRAPHIC

ASSEMBLY LANGUAGE TECHNIQUES FOR GRAPHICS PROGRAMMING

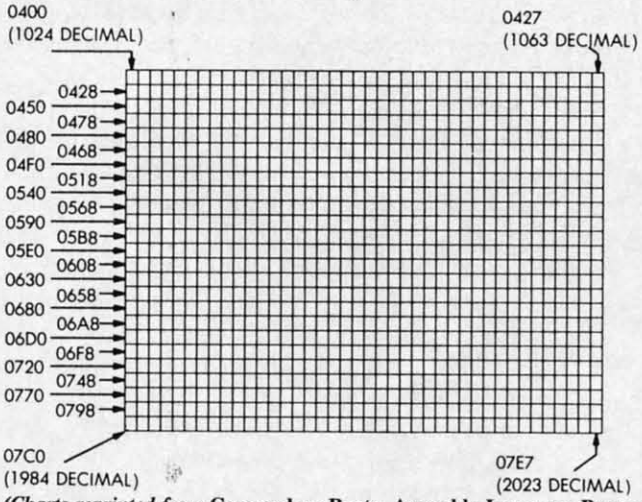
By Mark Andrews

If you've ever tried to write a high-resolution graphics program and wound up frustrated by the graphics limitations of Commodore BASIC, agonize no more. In this and the next few editions of *Commodore Roots*, we'll explore some of the techniques professional assembly language programmers use to write arcade-style games and other kinds of graphics-oriented programs.

As you may know, the Commodore 64 has two primary screen modes: a text mode and a high-resolution graphics mode. In text mode, the C-64 is capable of displaying up to 1,000 characters at a time on its screen, arranged in 25 lines of 40 characters each. To hold the 1,000 characters, the 64 uses a specific block of memory exactly 1,000 bytes long.

This segment, called screen memory, normally starts at Memory Address 1024 ($400 in hexadecimal notation) and extends to Address 2023 ($7E7 in hexadecimal). It is often pictured as a grid of rectangles measuring 40 columns wide by 25 rows high, with each rectangle representing one character on the screen. Figure 1 is a map of the segment of memory most often used as screen memory in Commodore 64 programs.

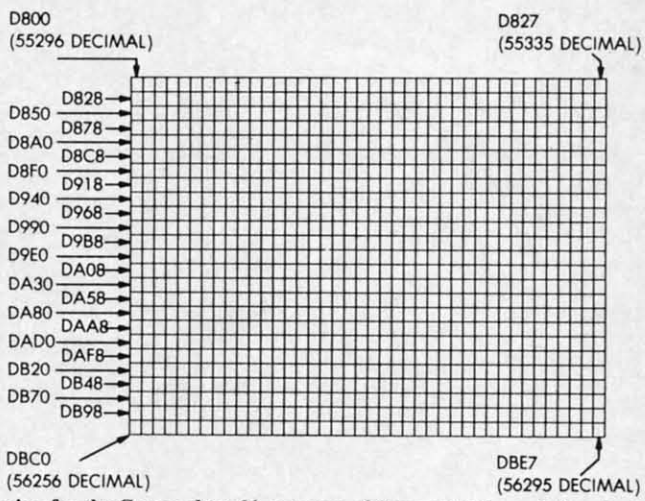
Figure 1: A C-64 Screen Map



Once you know what these screen display codes are, and where the screen display memory in your computer is, you can print text and graphics characters on your computer's screen by POKEing their screen-code values directly into the appropriate addresses in screen memory. In this way, you can bypass your computer's screen editor anytime you like, and print anything you like directly on your screen.

In addition to its 1,000-byte block of screen memory, the Commodore 64 has a corresponding block of color memory. This segment of RAM begins at Memory Address 55296 (hexadecimal $D800) and extends to address 56295 ($DBE7 in hex notation). Color RAM, like screen RAM, can be visualized as a 40-column by 25-line matrix of rectangles, with each rectangle representing the color of one of the 1,000 characters that can be displayed on the C-64 screen. Figure 2 is a map of the block of color RAM used by the Commodore 64.

Figure 2: Map of C-64's Color RAM



(Charts reprinted from **Commodore Roots: Assembly Language Programming for the Commodore 64,** *courtesy of Howard W. Sams & Co., Inc.)*

When you type a character, your Commodore 64's operating system translates that character into a code, then prints the character on your screen by storing its code number in the appropriate screen-map location. The codes used for this purpose are not the standard ASCII codes which computers often use to represent typed characters, but a special set of screen codes that includes many characters not present in the standard ASCII character set. A complete list can be found beginning on page 132 of the *Commodore 64 User's Guide*, and on page 376 of the *Commodore 64 Programmer's Reference Guide.*

When a character is to be displayed in a given color on the C-64 screen, the screen map illustrated in Figure 1 and the color map illustrated in Figure 2 are used together. First, the desired character's screen code is stored in the appropriate memory location on the screen map.

www.commodore.ca

Then another code, which represents the color in which the character is to be displayed, must be stored in the corresponding memory location on the color map. Since both maps are exactly the same size—40 columns wide by 25 rows high—the color map can be thought of as a color overlay which can be placed on top of the screen map. Each rectangle on this color overlay can be displayed in 16 different colors—and when the code for a given color is stored in a given location on the color map, that is the color in which the character in the corresponding location on the screen map will be displayed.

The 16 colors that can be displayed on the C-64 color map—and the code numbers that are used to display those colors—are listed in Table 1.

TABLE 1

Commodore 64 Color Codes

Code Number	Color	Code Number	Color
0	Black	8	Orange
1	White	9	Brown
2	Red	10	Light red
3	Cyan	11	Gray 1
4	Violet	12	Gray 2
5	Green	13	Light green
6	Blue	14	Light blue
7	Yellow	15	Gray 3

Make your C-64 or VIC versatile with:

Custom Character Sets: CHARED for the C-64

Features:

- ☐ Menu Driven with full screen display
- ☐ Save and Load Character Sets
- ☐ Use with Multi-color Mode
- ☐ Built-in Utilities for reverse, copy and boldface characters
- ☐ Use with BASIC or ML
- ☐ Create Fonts or Game Graphics
- ☐ Keyboard or Joystick Input
- ☐ Cut and Paste Feature
- ☐ Specify CHARED

Sports: Mr. GOLF for C-64

Features:

- ☐ Course design and game programs
- ☐ Trees, Sandtraps, user designed Greens
- ☐ Built-in Adjustable Pin Placement
- ☐ Hi-Res ML Graphics for speed
- ☐ 18 hole course included
- ☐ Unlimited number of Courses
- ☐ Slice and Hook around Obstacles
- ☐ Specify Mr. GOLF (joystick required)

Word Processing: TexED for the C-64 and VIC-20

Features:

- ☐ Document editing: Create, modify, save tape and disk SEQuential data files
- ☐ Line image Editor using simple commands
- ☐ Full Screen Editor
- ☐ Print command with indentation and margin control
- ☐ Complete User's Guide included
- ☐ Specify: TexED for C-64. TexED20 for VIC-20 with 16K expansion

Graphics: PLOTVIC8 Hi-Res Graphics Kit for the VIC-20

Features:

- ☐ Menu Driven with Full Screen Display
- ☐ Keyboard or Lightpen Input
- ☐ Basic Geometry (point, line, ellipse) element-color selection, 3-D
- ☐ Screen to printer or plotter (Hi-Res)
- ☐ Functions stand alone for use with personal programming (BASIC)
- ☐ User's Guide-illustrated graphics tutorial and technical explanations
- ☐ Specify PLOTVIC8 (requires 8K+)

APCAD
P.O. Box 2673
Ann Arbor, MI 48106 USA

Mail completed form to **APCAD** P.O. Box 2673 Ann Arbor, MI 48106 USA. Send check for $ 19.95 (U.S.) for tape, $22.95 disk, or $34.95 for two ($39.95 disk). S & H included.

Name: _____ Specify: _____ CHARED
Address: _____ _____ Mr. GOLF
City/State/Zip: _____ _____ TexED
Credit Cards also accepted: _____ TexED20
☐ VISA ☐ MC ☐ AMEX _____ PLOTVIC8
Acct. No. Expir. Date: Signature:

Reader Service No. 261

The short BASIC program titled BALLBOUNCE.BAS (see page 124) illustrates how the C-64 screen map and color map work together.

BALLBOUNCE.BAS is not a high-resolution graphics program; it uses the C-64's standard 40-column text mode. In Line 40, a loop is used to place a red overlay across the top two thirds of the screen—from the top line down to the line that begins at Memory Address 55976. When this red overlay is first put in place, it is invisible, since nothing has been drawn yet on the screen. But as soon as something is printed on the portion of the screen covered by the overlay, it will show up in red.

After the red overlay is in place, a yellow one is laid down. This yellow overlay is just one pixel high; it runs across the screen horizontally. In the BALLBOUNCE program, this overlay is used to draw a yellow line: the line that represents a floor (or the ground) as a red ball goes bouncing across the screen.

Once the red and yellow overlays are in place, the words "FOLLOW THE BOUNCING BALL..." are printed in white across the top of the screen. Then, in Lines 210 through 240, the ball is animated. The animation technique is crude, but quite effective; PRINT commands are used to erase the ball and redraw it as it bounces across the screen.

A HIGH-RES GRAPHICS PROGRAM

A more sophisticated technique for creating screen motion is illustrated in the program titled BLACKBOARD.BAS on page 124. Like BALLBOUNCE, it is written in Commodore 64 BASIC. But, unlike BALLBOUNCE, it is a high-resolution graphics program. The BLACKBOARD program clears a section of memory that will be used as screen memory, and then it draws a pair of crosshairs on the screen using high-resolution graphics. But it does this job very, very slowly, clearly illustrating the snail's pace at which BASIC usually executes high-resolution graphics programs.

Here's how BLACKBOARD.BAS works:

The C-64, as pointed out earlier, has two primary screen modes: a text mode and a high-resolution graphics mode. In the latter, the 64 produces a screen display that measures 320 dots (or pixels) wide by 200 dots (or pixels) high. That's a total of 64,000 separate dots, each one of which requires one bit of memory. So it takes 8,000 bytes of memory to produce a high-resolution screen display.

When the 64 is in its high-resolution mode, it does not generate a screen display using ASCII characters stored on a 40-column by 25-row screen map. To create a high-resolution display, the 64 uses a screen map that occupies 8,000 bytes, or 64,000 bits, of RAM. Each of these bits can be individually controlled by the programmer. If a bit is on, the dot that it controls will be lit. If a bit is off, the dot that it controls will be dark.

High-resolution screen maps can be placed in various memory blocks (the reason this is true will be explained in a later column dealing with memory management). Colors are controlled by "overlays" in high-resolution

www.commodore.ca

graphics, just as they are on a text screen; the subject of screen colors will be covered later in this column.

Since one bit equates to one dot in high-resolution graphics, plotting the position of a dot wouldn't be difficult if the high-resolution screen were laid out using straightforward X/Y coordinates. Unfortunately, that is not how high-resolution screen plotting works on the C-64. Instead of being arranged as a matrix of dots 8,000 dots wide and 8,000 dots high, the 64's high-resolution screen is laid out exactly like a text screen: in a grid of rectangles 40 columns wide by 25 rows high. Within each rectangle are eight bytes of data, each sitting on top of another like pancakes. For example, this is what the letter "A" would look like displayed in high-resolution graphics in the upper left-hand eight-dot by eight-dot rectangle on a high-resolution screen:

TABLE 2

Bit-Mapping a Character in High-Resolution Graphics

Screen Location	Screen-Map Location	Binary Notation	Appearance
Line 1	Byte 0	00000000	
Line 2	Byte 1	00011000	XX
Line 3	Byte 2	00111100	XXXX
Line 4	Byte 3	01100110	XX XX
Line 5	Byte 4	01100110	XX XX
Line 6	Byte 5	01111110	XXXXXX
Line 7	Byte 6	01100110	XX XX
Line 8	Byte 7	00000000	

If another "A" were to be printed in the second position of the first row on a high-resolution screen—that is, just to the right of the screen position illustrated in Table 2—the bit pattern in the third column of Table 2 (labeled "Binary Notation") would be repeated in Bytes 8 through 15 on the high-resolution screen map. The next rectangle on the first row of rectangles on the screen would be controlled by Bytes 16 through 23 on the high-resolution screen map—and so on, all the way down to the bottom right hand corner of the screen.

This zigzag layout makes it easy to mix text and graphics on a C-64 screen, since text and graphics are laid out on the screen in exactly the same way. But it also makes dot-plotting somewhat complicated. To plot a dot on a C-64 high-res screen, you first have to figure out where the dot lies on a 320-dot wide by 200-dot high screen, using straightforward column and row coordinates. Then, since the C-64 high-resolution screen is subdivided into 8-dot by 8-dot rectangles, you have to break the screen down into a 40-column, 25-row grid by dividing each coordinate by eight. This division operation takes place in Lines 90 and 100 of BLACKBOARD.BAS:

```
90 CHAR=INT(HPSN/8)
100 ROW=INT(VPSN/8)
```

The next step in plotting the position of a dot is to figure out just where in the appropriate 8-dot by 8-dot matrix the desired dot lies. This calculation is carried out in Lines 110 and 120 of BLACKBOARD.BAS:

```
110 LINE=VPSN AND 7
120 BYTE=BASE+ROW*320+CHAR*8+LINE
```

Finally, you have to turn on the bit you have selected with a line such as this:

```
140 POKE BYTE,PEEK(BYTE) OR (2^BIT)
```

The above formula takes a long time to calculate in BASIC, and that is why BLACKBOARD.BAS runs so slowly. Next month, you'll get a chance to see how much faster the program runs in assembly language.

INITIALIZING HIGH-RES GRAPHICS

Now that you know how the high-resolution screen map works, let's look at the first part of the BLACKBOARD program and see what has to be done to put the Commodore 64 into its high-resolution graphics mode. The first statement in Line 20—BASE=2*4096—defines a constant which will be used later in the program to point to the starting address of a high-resolution screen map. This screen map will start at Memory Address 8192, or $2000 in hexadecimal notation. In the second statement

C™ COMMODORE-64

SOFTWARE-OF-THE-MONTH CLUB®

(division of O.C.S., Inc.)

We guarantee you'll never again buy useless software for your personal computer.

CLUB MEMBERSHIP OFFERS BIG SAVINGS

NO OBLIGATIONS

When you enroll you will

- Keep only the software that you can use. Examine for a full 10 days; if it's not what you want, return it for a full refund.
- Receive discounts up to 30% on your software choices.
- Be eligible for our Bonus Point Plan—additional discounts applied toward purchases.
- Receive our informative monthly newsletter full of helpful tips for getting the most from your Commodore-64.
- Receive notice of *Special Sales* where you'll save as much as 50% off list.

The Software-of-the-Month Club is Unique.
No minimum purchases are required—no automatic shipments to you.

Enroll now and receive *absolutely free* Public Domain Software.
Please check ☐ Cassette ☐ Disk

GREAT IDEA! I can't lose. Enroll me now in the Software-of-the-Month Club. I understand there is no obligation. Enclosed is my $10 membership fee.

☐ Check ☐ Money Order ☐ Bank Card

Name _____

Address _____

City/State/Zip _____

Visa/MC# _____ *Expiration date* _____

Signature _____

Cut out and mail today to

SOFTWARE-OF-THE-MONTH CLUB
Ohio Computer Services, Inc.
P.O. Box 128723
Cincinnati, Ohio 45212

3

Reader Service No. 262

www.commodore.ca

in Line 20—POKE 53272,PEEK(53272)OR8—the C-64's video interface chip (VIC) is told where to place its high- resolution screen map and where to find the data that it will need to display a high-resolution screen. The memory address used in this statement, 53272 (or $D018 in hex), is the address of a memory register referred to in C-64 documentation as VMCSB. When the C-64 is in its bit-mapped mode, the lower four bits of the VMCSB register are used to specify screen colors, and the upper four bits are used to point to the location of the bit map that will be used for a high-resolution display.

In Line 30, the VIC chip is instructed to go into high-resolution mode. This instruction is issued by setting Bit 4 of a memory register known in C-64 literature as SCROLY. One function of the SCROLY register is to implement fine scrolling—a capability we will not go into in this column. The BLACKBOARD program makes use of another capability of the SCROLY register: determining whether the C-64 will generate a text screen or a high-resolution display. If Bit 4 of the SCROLY register is set, the 64 will generate a high-resolution screen. If Bit 4 is clear, the computer will produce a text display.

In Line 50 of BLACKBOARD.BAS, Memory Addresses 1024 to 2023 ($0400 to $07E7 in hex notation) are stuffed with the value 16, which will cause a white line to be printed on a black background. When the C-64 is in its low-resolution graphics mode, the segment of memory from 1024 to 2023 is used to hold the Commodore ASCII codes that are used to print characters on the screen. But when the 64 is in its bit-mapped mode, this segment of RAM is used as an "overlay" that determines what background colors will be printed on the screen. In each byte of this block of RAM, the lower nibble is used to determine the color of any bits that are turned off in the corresponding rectangle on the screen, and the upper nibble to determine the color of any bits within the rectangle that are turned on. Since BLACKBOARD.BAS draws a white line on a black screen, the value 16—or $10 in hexadecimal notation—is stored in each byte of color RAM. Since 1 is the color code for white and 0 is the color code for black, storing the value $10 in each byte of color RAM will cause a white line to be drawn on a black screen.

Line 60 of the BLACKBOARD program is nothing but a jump back to Line 200. In Lines 200 through 240, a vertical line is drawn down the center of the screen using a bit-mapping subroutine that appears in Lines 80 through 250. This subroutine employs the plotting formula described earlier in this column to print white dots on a black background on the screen.

The line that is drawn down the screen in Lines 220 through 240 is two dots wide. That's because it takes a two-dot width to form a good solid line on a Commodore 64 screen; a line only one dot wide tends to show up pale and gray. The loop that draws this line appears in Lines 225 to 240.

After the vertical line is drawn, a horizontal line is mapped across the screen in Lines 245 through 280. Horizontal lines that are one dot high look fine in Commodore graphics, so this line is just one dot high.

The BLACKBOARD program ends with an infinite loop at Line 290.

When you run the BLACKBOARD program, you'll see how your computer clears the bit map that extends from BASE to BASE+7999, then changes the background color of the screen to black. Then, ever so slowly, you'll see your computer draw a set of crosshairs on your screen. Next month, you'll get a chance to see how much faster the BLACKBOARD program would execute if it were written in assembly language. Then you'll get an opportunity to type, assemble, and execute a program that will enable you to draw pictures with a joystick on a high-resolution screen! □

SEE PROGRAM LISTINGS ON PAGE 124

Put your Computer slave to work now!

SECTION ONE Instructions for finding the business best suited to you.

SECTION TWO The Menu—Condensed outlines for each business venture.

SECTION THREE Full details of each business venture . . .

SECTION FOUR Step-by-step instructions on how to organize each business venture.

SECTION FIVE How to advertise, promote and sell your services.

SECTION SIX How much capital is needed for each business venture, and how you can raise it quickly.

SECTION SEVEN Sources and Resources—Thousands of names, titles product sources, etc.

ABSOLUTE 90 DAY MONEY BACK GUARANTEE

—If you're not able to make money with your home computer and the information contained in our book! "COMPUTER ENTREPRENEUR" within 90 days, we'll refund the purchase price in full.

A 900+ PAGES 8½x11 SIZE 100 DETAILED MONEY MAKING BUSINESS PLANS

SOME OF THE 100 BUSINESSES THAT WILL MAKE YOU MONEY NOW!!!

Be an Economic Forecaster
Start a Computer Tax Service
Become a Computer Consultant
Unique Computer Reminder Service
Operate a Computerized Collection Letter Service
Start a Local Market Survey Service
Sell Computerized Legal Forms
Provide Computerized Loan Packages
Run a Payroll Preparation Service
Publish a Computer Newsletter
Enter the Sports Forecasting Business
Start a Secretarial Service
Run a Typesetting Service
Sell Local Mailing Lists at $1 Per Name
Start a Bookkeeping Service
Run a Custom Electronic Mail System
Start an Information Research Service
Operate a Legal Research Service
Sell Inventory Controls for Small Businesses
Start an Accounts Receivables and Statement-Mailing Service

FREE!!! SPECIAL REPORT 'HOTTEST COMPUTER BUSINESS FOR 1985' Is yours to keep when you order our book 'COMPUTER ENTREPRENEUR'.

WOULD A SECOND INCOME OF $1000 A MONTH . . . $2000 a month . . . $5000 a month come in handy?

If so, choose the amount that suits you and it's yours in exchange for a few hours of your spare time and use of your personal computer.

"THE COMPUTER ENTREPRENEUR"— Is the hot new book you may have read about. Over seven years in research and development—this book has 100 fully detailed business plans for you to choose from. They range from a traditional local bookkeeping service for small businesses to a far-out program for handicapping the horses that makes a substantial income—and 98 others in between.

"THE COMPUTER ENTREPRENEUR"—Is not another one of those "101 Ways" books with a paragraph or two of information about each business. Each business plan is fully detailed—how to start-up, how to operate, how to sell the service, how to keep the books, number of working hours required, capital needed, best location, city or county, etc. It even gives you a simple method of determining the best kind of business for you.

AND YOU DON'T HAVE TO BE A COMPUTER WHIZ—In order to start or operate these ventures. No programming or fancy special materials needed. All of these programs run from "off the shelf" programs you probably already have.

A 65 year old man in Washington State makes $500 a day with a TRS-80 model 1 and our book, "COMPUTER ENTREPRENEUR", using report #163.

A man and his wife in Arizona are making $4000 a month, with only a couple of hours a day, using the methods in our business report #102.

There are hundreds just like these making a living or extra income using the methods in our book.

CALL TOLL FREE FOR CREDIT CARD ORDERS 1-800-522-1500 EXT. 158

CHECK ONE
VISA ☐
MASTER CHARGE ☐
AM EXPRESS ☐
CARTE BLANCHE ☐
DINERS CLUB ☐

Name _____
Address _____
City _____ State _____ Zip _____
Please send me 1 book @ $29.95 + postage & handling ($2.50), total $32.45
(California residents add 6% Sales Tax)
☐ Air Mail $6.00 per book ☐ CASH, CHECK OR MONEY ORDER ENCLOSED
☐ Charge Credit Card #
Expiration Date _____ Signature _____

MAIL PAYMENT & MAKE CHECKS PAYABLE TO:
Computer Information Limited
1330 Orange Ave., Ste. 320
P.O. Box 181948
Coronado, CA 92118 (619) 266-9141

A3A1

Reader Service No. 256

www.commodore.ca

CREATED BY USERS FOR USERS

PROTECTION REVEALED

ALL C-64

Take command of your software. End the spiral of buying copy-programs. *The Protection Revealed Handbook* unravels the puzzle of software protection. For authors and users – become a master of illusion. Learn how to protect or un-protect your software. Includes a large section on successful BASIC program security. Covers the latest evolution of copy protection techniques. Topics include:

Disk Errors	De-Compilers	Extra Sectors	Nibble Counting	Header Modification
Compilers	Density Changes	Gap Bytes	DOS Protection	Non-Standard Sectors
Half Tracks	Nibble Back-Up	Guard Band	Auto-Loaders	And much, much more

Reveals the mystical professional trade secrets of the Commodore 64 and 1541 Disk Drive. A disk containing 21 utility programs is included. This is the book for everyone!

Book & Disk **$19.95**

THE PROTECTOR

A menu-driven collection of programs which promises to mystify the curious. Basic file *Encryption* and *Decryption* are the heart of this system. Also includes autoboot maker, write and un-protect disks, fast disk eraser, ML to data statements, sector editor, relocates files to new location in memory, un-new scratched files, plus much more.

A dozen plus programs – the perfect companion for the book, *Protection Revealed.*

Disk **$19.95**

SUPER HELP

This is the programmer's friend – *three* aids for the price of one – a must for all users:

1. Poster size (21x25) screen and color memory map. Hang on the wall and never hunt for that *poke* location again.
2. One of the most complete C-64 memory maps available today. Compact and easy to read. Instructions included.
3. Set of 18 cheat cards. Includes: 1541 disk drive, DOS wedge, error messages, dassette, 1525, MPS-801, 1526, screen codes, Kernal ROM, sprites, color, sound, conversion charts, Pokes & Peeks, etc. Spiral bound.

All Three Items For: **$15.95**

BRAND X — THE DISK LIBRARIAN

Has your disk library grown? Can't find your favorite program when you need it? Are you tired of odd labels? If you answered "yes" to any of the above, *you* need *Brand X. Brand X* can catalog 40-plus disks in one file – this is over 4000 programs. Save the catalog file, add to, change, or delete when you want. Get rid of duplication and save disk space. Identifies 14 types of files. Disk commands include: format (in 10 seconds), validate, scratch, change disk ID, change disk name, plus directory. Prints custom disk labels (great for the generic user), lists used ID's and disk names, catalogs in one, two, or three columns. End disk confusion with *Brand X, The Disk Librarian.*

Disk **C-128 Compatible** **$19.95**

PROGRAMMER'S KIT

Over 150 ready-to-run subroutines. Easy instructions on how to incorporate into your own programs. Why re-invent the wheel? Your time is valuable. Some of the types included are: TI$, protection, disk operation, printer operation, sorts, joyticks, input, graphics, complete alphabets, graphic screen prompts, sound, random choice. Program the easy way. Full documentation included.

C-128 Compatible $18.95

PHONE BOOK/POSTAL CLERK

A must for small businesses, clubs, schools, or individuals.

2,500+ ENTRIES

Menu-driven, compatible with all printers. Auto save entries and changes. Sorts and prints phone book or address labels by name, city, state, ZIP code, or group. Eight user-definable mail lists in one program. Keep groups, friends, clubs, Christmas lists, etc. all separate.

C-128 Compatible $24.95

ATTENTION GUN RELOADERS: BALLISTICS

Check trajectory for all popular rifle bullets by weight, caliber, or style. Includes tables for *Hornady* and *Speer* bullets, plus others. Allows for standard metro or high altitude and low temperatures. Prints out table of trajectory (choice of range increments), bullet drop, wind deflection, and more. If you reload your own shells, this program is a must!

C-128 Compatible **$39.95**

$-SAVER SWITCH

Use the *$-Saver Switch* to connect two computers to one printer, or one disk drive. Better yet, use only one drive and one printer for two computers. A flip of the switch changes the device from one use to the other. Save dollars where more than one computer needs to access the same device. Plugs in – no alterations or extras.

V-20, C-64, C-16, Plus 4, C-128 **$29.95**

Include check or money order – or charge your order to your VISA/MasterCard account.

Add $2 shipping on orders under $100 total. All orders over $100 will receive FREE shipping.

Your order will be shipped within 48 hours.

VALUE-SOFT

MasterCard VISA

9513 S.W. Barbur Boulevard, Dept. A-56
Portland, Oregon 97219 (503) 246-0924

Reader Service No. 290

www.commodore.ca

☆☆ NEW ☆☆ **Commodore - 64** ☆☆ NEW ☆☆

WORD PROCESSING BREAKTHROUGH!

SALE SALE

40 or 80 Columns in Color

Paperback Writer 64

40 or 80 Columns in Color

This is the easiest to use and most powerful word processor available for the Commodore 64. As you type on the screen, you will see your letters and words appear on the screen exactly as they will be printed (i.e. Italics will be *Italic*, Bold Face will be **Bold Face**). With the printer files you can customize Paperback Writer 64 to use all the fancy features of your printer. Loads EZ Script®, Paperclip®, & Wordpro 64® Files so you can easily upgrade your past wordprocessing text that you've written with obsolete wordprocessors. Take a look at some of the other features:

- **Wordwrap** — No Words Break At The Edge Of The Screen.
- **Flexible Cursor Movement,** Including Tabs And Other Timesavers.
- **Deletion And Insertion** Of Characters, Lines And Blocks Of Text.
- **On-screen Text Enhancement**, Such As Bold Face, Italics, Underlining, Superscripts And Subscripts, And Foreign And Other Characters.
- **Manipulation Of Blocks (ranges)** Of Text For Functions Such As Moving And Deleting, Even Between Files.
- **Sorting Lists** In Order Of Numbers And Letters.
- **Aligning And Adding Numbers** In Columns, Helpful With Tables.
- **Variable Margins** At Left And Right, And Paragraph Indentation.
- **Lines Centered, Justified** Or **Aligned** At The Right Side.
- **Variable Page Lengths** And **Line Spacing.**
- **Borders** At Top Or Bottom With Optional Title Lines And Page Numbers.
- **Linked Files** To Print Extra-long Documents In Sequence.
- **Flexible Printer Set-up** To Allow Use With Any Printer.
- **Find And Replace** Text Functions That Can Be Automatic.
- Complete Or Selective **Directories** Of Files On The Disk.
- **Sequential Files** For Mail Lists And Communication With Other Computers.
- **Spelling Checker**, Checks Your Spelling.

INTRODUCTORY PRICE

$29.95

COUPON

Sale Price $39.95

☆☆☆ **Plus** ☆☆☆

Full help screens on line with additional help on the disks mean you don't even need a manual. If you're in the middle of a page and you want to know how to use a special function just hit F7 and the information will appear before your eyes. If you still don't understand hit F7 again and a more detailed explanation appears. Then simply hit F8 and you're back in the letter where you left off. No manual lookup necessary. This is the easiest word processor in the world to use. List $99.00.
Introductory Sale Price $39.95. *Coupon $29.95.*

DOUBLE PROTECTO FREE TRIAL
We are so sure this is the easiest and most powerful word processor available, we will give you 30 days to try it out. If you are not completely satisfied, we will refund your purchase price.

PAPERBACK DATABASE (Datafax) List $69.00. **Sale $34.95.** * *Coupon $24.95.*

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% 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!
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. 293

EZ Script ® is a trademark of Commodore Business Machines. Paperclip ® is a trademark of Batteries Included, Inc. & Wordpro 64 ® is a trademark of Professional Software, Inc.

www.commodore.ca

CADET'S COLUMN

BASIC Programming for Beginning Commodore Computerists

Including *Compuloan*, a Loan Payment Calculator for the C-64

BY CHERYL PETERSON

This month I'm going to focus on BASIC. But even though this is a beginner's column, I am not going to degenerate into the standard explanations of PRINT, GOTO, GOSUB, OR, IF/THEN, and WHO CARES. There are many good books on BASIC programming, so who wants me to waste space saying it all again? If that isn't enough, Dale Rupert's column does a great job of teaching programming. Instead, let's take a look at some BASIC programming tricks and aids that will speed things up. I'll even throw in a program that I wrote. It's not fancy, but it works. Anyway, more on that later.

One great way to expedite BASIC programming is to cheat and use someone else's routines. Although it is illegal to steal programs that are copyrighted, many good programs are available in the public domain. (No, all those advertisements for "almost free" software are not rip-offs.) In fact, public domain software is a good source to build on. I intend to devote a whole column to public domain software later, but for now I'll just say that these programs are a good source of inspiration. They can also help you do very useful things.

One such program goes by the name *TX2BAS*. It takes a sequential (SEQ) file (like those created by many text editors) and converts it into a tokenized BASIC program file. Why would you want to do that? Although the Commodore screen-oriented BASIC editor is a vast improvement over the editors offered with other computers' versions of BASIC, it is still limited to the lines you can list on the screen. You don't have complete freedom of movement from top to bottom. With long programs, this can be a pain. Block moves, copies and deletes, along with global find and replace options, can really make programming flow faster. BASIC's editor just doesn't have them.

Text editors, on the other hand, offer such features, but the Commodore won't read "text" files as BASIC programs. *TX2BAS* solves the incompatibility problem.

Electronic bulletin board systems, accessed through modems, frequently contain public domain BASIC programs that need only slight modifications to run on the Commodore. Unfortunately, these programs are usually stored as ASCII text files and although most terminal packages (modem software) will translate ASCII into PET ASCII, they won't translate SEQ files into tokenized BASIC files. Once again, *TX2BAS* solves the problem. There are several versions of it, the latest being *TX2BA4*.

Where can you get this wonder program? In addition to its presence on the following page, it can be found on CompuServe in the Commodore Beginners Special Interest Group (GO CBM 963) in Data Library 2. Type BRO TX2BA4.SEQ at the DL2: prompt, if you are using a terminal program that doesn't support CIS protocol. Once the file is found for you, choose the R prompt to READ the file into your buffer. You'll need to save the program to disk or tape to use it. If you are using Vidtex or some other package that can download .IMG files, type BRO TX2BA4.IMG. When the file has been located, use the D command to DOWNLOAD the file. Your terminal program should automatically save the file for you.

A document file, *TX2BAS.DOC*, explains how to use the program. I'd recommend downloading this file, because it has a useful description of how to use *TX2BAS* to merge programs together—a very handy feature if you have a library of routines that you like to use regularly. You can append the routines at the end of your program instead of typing them in repeatedly.

For those who don't have CompuServe protocol compatible downloads, you'll have to use the BASIC editor to type *TX2BAS*. Once you have it in runable form, you need only follow the prompts to convert your files from SEQ to PRG. It only requires the name of the file to convert.

I must warn that my version is slightly different from the CIS program. The doc file mentions that once you've downloaded theirs, you must edit it slightly. In the original version, two lines that were included as remarks actually controlled whether *TX2BAS* sought the SEQ file

www.commodore.ca

TX2BAS

Letters on white background are *Bug Repellent* line codes.
Do not enter them! See page 122 for instructions.

```
·62990 A=PEEK(61)+256*PEEK(62)+3:POKE786,
 INT(A/256):POKE785,A-256*PEEK(786)          CC
·62995 IFERTHENPOKEA-2,0:POKEA-1,0:POKE45
 ,PEEK(785):POKE46,PEEK(786):CLR:END         GN
·63000 POKE53280,0:POKE53281,0:POKE646,1     AG
·63009 PRINTCHR$(147)                        FG
·63010 PRINT "THIS PROGRAM WILL TURN TEXT
  BASIC FILES INTO RUN BASIC PROGRAMS";      GC
·63020 PRINT "USING THE COMMODORE DATASET
 TE OR DISK DRIVE."                          CF
·63030 PRINTCHR$(17)"THE PROGRAM WILL DEL
 ETE ANY LINE"                               JN
·63040 PRINT"OF THE FILE IN CASE IT IS NO
 T A PROGRAM LINE."CHR$(147)                 ED
·63043 INPUT"NAME OF FILE TO BE CONVERTED
 ";F$                                        DJ
·63045 PRINT "[4" "]"CHR$(18)"T"CHR$(146)
 "APE OR "CHR$(18)"D"CHR$(146)"ISK?"         MI
·63046 GETA$:IFA$=""THEN63046                PP
·63047 IFA$="T"THEND=1:SA=0:N=1:GOTO63080    EG
·63048 IFA$<>"D"THEN63046                    KO
·63050 D=8:SA=3:N=3                          FF
·63080 PRINT "HIT A KEY WHEN READY!"         CI
·63085 GETA$:IF A$=""THEN63085               PI
·63090 PRINTCHR$(147)                        FG
·63097 OPEN3,D,SA,F$:PRINT"READING FILE,
 PLEASE BE PATIENT!":FORX=1TO1000:NEXT       AM
·63099 POKE152,3:T$=""                       EJ
·63100 GET#3,A$:IF A$=""THEN63100            KC
·63105 IF ST=64THEN63250                     AG
·63106 IF A$=CHR$(13)THEN63130               KF
·63110 T$=T$+A$                              MK
·63115 A$="":GOTO63100                       EF
·63130 IFLEN(T$)<3THENT$="":GOTO63100        LN
·63140 IFLEFT$(T$,1)=CHR$(10)THEN63250       BM
·63142 IFVAL(LEFT$(T$,1))=0THENPRINTT$:T$
 =RIGHT$(T$,(LEN(T$)-1)):GOTO63140           OO
·63220 PRINTCHR$(147)"[6" "]TRANSLATING T
 EXT INTO BASIC[3"."]"                       HG
·63225 PRINTCHR$(17)CHR$(17);T$:PRINT"GOT
 O 63099"                                    LL
·63235 POKE198,2:POKE631,13:POKE632,13:PR
 INTCHR$(19):END                             EA
·63240 IFVAL(LEFT$(T$,1))=0THENT$=RIGHT$(
 T$,(LEN(T$)-1)):GOTO63140                   KB
·63250 PRINTCHR$(147)"CONVERSION COMPLETE
 [3"!"]":CLOSE3:PRINT                        EE
·63260 PRINT"DO YOU WANT THIS CONVERTER"      NP
·63262 PRINT "DELETED BEFORE YOU SAVE THE
  NEW PROGRAM?"                              HG
·63265 GETA$:IFA$=""THEN63265                PM
·63266 IFA$="Y"THENER=1:GOTO62990             EI
·63270 PRINT"ALL DONE! CHECK IT OUT!"         MP
·63280 END                                   IC
```

on tape or disk. *TX2BAS* required removing the REM at the beginning of the line you wanted to use. My version asks whether the SEQ file is on tape or disk. It adds an extra keystroke, but gives more flexibility.

CompuServe's latest version, *TX2BA4* looks for the SEQ file on the device from which it was loaded. Both my version and *TX2BA4* required only a few lines of changes to make them easier to use.

This is just the type of thing you should think about doing: taking an already written program and modifying it to make it do what you want. Why duplicate someone else's work if they choose to give it away?

FIGURING LOAN PAYMENTS

Although there are several programs available that will compute the payments for a loan, I decided to write my own. I used a formula for calculating loan payments from a general BASIC manual. I wrote the necessary GET statements and prompts to put the numbers in the formula, tried out several routines for rounding numbers down to two decimal places, wrote some print formatting routines, and added a couple of error trapping lines just to keep things from getting messy. Sounds easy, doesn't it? It took two days!

Even when you have a pretty good idea of what you want to do, problems crop up that you don't expect. If you're like me, you do a minimum amount of planning and jump right in with both feet. As you can see from the line numbers in my program listing, things weren't always as simple as I thought they'd be. And then when I really got moving I thought of a couple of neat additions to make the program nicer.

Earlier, I mentioned borrowing routines. Well, I went through three different ones to round numbers to the second decimal place. I finally settled on a one-liner, but at one time the routine at 2000 had eight lines containing two separate routines. As it turned out, the two of them just refused to work together. I wasted about five hours before I finally got through that section.

You see, programming is an inexact science. Actually, it's more of an art. Programming languages come in various shades, with peculiarities bound to each. And even within the same language, there may be many ways to accomplish the same end. BASIC is a flexible medium. As in painting, there is no "correct" technique.

Programs that you write for your own use do not have to be perfect! If they work for you, that's what counts. But there is always someone out there who just can't resist trying to fix a program up a bit. And that is what the public domain is all about. By putting your program out there, where people can see it, someone may just perfect it for you. The changes in *TX2BAS* are a "perfect" example.

Of course, you'll have to suffer the slings and arrows of those who think every program must be a work of art. Proponents of "structured programming" may well use your handiwork as an example of how *not* to write programs. And the ten year old down the block may laugh hysterically when he finds out the program wasn't writ-

www.commodore.ca

NEW LOW PRICES FROM THE WORLD'S LARGEST SPORTS SOFTWARE COMPANY!

Whether you're a coach, player, computer bug, statistician, dabbler, doer or just a real sports fan, PDS has a package for you...at our lowest prices ever! Check our line-up below, and then order your first PDS Sports Package.

FOOTBALL STATISTICS. A comprehensive computer software system. You compare teams in over 60 offensive and defensive situations and watch the matchups on the screen. Every team is given a power rating. PDS also provides "year-to-date" data diskettes for a nominal charge. 39.95

FOOTBALL HANDICAPPING. (Includes FOOTBALL STATISTICS Package). Forecasts the outcome of football games based on massive amounts of statistical data, morning lines and point spreads. 49.95

BASKETBALL STATISTICS. The most comprehensive basketball program ever written. "What-if" your way into every imaginable comparison and formulation. 39.95

BASKETBALL HANDICAPPING. (Includes BASKETBALL STATISTICS). Gives you an inside look into the outcome of games by mathematically equating statistics, lines and point spreads. 49.95

BASEBALL STATISTICS. See how opposing baseball teams stack up against each other. See summarizations. Performance stats on every team in the major leagues. 39.95

BASEBALL HANDICAPPING. (Includes BASEBALL STATISTICS) Makes you better-prepared to predict the winner of any game. At least, mathematically. 49.95

THOROUGHBRED HANDICAPPING. This "world class" program gives you an edge in predicting winners. A proven system...by PDS and many horseracing enthusiasts. 129.00

HARNESS RACE HANDICAPPING. The fast, easy way to find out what races and what horses should be looked at...all summarized in a "power rating" format. 129.00

QUARTER HORSE HANDICAPPING. the computer system actually "rates" the horses in each race of what is the fastest growing segment in the sport of kings. 129.00

TRAINER STATISTICS. Analyzes all local racehorse trainers and gives you a rating for each one. 39.95

JOCKEY STATISTICS. A short time with this software package will show you why the top 15 jockeys at local tracks win over 90% of all races. 39.95

HARNESS DRIVER STATISTICS. A complement to the HARNESS HORSE HANDICAPPING System, this package tells you all you need to know about the man (or woman) *behind* the horse. 39.95

PDS SPORTS™

P.O. BOX E / TORRANCE, CA 90507 / (213) 516-6688
Please send me the following PDS SOFTWARE PACKAGE:

_____...$_____
_____...$_____
_____...$_____
Total

Add $6.00 for postage and handling.
Calif. residents add 6½% sales tax
I have an APPLE ( ) IBM ( ) TRS-80 ( ) COMMODORE-64 ( )
DISKETTE ( ) CASSETTE ( ); MODEL # _____
( ) Check enclosed
( ) Charge my credit card: A.E. ( ), VISA ( ), M.C. ( )
Card No._____Expires_____
Signature_____
(As it appears on credit card)
OR CHARGE BY PHONE...CALL (800) 222-2601
(In Calif: (213) 516-6688)

NAME_____
ADDRESS_____PHONE_____
CITY_____STATE_____ZIP_____

Dealer Inquiries Invited

Reader Service No. 134

www.commodore.ca

ten by a nine year old. But, every time you get a program to do what you want it to, you're learning or creating a new technique that works for you. And eventually, you'll feel comfortable saying, "Yeah, I do a little programming now and then. Why? What are you working on? Maybe I can help."

By the way, I know there are loan programs in the public domain, but since I'm writing about programming I thought the least I could do was put something original in my column.

For the education of those who care, I'll try to explain what the program does and how it does it. For those of you who couldn't care less about programming, skip to the "Future Columns" header. Unless you need this program to figure out how much new car you can afford to buy and still make the payments! Or maybe you're looking for a new house?

Are you sure you're going to want that $76,000 beauty when you find out that you'll pay $177,609.97? That's $101,609.97 in interest at 13.5% over 15 years, $986.72 every month. The monthly payment comes down, but the interest doubles, if you extend the payments over 30 years.

My program takes the amount you plan to finance, the number of payments per year, the total number of payments, and the interest rate, and spits out the size of individual payments, the total amount you'll be paying back, and how much of that is interest. It will also print out a payment schedule showing the balance owed, the payment made, how much of that payment is interest, and how much applies toward the principal. This can be a long list when you're thinking about a 30-year mortgage.

As mentioned before, I don't use a very structured style. The opening lines identify the program. Then, a couple of GET statements are used to determine whether the screen or printer is used for output and whether the payback schedule is generated or not. (More on these later.) INPUT statements are then used to get the numbers to feed into the equations. The GOSUB to 1000 is an error trap to prevent using negative numbers. By taking the absolute value, the negative is stripped off.

"I programmed it to wake me up if the boss comes in."

Lines 250 and 251 check to see if you want a printout and open a channel to the printer, if appropriate. Line 260 clears the monitor screen. I chose to have the PRINT statements reprise the information that has been entered, so that the upper section gives all the details; amount financed, interest rate, payments per year, and how many payments total.

By doing it this way, it's convenient to refer to them later. (Like when you're trying to convince your husband that you really can afford that new washing machine. You'll have to write your own program to show how the thing will pay for itself by saving on the laundry bill.) The printouts are also helpful when doing comparison shopping. One dealer may give you a better price, another a lower interest rate, and a third a longer term loan. Comparing the printouts may help you see which is the best deal and which deal you can afford.

Line 275 sends the amount financed figure off to be rounded by the routine at 2000, via line 4030. I had two other rounding routines, one a three liner, another six lines long. I chose this one because it's short, sweet, and it works. Lines 400 and 410 actually compute the individual payments and amount of interest paid over the life of the loan.

The subroutines at 3000 and 4000 take the active variables and reassign them to the variable B to be rounded by GOSUB 2000. When the result of the truncation comes back, it has to be reassigned to its old variable for printout.

Lines 600-640 give the results of the calculations. Line 690 checks to find out whether you want a loan schedule printed or not. If not, it sends execution to the END statement.

Lines 691, 695, and 697 calculate the values for the first line of the loan schedule. Lines 700 and 760 control the output format. Since the screen is only 40 characters wide and most printers are 80 characters wide, I set the program to print out in two different ways. Although the screen format would work with the printer, I wanted to take advantage of the printer's wider line length. (Add an extra three hours of programming time to get the printout right!)

The X variable is used to keep track of how many lines have been printed. The X2 variable tracks what page is printing. The first page has 50 lines of loan schedule, as the first 10 lines are used to print the loan information. The other pages have 60 lines. Lines 730 and 740 use the X variables to track the lines and pages that have been printed. CHR$(127) is a form feed. When the printer receives one, it will page over the perforations between sheets of paper. (This assumes you're using a tractor feed printer.)

For those who use a printer that takes cut sheet paper and want a challenge, try writing in a couple of routines to pause at the end of pages. You'll need to use a routine that takes keyboard input to restart the printout.

Lines 750-753 are used to get the numbers lined up on the decimal point. Again, this is a borrowed routine someone else submitted to a magazine for others to use.

www.commodore.ca

presenting . . . CAPTURE™

A NEW WAY TO UNLOCK THE POWER OF YOUR C64 OR C128*

- **CAPTURE** IS A CARTRIDGE THAT PLUGS INTO YOUR COMPUTER'S EXPANSION PORT.
- **CAPTURE** DOES NOTHING — UNTIL YOU PRESS ITS **CAPTURE** BUTTON. UNTIL THEN, A RUNNING PROGRAM CANNOT DETECT ITS PRESENCE.
- **CAPTURE** THEN TAKES CONTROL — NO IFS, ANDS OR BUTS — AND PRESENTS A MENU.
- **CAPTURE** WILL NEATLY SAVE EVERYTHING IN YOUR COMPUTER TO YOUR 1541 OR COMPATIBLE DISK DRIVE — ALL 64K OF RAM, CPU, VIC AND SID CHIP REGISTERS — EVERYTHING. IN EASY TO VIEW CHUNKS.
- **CAPTURE** WILL, IF YOU WANT, PRE-CONFIGURE YOUR COMPUTER'S RAM SO THAT ONLY MEMORY ALTERED BY YOUR PROGRAM NEED BE SAVED.
- **CAPTURE** WILL WRITE A BOOT ON YOUR DISK SO YOU CAN RELOAD AND BEGIN EXECUTION RIGHT WHERE YOU LEFT OFF.
- **CAPTURE** DOES ALL THIS AT A COST OF JUST **$39.95**

BUT HERE'S THE BEST PART

- **CAPTURE** WILL MAKE AN AUTO-START CARTRIDGE OF YOUR PROGRAM. IT'S EASY! JUST FOLLOW THE DIRECTIONS ON THE SCREEN. NOW PLUG IN YOUR CARTRIDGE AND TURN ON YOUR COMPUTER. IN LESS THAN TWO SECONDS YOUR PROGRAM BEGINS AGAIN AT PRECISELY THE POINT WHERE YOU **CAPTURE**'D IT. MAGIC!

  BESIDES **CAPTURE,** YOU NEED A *promenade* C1 AND A SUPPLY OF CPR3 CARTRIDGE KITS.

ORDERING INFORMATION

- **CAPTURE** CARTRIDGE — COMPLETE WITH INSTRUCTIONS ..... $ 39.95
- *promenade* C1 — EPROM PROGRAMMER WITH DISK SOFTWARE 99.50
- CPR3 CARTRIDGE KIT — PC BOARD, CASE AND 3 EPROMS 29.95
- DR—EPROM ERASER, TWO AT A TIME, 3 TO 10 MINUTES 34.95
- STARTER SET — **CAPTURE,** *promenade* C1 AND 1 CPR3 ..... 149.95
- DELUXE SET — **CAPTURE,** *promenade* C1, DR AND 2 CPR's 199.95

SHIPPING AND HANDLING — USA: UPS SURFACE $3.00
BLUE LABEL $5.00 NEXT DAY AIR $13.00
CANADA: AIR MAIL $7.00 OTHER FOREIGN AIR $12.00

CALIFORNIA RESIDENTS ADD APPLICABLE SALES TAX
COD ORDERS, USA ONLY, ADD $3.00

C64 AND C128 TM COMMODORE ELECTRONICS, LTD. *WHEN OPERATING IN 64 MODE

TO ORDER: TOLL FREE 800-421-7731
FROM CALIFORNIA 800-421-7748

TECHNICAL SUPPORT AND 408-287-0259
FROM OUTSIDE THE US: 408-287-0264

JASON-RANHEIM
580 PARROT STREET
SAN JOSE, CA USA 95112

VISA MasterCard AMERICAN EXPRESS

Reader Service No. 131

www.commodore.ca

Next is a perfect example of sloppy programming. (One of those where a picky person would start getting critical and recommend a course in structured programming.) Line 770 duplicates the calculations performed by lines 691, 695, and 697. A structured programmer would have stuck them in a subroutine at the end. But I didn't realize I would need them more than once until I started working out the page formatting. And I was too lazy to rewrite eight lines of code. (Doesn't that make you feel better? Just to know that someone else is lazy, too?)

820 tidies up the loose ends at the bottom of the printout. (After all, the one payment at the end usually turns out to be less than the rest.) 825 checks to see if the schedule is complete. If so, it ends the program. Otherwise, the program loops back to print the next line of the table.

That's it, folks. It isn't pretty. It isn't fancy. It just works!

There is usually a hidden point to my meanderings. The obvious purpose of this exercise was to show a little bit about BASIC programming. But the covert messages are that anyone can program, you don't have to be a perfectionist, and you learn by doing.

Many programs you see in magazines look so clean and neat and tidy because they've been done by someone who has been programming for quite a while. (Or they've been edited by someone equally adept.) But programs don't have to be pretty to be functional. And when you first start programming, the results aren't likely to be pretty. (Actually, they probably won't be functional either!) With practice, though, the pieces start to fall into place.

Once you've decided to jump into programming, there are a few aids that can make it go faster. *Ahoy!* has already published a program that generates line numbers for you and another that renumbers them. Though there are extended versions of BASIC available, beginners should probably stick with the basic BASIC until they've gotten a bit more experience.

Two commercial programs will help speed things up for you: *XREF 64* and *BASIC 64* from Abacus Software.

X-REF 64 is an inexpensive little program that analyzes your program and prints a list of all keywords, functions, and variables used. It gives a nicely formatted, paginated listing of all the numerical values, variables, and BASIC keywords, as well as the lines in which they appear.

When programming in BASIC, it is often difficult to remember just where everything is in a program. If the program doesn't have comments identifying sections where specific processes take place, it is especially tough. Most programmers use variable names that correspond to the functions or ideas represented by the variables. While this helps make deciphering printouts easier, trying to find a given variable in an 800-line program is like trying to find a dime in a pile of nickels. When you have a listing of all the variables and the lines in which each appears, it's easier.

Complaints that BASIC programs run slowly are well founded, and when you can buy an inexpensive compiler that will improve the speed at which your programs run, it's a shame not to have one. *BASIC 64* is such a program. Depending on how often you use a program you've found or written, it may or may not be worth compiling. A long program can take up to 30 minutes to compile, but once compiled it will run much faster. The compiler won't speed up output to the screen or to a printer or modem. It will speed up internal calculations and processing that the computer does.

Castle, one of my favorite public domain games, takes a couple of minutes to assign variables and manipulate data when the game starts. Of course, it displays a short text explanation of the game while it is doing all this, but the wait is annoying. Enter *BASIC 64*. It took about 20 minutes to compile *Castle* and now the initialization wait is only about 15 seconds. Although there is a noticeable improvement in the opening segment, the screen display is still slow since BASIC doesn't use direct screen addressing.

I should mention that *Castle* was originally written for a TRS-80 computer running MicroSoft BASIC. I originally downloaded it from a public BBS with my Osborne 1 computer (the machine I used before I got my Commodore). It required only small modifications to run. I later used the Commodore RS232 module and a null-modem (direct connect) cable to transmit *Castle* to the Commodore as an ASCII file. After using *TX2BAS* to convert it to a PRG file, I was able to modify it to run on the Commodore. Although all three machines run MicroSoft BASIC, each version is slightly different (just as BASIC 7.0 for the 128 is different from the 4.0 than runs on the 64).

WHO WILL BENEFIT FROM *BASIC 64*?

For those who find BASIC programs that they use consistently, *BASIC 64* would be a great asset. It's surprisingly easy to use. The documentation is concise; simple explanations for the simple features. Although the program has advanced features that more experienced programmers will find useful, it is designed to be usable by programmers of any experience level. Using the advanced features, assembly language routines can be included and modules can be strung together.

The resulting compiled programs usually take up just as much (if not more) disk space, but the speed makes up for it. While other compilers cost close to $100, *BASIC 64* costs $39.95. And as you get more heavily into programming, it expands to suit you.

BASIC 64 is available for $39.95 and *X-REF* for $17.95 from Abacus Software, P.O. Box 7211, Grand Rapids, MI 49510 (phone: 616-241-5510).

FUTURE COLUMNS

If any of you have a particular topic you'd like to hear more about, please write me in care of *Ahoy!* All your letters will be forwarded to me, here in Miami. For any of you who'd like to contact me via CompuServe, my user number is 72366,2645. Since I'm only an occasional visitor to the CBM SIG's, EasyPlex is the best way to get in touch. □

SEE PROGRAM LISTING ON PAGE 135

www.commodore.ca

SPECIAL OFFER
$7.00
(NORMALLY $19.95)

Seeing Is Believing

"I don't have enough time or space to list all the good points!" -- ***Noland Brown, MIDNITE SOFTWARE GAZETTE***

"This disk is fantastic!" -- ***Tom Lynch, THE USERS PORT***

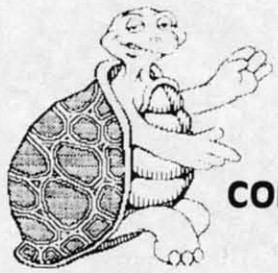
"Why all the enthusiasm? Because **COMAL** is a composite of the best features of the most popular programming languages... the familiarity of BASIC commands with the structural programming environment of Pascal and the turtle graphics of Logo." -- ***Mark Brown, INFO 64***

"**COMAL** was just what I was looking for." -- ***Colin Thompson, RUN***

Seeing ***is*** Believing. Take a look at what **COMAL** has to offer:
the complete **COMAL 0.14 System** for Commodore 64™ includes the
Tutorial Disk* *(teaches you the fundamentals of COMAL)*, plus
the **Auto-Run DEMO Disk*** *(demonstrates 26 COMAL programs including games, graphics, sprites and sounds)*,
all for just **$7.00!**
You can add the reference book, *COMAL from A to Z*,
for just **$4.00 more**.
$7 or **$11** -- either way you're a winner!

"Everybody who gets it, likes it! (I'll guarantee it.)" -- ***Len Lindsay, President, COMAL Users Group***

Call TOLL-FREE: 1-800-356-5324 ext. 1307
VISA or MasterCard Orders ONLY.
Questions and information must call our Info Line: 608-222-4432.
All orders prepaid only – no C.O.D.



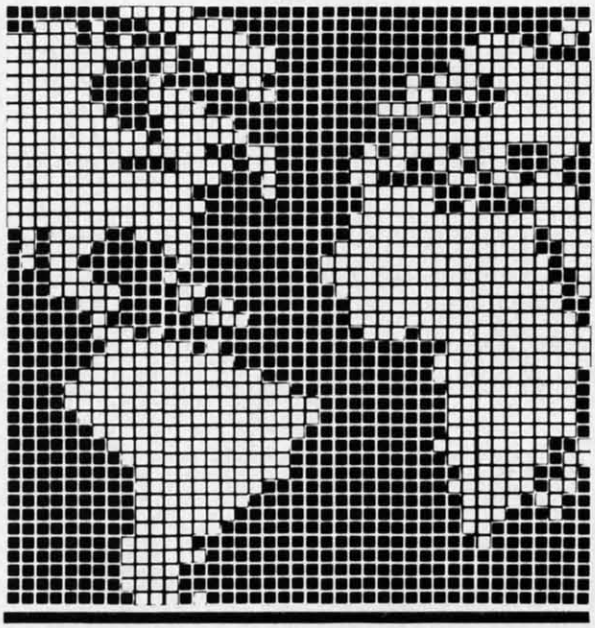
Send check or money order in US Dollars to:

COMAL USERS GROUP, U.S.A., LIMITED
6041 Monona Drive, #111, Madison, WI 53716
phone: (608) 222-4432

*Programs will come on 2 disks or 1 double sided disk -- each disk includes COMAL.
Commodore 64 is a trademark of Commodore Electronics

www.commodore.ca

THE WORLD OF COMMODORE III



The 1984 Canadian World of Commodore show was the largest and best attended show in Commodore International's history. Larger than any other Commodore show in the World and this year's show will be even larger.

World of Commodore III is designed specifically to appeal to the interests and needs of present and potential Commodore owners.

Everything about your present or future Commodore computer – from hardware to software, Business to Personal to Educational – from over 90 International Exhibitors. Price of admission includes free seminars, clinics, contests and free parking.

PLAN TO EXHIBIT

INTERNATIONAL CENTRE

DECEMBER 5 TO 8 '85

TORONTO

A HUNTER NICHOLS PRESENTATION

To exhibit contact:
Hunter Nichols Inc., 721 Progress Ave., Scarborough, Ont., M1H 2W7
(416) 439-4140

Reader Service No. 287

www.commodore.ca

COMMODARES
PROGRAMMING CHALLENGES

By Dale Rupert

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

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

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

Programs on diskette 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 #23-1: EDGE WEDGE

This problem was submitted by Charles Grady (Cleveland, TN). The user inputs a string. The computer prints the string as a wedge against the right side of the screen. An example gives the details: if the input is "A TEST", the output is

```
     T
    ST
   EST
  TEST
  TEST
A TEST
```

One restriction—no cursor arrow keys are allowed. The output should be displayable on the screen or on a printer.

PROBLEM #23-2: ROOTING ROUTINE

Allan Flippin (San Jose, CA) suggested this problem for the mathematically inclined. The user inputs a number between 1 and 65535. The computer displays the square root of the number to the nearest whole number. Of course there's a catch. The routine must be in BASIC and must use only +,−,/,*, and relational operators (no BASIC functions such as SQR or LOG). Dig out those algorithm books.

PROBLEM #23-3: SPEEDY BUG

This classic problem suggested by James Borden (Carlisle, PA) should make a good *Commodare.* Two trains head toward each other on the same track. Train A goes 62.5 mi/hr and Train B goes 37.5 mi/hr. The trains are 800 miles apart when a fast bug leaves Train A, flies to Train B, immediately turns around and flies back to Train A, whereupon it instantly turns and returns to Train B, et cetera ad infinitum, or at least until the trains meet, smashing the bug between them. (Please substitute a less violent scenario if you prefer.) How far does the bug fly?

Already some of you are writing down the answer and preparing to send it to us, but wait. What we want here is a computer simulation, not just the right answer. Pretend that the answer is not easy to calculate (of course it is if you know how), and let the computer figure it out. (Any graphically animated solutions should be sent on tape or disk.)

PROBLEM #23-4: TYPING TUTOR

The computer displays a sentence. The user types the sentence. The computer then displays the number of errors the user made and the amount of time in jiffies it took to type the sentence. Nothing too difficult or tricky here. Let's see some solutions from you first-time programmers.

This month we will look at readers' solutions to *Commodares* from the July issue of *Ahoy!* There are a few other items of business to handle as well. First a special word of thanks to the readers from other countries who sent solutions and letters to *Commodares.* In addition to the Canadian readers mentioned later, we have received "international solutions" to various problems from Yilmaz Atila (Izmir, Turkey), Christian Leyer (Quakenbrueck, Federal Republic of Germany), and Ricardo Chan (Panama City, Panama). Readers in other countries should feel free to write, whether you have solutions to *Commodares* or not. It is a pleasure to hear from all of you.

Tony Ruperto (Kitimat, BC, Canada) succinctly answered the question posed in a previous *Commodares* column as to when a space is mandatory in a BASIC program. He states that the only time a space is needed is when the combination of two words or letters make up a third which is a reserved word or another BASIC keyword. For example, A=TAND128 requires a space between the T and the AND or else the computer will

www.commodore.ca

assume that the tangent function is desired and give a syntax error. Bugs like that can be quite puzzling at first.

Joseph Taylor (Jenison, MI) was the first to send the translation of the cryptogram listed at the end of the August *Commodares*. Joseph solved it without help of a computer in 45 seconds. It was a one-letter displacement code where "ABC..." were replaced by "BCD...". I would still like to hear from someone who has a program to help arrive at the result based on standard letter frequencies. This is a difficult problem with such a small sample of encoded text, but it might be fun to try.

In response to the challenge for a significant-digit rounding function, Michael Skloff (New York, NY) sent the following:

```
DEF FNL(N)=INT(LOG(ABS(N))/LOG(10))
DEF FNS(X)=INT(N*10^(X-FNL(N)-1)+.5)*10^
(FNL(N)-X+1)
```

These functions properly round the number N to X significant digits. The first function calculates the base−10 logarithm of N. This is used to gauge the size of N. The second function performs the actual rounding. The 80-character line limit is the only reason for having two functions. The variable N must store the number to be rounded. Then PRINT FNS(3) will display the value of N rounded to 3 significant figures. (Unfortunately 1.235E+15 rounded to 3 significant figures resulted in 1.23E+15 instead of 1.24E+15, but perhaps that is because of the internal storage of the values. Can any readers provide further insight into this problem?) Thanks to Michael for these functions.

Thanks also to Scott Duncan (Superior, NE) and John R. Prager (Bay City, MI) for detailed analyses of the "N elements printed in C columns" problem presented in *Commodares #19-1* in the July issue. John Prager summarizes the situation as follows: assume that R is the remainder when N is divided by C. If R is greater than zero but less than C−1, it is not possible to display N numbers in C columns as described in the problem. If anyone is interested in the proofs, send a self-addressed stamped envelope to *Commodares* at the above address.

One final tidbit before looking at July's problems. Charles Grady (Cleveland, TN) sent the following program in response to the cycling function challenge *(Problem #18-1):*

```
10 FOR I=0 TO 100 STEP 5:GOTO 30
20 FOR L=95 TO 5 STEP -5:GOTO 40
30 PRINT I:NEXT I:GOTO 20
40 PRINT L:NEXT L:GOTO 10
```

First let me say that the program works as advertised on the Commodore 64. The reason I have listed the program is that before I entered it into the computer and ran it, I would have sworn that it wouldn't work. Do you know why? (Study it for a moment if you are a long-time BASIC programmer.)

Back in the old days, rule number one regarding nested FOR-NEXT loops was that they must not overlap. It is okay for one loop to be contained completely within another, but the inner loop must not meander outside of the outer loop as in this example. Well, so much for the old days. The C-64 handles this program without any trouble. Out of curiosity, I tried the program on an IBM PC. It gave a "NEXT without FOR in line 40" error before it reached line 20. If any of you have other computers to try this on, let me know your results. How about the C-64 BASIC Compiler? What happens if you use the NEXT statements without variables? Thanks to Charles for an interesting problem.

Now on to July's *Commodares. Problem #19-1: Binary Palindrome* brought dozens of responses. There were several significantly different categories of solutions. For a binary number to be a palindrome, it must be a sum of one or more of the following values: 129, 66, 36, and 24. This is easily seen by writing the binary values for those numbers. Each number contains two symmetrically placed ones. The solution from Mark Tillotson (Tulsa, OK) typifies the approach that looks for such terms within a given number.

```
1 REM
2 REM PROBLEM #19-1:
3 REM     BINARY PALINDROME
4 REM BY MARK TILLOTSON
5 REM
10 INPUT"VALUE";N:IF N<0 OR N>255THEN10
20 IF N>=129 THEN N=N-129
30 IF N>=66 THEN N=N-66
40 IF N>=36 THEN N=N-36
50 IF N>=24 THEN N=N-24
60 IF N=0 THEN PRINT"YES":PRINT:GOTO 10
70 PRINT"NO":PRINT:GOTO 10
```

Mark's trial subtraction essentially "crosses out" symmetrical pairs of ones in the binary representation of the given number. If anything is left over, line 70 concludes that the number does not have a binary palindrome.

Another approach to this problem is to perform bit-by-bit analysis on the given number. If the number is a palindrome, the seventh bit must match the zeroth bit, bits six and one must match, and so forth. Most solutions using this approach contained statements such as

```
IF (N AND 64)/64 <> (N AND 2)/2 THEN (NO
 PALINDROME)
```

or

```
IF (N AND 64)=64 AND (N AND 2)=2 THEN (P
OSSIBLE PALINDROME)
```

The most impressive program of this type is the following from Patti Beadles (Hazelwood, MO):

```
1 REM
2 REM PROBLEM #19-1:
3 REM     BINARY PALINDROME
4 REM BY PATTI BEADLES
5 REM
10 INPUT"NUMBER TO BE CHECKED";N
```


www.commodore.ca

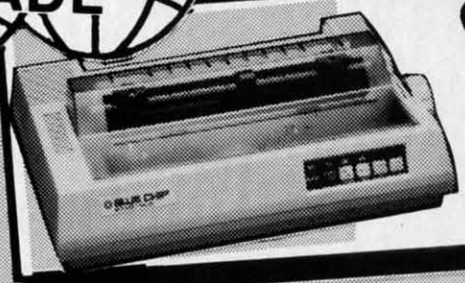
SPECTACULAR HIGH TECH SAVINGS AT WORLD TRADE

COMPUTER WHOLESALE OUTLET

Commodore C-64 COMPUTER $137

• Upper & lower case • Sprite graphic • Sid, sound chip • Built in basic.

COMMODORE 128 Now Available



BLUE CHIP M120/NLQ PRINTER $189

Near letter quality button. Tractor/Friction 120cps/25cps. NLQ.

BLUE CHIP DISK DRIVE $149

100% Commodore compatible. Reads & writes programs faster than Commodore disk dribe. Enhanced technology enables the BCD/5.25 to be more compact, use less power & gives off less heat.

TEKNIKA COMPOSITE MONITOR $159

SOFTWARE FOR COMMODORE 64

CBS SOFTWARE

Grand prize $10,000 cash for tuition. 1st prize $1,000 educational software 2nd prize *100 each for school supplies Plus...with proof of purchase you will get *10 rebate on CBS software's Easy Key programs.

Dr. Seuss Fix Up The Mix-Up Puzzle	19.95
Many ways to say I Love You	18.95
Letter go-round	CALL
Astro-grover	18.95
Big Bird's funhouse	18.95
Ernie's magic shapes	18.95
Weather Tamers	14.95
Forcast	32.95
Dinosaur dig	24.95
Coast to coast	24.95
Mastering the SAT	54.95
Success with Math series	14.95 ea.
Success with Algebra series	24.95 ea.
Muppet Learning Keys	CALL

WORD PROCESSING

Fleet system 2 w/70,000	
Word spellpak	54.95
Font master	14.95
Creative writer	34.95

MODEMS

Video 7	59.95
Miley Mo (1 hour free compuserve time)	64.95
West Ridge	69.95

TIMEWORKS SOFTWARE

Data Manager II	CALL
Evelyn Wood Dynamic Reader	CALL
Word Writer with Spellcheck	CALL
Swiftcalc w/Sideways	CALL
Cave of the Word Wizards	CALL

EPYX SOFTWARE

Hot Wheels	CALL
Summer Games II	CALL
Fast Load	CALL
Barbie	CALL
Breakdance	CALL
Baseball	CALL
Pitstop	CALL
Pitstop II	CALL
9 to 5 Typing	CALL

ACCESS SOFTWARE

Neutral Zone	CALL
Mach 5	CALL
Beach-Head II	CALL

INTERFACES

XETEC SUPER GRAPHIX 69.95

8K buffer
10 printing modes

Super Graphix Jr.	49.95
Tymac Connection	79.95
G-Wiz	CALL

JOYSTICKS

Wico Boss	10.50
Wico Bat	13.95
Wico Threeway	18.95
Gim 1000-left/right firing button	9.95

DISK DRIVES

Commodore 1541	169.00
Enhancer 2000	159.00

DISKETTES

FREE EXTRA DISK INSIDE! Featuring Success with Math, and Success with Algebra by CBS.

Elephant (box of 10) 11.95

Nashua DD/DS (box of 10)	13.95
Nashua diskettes 3½" (box of 10)	27.99
Nashua (box of 10)	9.95
Scotch (3M) (box of 10)	12.95
Scotch Head Cleaning Kit	9.95

Turns your 5½" single Disk into A 2 Sided Disk

NEW! NIBBLE NOTCH 7.95

DISK FILES

10 Disk Holder	6.00
60 Disk Holder (stackable)	9.95

MONITORS

Commodore 1702	169.95
Sharp 13" Color Compsite	144.95
Sharp green 12"	59.95
Sharp amber 12"	69.95
Teknika RGB/Composite 13"	269

PRINTERS

Blue Chip M150/10 Tractor/Friction 130cps	189.00
Blue Chip M150/15	239.00
Star SG10 Tractor/Friction 130cps	205.95
SG10-C w/built in Commodore Interface	CALL
Okimate 10 color printer	CALL
Legende 808	159.95
Legende 880	219.95
Legende 1080	239.95
Legende 1380	279.95
Legende 1385	349.95
Cardco LQ3 letter quality	189.95

PAPER & RIBBON

Delivery on paper included in price

paper size 9½ x 11"

1000 x 3 Boxes Microperf	60.00
2700 microperf	55.00
Ribbon bluechip M120/M150	9.95
Ribbon SG10/SG15/ Okidata 82A	3.95

SURGE PROTECTORS

19.99 & Up

6 Outlet with lighted switch

ATARI

130XE 128K	CALL
800 XL 64K	CALL
1050 Disk Drive	CALL
520ST 520K	CALL
128ST 128K	CALL

SOFTWARE FOR ATARI

ATARI SOFTWARE

Ms. Pac Man	CALL
Pac Man	CALL
Foot Ball	CALL
Millipede	CALL
Robotron 2084	CALL
Jungle Hunt	CALL
Pengo	CALL
Qix	CALL
Donkey Kong	CALL
Joust	CALL
Caverns of Mars	CALL
Dig Dug	CALL
Micro Soft Basic	CALL
Atari Writer	CALL
Assembler Editor	CALL
Visi Calc	CALL

OSS SOFTWARE

Basic XL	54.95
Basic XE	59.99
Mac 65	59.95
Action	59.95
Dos XL W/Bug 65	24.95
Bug 65	29.95
Writers Tool	59.95
Action Tool Kit	24.95
Basic XL Tool Kit	24.95
Mac 65 Tool Kit	24.95

PROFESSIONAL SOFTWARE

Fleet System II Word Processor w/built in spell check	CALL

BRODERBUND SOFTWARE

Print Shop	32.95
Print Shop Lib I	19.95
Print Shop Lib II	19.95

TK SOFTWARE

Monday Morning Manager	24.95
Bowling Version 2	24.95

NEW! BIG FIVE SOFTWARE PRESENTS "BOUNTY BOB STRIKES BACK! ..... CALL
FOR USE WITH BOTH ATARI AND COMMODORE

TO ORDER CALL OUTSIDE N.Y. 1-800-722-4002 ORDER LINE ONLY

NEW! MINDSCAPE INC. PRESENTS "THE HALLEY PROJECT: A MISSION IN OUR SOLAR SYSTEM" .. 28.95
FOR USE WITH BOTH ATARI AND COMMODORE

IN NEW YORK 1-718-895-2388

WHEN IN THE NEW YORK AREA VISIT OUR 6 BIG L STORES. • West Babylon • New Hyde Park • Syosset • West Hempstead • Patchogue • Bayside

Check, Money Order, MC or VISA accepted/Call for shipping and handling information / NYS residents add applicable sales tax / Prices and availability are subject to change without notice / Air factory fresh merchandise with manufacturer's warranty. Dealers welcome. Call for recent price reductions and new rebate information.

FOR INFORMATION & CUSTOMER SERVICE CALL 1-718-895-2388.

VISA MasterCard C.O.D.

Reader Service No. 137

www.commodore.ca

```
20 IF N<0 OR N>255 THEN END
30 IF SGN(N AND 128) <> SGN(N AND 1) THE
N 80
40 IF SGN(NAND64)<>SGN(NAND2) THEN 80
50 IF SGN(NAND32)<>SGN(NAND4) THEN 80
60 IF SGN(NAND16)<>SGN(NAND8) THEN 80
70 PRINT"PALINDROME":GOTO 10
80 PRINT"NOT A PALINDROME":GOTO 10
```

Notice Patti's use of the oft-forgotten SGN function. The result of each AND operation will be either zero or positive. Both AND operations on each line must agree if the number is a palindrome. The SGN function returns the value one if the result is positive and zero if it is zero. No more information is needed to determine the palindromicity (to coin a word) of the number.

Jim Speers (Niles, MI) took the problem a step further to find all 16-bit binary palindromes. Problems occur when dealing with negative numbers. Readers up for an additional challenge might give it a try.

The solution to *Problem #19-2: Memory Locator* from John R. Prager (Bay City, MI) is listed below.

```
1 REM
2 REM PROBLEM #19-2:
3 REM    MEMORY LOCATOR
4 REM BY JOHN R. PRAGER
5 REM
50 A$(8)="HELLO" + " THERE!" :REM DEMO
60 REM
100 DEF FNM(X)=PEEK(X)+PEEK(X+1)*256
110 PRINT"PROGRAM TEXT:" FNM(43) "TO" FN
M(45)-1
120 PRINT"VARIABLES:" FNM(45) "TO" FNM(4
7)-1
130 PRINT"ARRAYS:" FNM(47) "TO" FNM(49)
140 PRINT"STRING STORAGE:" FNM(51) "TO"
FNM(55)
```

Line 50 of John's program demonstrates that the string concatenation causes the computer to use a temporary storage area for the string values. If the "+" and the second string literal were not on line 50, a separate string storage area would not be used.

James Borden (Carlisle, PA) wrote his solution to this problem as a subroutine. You might do the same to see how the storage areas change as your program is executed.

I suggest that you add some FOR-NEXT loops to PEEK into the storage areas to see how the various variables are stored. You might also refer back to the April, May, and December 1984 editions of the *Rupert Report* for some further clues about the inner workings of BASIC. Why is so much variable space used in John's program when there aren't any non-array variables?

Of the numerous solutions to *Problem #19-3: Orthogonal Time,* William Lott's (Coventry, CT) was the easiest to understand.

```
1 REM
2 REM PROBLEM #19-3:
3 REM    ORTHOGONAL TIME
4 REM BY WILLIAM LOTT
5 REM
10 PRINT"+" : X=5400 : GOSUB 1000
20 X=X+10800 : GOSUB 1000 : GOTO 20
1000 S=X/5.5:Y=INT(S):Z=S-Y:IF Z>.5 THEN
 Y=Y+1
1010 IF Y>43200 THEN END
1020 H=INT(Y/3600):M=INT((Y-H*3600)/60):
SEC=Y-H*3600-M*60
1030 PRINT"HOUR="H,"MIN="M,"SEC="SEC:RET
URN
```

In order to find the times at which the hands of the clock form right angles, William uses the fact that the minute hand gains 5.5 (angular) minutes on the hour hand every second. This follows from the fact that the hour hand moves half an (angular) minute every second, and the minute hand moves six (angular) minutes every second.

The program starts at 12 o'clock (0 degrees between the hands) and determines the time at which the minute hand has gained 5400 (angular) minutes (90 degrees) on the hour hand. Every other occurrence of a 90 degree angle is 180 degrees (10800 angular minutes) after the previous occurrence. The subroutine at line 1000 converts angular minutes into hours, minutes, and seconds.

Those of you familiar with the concepts of relativity will recognize that this program uses the hour hand of the clock as a reference frame. The calculations are much simpler if the face of the clock is used as the frame of reference.

A program using a similar approach but written in COMAL by Ray Carter (Las Cruces, MN) is listed here for comparison.

```
1 REM
2 REM COMMODARES #23
3 REM PROBLEM #19-3 : ORTHOGONAL TIME
4 REM COMAL SOLUTION BY RAY CARTER
5 REM
6 REM    (LINE NUMBERS AND LEADING
            COLONS ARE NOT USED)
7 REM ---THIS IS NOT A BASIC PROGRAM---
8 REM
10 :MINUTE'RATE:=.1
11 :HOUR'RATE:=1/120
12 :DELTA:=MINUTE'RATE - HOUR'RATE
13 :ANGLE:=90
14 :REPEAT
15 :    TIMEOUT:=ANGLE/DELTA
16 :    IF  (TIMEOUT<43200)
          THEN PRINTOUT(TIMEOUT)
17 :    ANGLE:=ANGLE+180
18 :UNTIL  TIMEOUT>43200
19 :STOP "THAT'S ALL FOLKS"
20 :PROC  PRINTOUT(TIMEOUT)  CLOSED
21 :    TTIME:=TIMEOUT
22 :    HOURS:=TTIME  DIV  3600
23 :    TTIME:=TTIME  MOD  3600
24 :    MINUTES:=TTIME  DIV  60
25 :    TTIME:=TTIME  MOD  60
```


www.commodore.ca

Aquarian Software Introduces . . .

I.E.A.
Instant Editor Assembler
for the Commodore 64

- ★ Written 100% in machine language
- ★ Assembles 17K source code in 5 seconds!
- ★ Co-resident editor assembler monitor
- ★ Compatible with HES MON
- ★ Slow-motion code analyzer
- ★ Technical assistance number

"I would have to recommend the IEA Instant Editor Assembler, the BEST VALUE-FOR-PRICE development package on the market!"
Jim Gracely
Technical Editor
Commodore Microcomputer Magazine

"IEA is a nice, simple assembler, good for beginners at an UNBELIEVABLE PRICE!"
Jim Butterfield

"...Should be called the POOR MAN'S MERLIN — a great assembler, and you can't beat the price."
William B. Sanders
Author of Assembly Language For Kids

Now Only $17.95

FAST BOOT

Now Available!
Mike J. Henry's
"FAST BOOT!"

The ONLY Fast Booter that works with **Both** 1541 and MSD Drives!

Relocatable in memory.

Disk Based & Non-Protected

Only $14.95

TURBO CALC/64

A Great Beginner's Spreadsheet
At an UNBELIEVABLE Price!!

- 100% Memory Resident
- 100% Machine Code and P-Code
- 100% Menu Driven
- Over 13 K RAM for Data
- Large Matrix, 100 Rows by 26 Columns
- On-Line Help Screens
- On-Screen Menu at All Times
- Full Printer Support
- Built-in Functions (SUM, AVG, MIN, MAX)

At last, a powerful, yet affordable spreadsheet!

ONLY $17.95

Nova Decoder 64
Version 3.0

A 6502/6510 Disassembler
Designed for the Commodore User

A MUST for anyone just learning Machine Language.
A TREAT for those examining the operation of the C-64.
A BLESSING to machine language programmers trying to de-bug their programs.

- ★ Choose Screen or Printer Options
- ★ Addresses Printed in Either Hex or Decimal
- ★ Complete Calculations of Branch Commands
- ★ Brief Explanations of Each Command Encountered
- ★ Generates Tables of Zero Page Addresses Used, Branch Addresses, All Jump Routines, and More . . .

Only $19.95

The Cataloger

The Ultimate Disk Cataloging System for the 64!

Will find any program instantly. Tells exactly which disk number the program is on, the disk name, disk ID, front or back of disk, category (games, utility, music, etc.), if the disk is protected (if so, what errors), the date the program was entered, the block count, and any comments concerning that program. Sort and print a list by any of 12 different fields.

Features of The Cataloger V3.5A Include:

- ★ Easy loading of information. Loads directly from the disk itself.
- ★ Ability to change name of entry.
- ★ Ability to change any information instantly — uses relative files exclusively.
- ★ Search, Sort and Print by any of 12 fields.
- ★ Capacity for 1100 different programs (or disks) per data disk.
- ★ FAST — all machine language.
- ★ Menu driven — very easy to use.
- ★ Works with ALL printers.
- ★ Works with one or two drives.
- ★ Allows duplicate ID's.
- ★ Allows selection of items to be entered.
- ★ Unlimited category names.

Only $24.95

Modem Master

The Fastest Commodore BBS Available!

- Works with 1541 or MSD Dual Drive
- 300/1200 Baud Operation
- New Punter File Transfer Protocol
- Fully Menu Driven
- Over 25 Commands
- Public Message Base
- Private E-Mail Base
- E-Mail Check at Sign-On
- Sub-Directories for File Transfer
- 250 User Capacity
- Accurate Clock/Calendar
- Printer Output
- Information Files
- "Old" E-Mail Deleted After One Week
- User Survey/Poll
- Set Up in Only 10 Minutes!

Only $29.95

Omiterm Terminal

- Fully supports the new 1660 Modem 300!
- Punter protocol — upload & download — 300/1200 baud
- Ten programmable function keys
- 15 number phone directory
- 20k receive buffer
- Tone or pulse dialing
- Auto dial re-dial
- Half/full duplex

Only $19.95

BOOKS!

Easy Script Simplified	$5.00
Printer Codes and Interfaces	$5.00
1541 Revealed	$5.00
Programming Development Language	$5.00
I'm Sorry, But I Don't Speak Hexidecimal	$5.00
1541 Survival Guide	$5.00
Computer Talk	$8.00

Call For Details

Canadian Best-Sellers NOW Available in the U.S.

The Gold Disk Series

Quality Software for Your 64 at an Affordable Price!

Only $14.95 Per Disk*
* Plus Shipping and Handling

Each Disk Contains:

The Feature Program
A Home Business Program
Two High-Quality Games
Tutorials on Programming
And MUCH MORE !!

Volume 9 Now Available — Featuring an Icon-Driven Card File
Volumes 1 through 8 Are Also Available
Dealers Welcomed! Call (503) 654-2641 For Details.

Aquarian Software
P.O. Box 22184
Portland, OR 97222

VISA
MasterCard

To Order, Call: (503) 654-2641
VISA and MasterCard Accepted
Add $3.00 Shipping and Handling Per Order
Dealer Inquiries Welcomed!

www.commodore.ca

```
26 :    SECONDS:=INT('TIME+.5)
27 :    PRINT USING "##:##:##":HOURS,
        MINUTES,SECONDS
28 :ENDPROC  PRINTOUT
```

The program structures and functions available in this language can make most of us "bare-bones BASIC" programmers somewhat envious. The program flow should be fairly obvious even to programmers that have not used COMAL. The rates defined in this program are in angular minutes per second. The minutes on the clock face are six angular minutes apart, and the minute hand travels six angular minutes in six seconds or 0.1 angular minute per second. (1 angular degree = 60 angular minutes.)

The final problem this month left a few readers somewhat frustrated. *Problem #19-4: Screen Scramble* did not prescribe how to handle the fact that a 40 by 25 screen cannot simply be rotated into a 25 by 40 area. Such details were left to the programmer's creativity. Both solutions listed below chopped off the screen at the 25th column and gave it a 90 degree clockwise rotation. There is one significant difference between the ways they did it, though.

The solution from Allan Flippin (San Jose, CA) actually rotates each character 90 degrees. After you run his program, you may stand your monitor on its side and see the original screen appearance!

RENT BUY

commodore 64

IBM Apple — Atari Software

EXCITEMENT

Hint Book 75 Adventures $19.95

*DATABASES & UTILITIES

*FOREIGN LANGUAGES

*ADVENTURE

*AT

SPECIAL SALE $4.95

The C-64 Master Memory Map is a clearly written, friendly guide to the inner workings of your computer. This book is regularly priced at $15.95 by Prentice-Hall, and at $4.95 is such a bargain that we guarantee your money back if you are not happy with it.

*EDUCATIONAL SOFTWARE

*CHILDREN'S PROGRAMS

3-HOUR SHIPPING CHARGE

Call toll-free outside Texas: 1—800—433-2938

— Inside Texas call: 817—292-7396

WEDGWOOD RENTAL

5316 Woodway Drive

Fort Worth, Texas 76133

Reader Service No. 111


```
1 REM
2 REM COMMODARES #23
3 REM PROBLEM # 19-4 : SCREEN SCRAMBLE
4 REM SOLUTION BY ALLAN FLIPPIN
5 REM
400 X=4:PRINT"+"
402 FOR Y=0 TO 11:FOR X=0 TO 24:POKE 102
4+40*Y+X,(25*Y+X)AND255
403 POKE55296+40*Y+X,11:NEXT:PRINT:NEXT
404 FORY=13TO20:FORX=0TO24:POKE1024+40*Y
+X,160:POKE 55296+40*Y+X,XAND15
406 NEXT:PRINT:NEXT:PRINT:PRINT"PRESS AN
Y KEY WHEN READY":I=49152
407 GET A$:IF A$="" GOTO 407
408 READ A:IF A<>-1 THEN POKE I,A:I=I+1:
GOTO 408
410 POKE 780,238:POKE 781,4:POKE 49187,9
2:SYS(49183)
411 POKE 53272,120:POKE 53265,59:POKE 56
576,150
412 FOR Y=0 TO 24:FOR X=0 TO 24
414 POKE 23584+40*X-Y,16*(PEEK(55296+40*
Y+X)AND15)+6:NEXT:NEXT
416 POKE 780,0:POKE 781,32:POKE 49187,96
:SYS(49183)
418 POKE 56334,0:POKE 1,51
420 FOR Y=0 TO 24:FOR X=0 TO 24
422 C=PEEK(1024+40*Y+X)
424 POKE 49160,208+INT(C/32):POKE 49159,
(C*8)AND255
426 S=24832+320*X-8*Y:POKE 252,INT(S/256
):POKE 251,S-256*PEEK(252)
428 SYS(49152):NEXT:NEXT
430 POKE 1,55:POKE 56334,1
434 GET A$:IF A$="" GOTO 434
436 SYS(65126)
438 DATA 169,1,133,253,162,0,189,0,0,160
,7,74,144,8,72,177,251,5,253,145,251
440 DATA 104,136,16,242,6,253,232,144,23
2,96,160,0,153,0,0,200,208,250,238
442 DATA 35,192,202,208,244,96,-1
```

A discussion of Allan's program is too lengthy for this column. If you want a copy of his source listing and remarks, send a self-addressed stamped envelope to *Commodares* with your request and I will send it to you. Basically Allan takes each character's bit map from ROM, rotates it, and puts it onto the screen at the new location. Anyone interested in having some serious programming done might do well to check with Allan.

The program from Bob Martin (West Lafayette, IN) involves some fancy work as well. The screen characters are printed from top to bottom starting at the upper right corner of the screen.

```
1 REM
2 REM COMMODARES #23
3 REM PROBLEM # 19-4 : SCREEN SCRAMBLE
```


Continued on page 146

www.commodore.ca

WHITE HOUSE COMPUTER

In PA: 1-717-322-7700
1-800-351-3442

P.O. BOX 4025, WILLIAMSPORT, PA 17701

HOURS: Monday thru Friday 9:00 am till 6:00 pm

VISA and MC Accepted 4%

★ PRINTERS ★

C-64
- MPS 801 ..... 179.95
- 802 ..... 208.95
- 803 ..... 175.95
- MCS 801 ..... 314.95
- DPS 1101 ..... 314.95

CITIZEN
- MSP-10 325
- MSP-15 479
- MSP-20 469
- MSP-25 629

EPSON
- RX80 219
- LX80 249
- FX80 369
- JX80 529
- RX100 369
- FX100 550

C. ITOH
- 8510 A 285
- 8510 BC1 329
- 8510 BC2 389
- 8510 BP1 339
- 8510 SP 389
- 8510 SR 429
- 8510 SCP 459
- 8510 SCR 477
- 1550 BCD 485
- 1550 P 445
- A10-20P 465
- F10-40P/S 875
- F10-55 P/S 1059
- F10 Tractor 169
- F10 Sheet Feeder 349

MANNSEMAN TALLY
- Spirit 80 249
- MT 160L 495
- MT 180L 595

STAR MICRONICS
- SG10 214
- SG15 379
- SD10 345
- SD15 450
- SR10 485
- SR15 585
- Powertype 309

CARDCO
- LQ 1 365
- ★LQ 3 209

PANASONIC
- 1090 187
- 1091 237
- 1092 389
- 1093 585
- 3151 455

LEGEND
- 880 209
- 1080 229
- 1380 269
- 1385 305

OKIDATA
- 182 ..... 219.95
- 84 ..... 640.95
- 92 ..... 349.95
- 93 ..... 560.95
- Imagewriter 425.95

DISK DRIVES
C-64
- 1541 ..... 185.95
- 1571 ..... 249.95
- 1572 ..... 375.95
- MSD Single 229.00
- MSD Dual 459.00
- Indus GT 239.00
- Cardco SCD-1 ..... 199.00

INTERFACES
- Tymac Connection/2K 62.95
- Cardco +G 59.95
- Cardco B 37.95
- OR-D GPC w/2K 55.95
- Grappler CD 89.95

PRINTER PAPER
- Lazor 1000 shts 15.95
- Lazor 2500 shts 29.95

DISKETTES ★
SKC
- SS/SD 10.50
- SS/DD 13.50
- DS/DD 15.50

MAWELL
- MD-1 ..... 15.95
- MD-2 ..... 21.95

ELEPHANT
- SS/SD 13.50
- SS/DD 15.50
- DS/DD 19.50

INNOVATIVE CONCEPTS
Flip-N-File
- 10 ..... 3.50
- 15 ..... 8.50
- 25 ..... 18.50
- w/lock 50 ..... 17.50
- 50 ..... 23.50
- ROM Holder 17.50

KOALA
- Pad Disk 39.95
- Pad ROM 45.95

MONITORS
AMDEK
- 300G- 119
- 300A 129
- 310A 145
- Color 300 235
- Color 500 345
- Color 600 399
- Color 700 469
- Color 710 539

TEKNIKA
- MJ-10 199
- MJ-22 285

XTRON
- Comcolor I 189

ZENITH
- 122 ..... 85.95
- 123 ..... 75.95

BMC
- Amber 82.95
- Green 79.95

COMMODORE
- C-1702 ..... 199.95
- C-1802 ..... 204.95
- C-1902 RGB 259.95
- C-1901 Monchrome 129.95

Dealers Inquiries ★ Invited

SOFTWARE ★

BATTERIES INCLUDED
- Paper Clip w/Spell Pk 69.95
- Paper Clip 59.00
- Home Pack 32.95
- Bus Card II 124.95
- B180 Board 99.95

BRODERBUND
- Print Shop 29.95
- Graphics Library 18.95
- B.S. Writer 32.95

EPYX
- FAST LOAD 23.95

VIDEO CASSETTE ★★ RECORDER $275.95

Symphonic VHS/Remote

- 105 Channel Cable Ready
- 14 Day / 1 Program Timer
- 12 Pre-set Channels
- Auto Rewind

Plus much, much more!

CHECK OUR PRICES!

COMPUTERS

C-64 ..... 149.95

C-128 ... 275.95

PRINTER RIBBONS

DUST COVERS

- Mitey Mo ... 59.95
- Westridge ... 69.95
- Compuserve Starter Kit ... 21.95

MODEMS
- C-1600 ..... 44.95
- C-1650 ..... 54.95
- C-1660 ..... 54.95
- C-1670 ..... 185.95
  300/1200 Band

Cardco
- MOD-1 ..... 59.95

C-64
- **MPP-1064 ... 65.95**
- **Tele Learning 49.95**

WE CARRY A FULL LINE OF:
Batteries Inc., Broderbund, Contineal, Hess, Infocom, Innovative Concepts, Koala, LJK, Microbits, Micropose, Microtek, O.S.S., Professional S.W. Scarborough, S.S.I., Synapse, Time Works, Wico.

POLICY: No deposit on COD orders. FREE freight on all prepaid cash orders over $300 in the continental USA. APO and FPO add $5 per hundred. For priority Mail add $8 per hundred. PA Residents add 6% sales tax. Defective products must have Prior RA number. Schools net 15.

Reader Service No. 132

www.commodore.ca

FREE!

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

ARCADE

Archon	19.95
Archon II Adept	26.95
Beachead II	25.95
Blue Max 2001	20.95
Bruce Lee	24.95
Castles of Dr. Creep	20.95
Championship Loderunner	24.95
Computer Quarterback	28.95
F15 Strike Eagle	23.95
Flight Simulator II	35.95
Impossible Mission	23.95
Karateka	20.95
Kennedy Approach	23.95
Mail Order Monsters	26.95
Micro League Baseball	28.95
Statistics Disk (1984)	13.95
Night Mission Pinball	21.95
Pitstop II	27.95
Questron	27.95
Racing Destruction Set	26.95
Raid Over Moscow	27.95
Rescue on Fractalus	20.95
Skyfox	24.95
Solo Flight	23.95
Spy vs. Spy	20.95
Stealth	20.95
Strip Poker	20.95
Summer Games II	27.95
Super Bowl Sunday	20.95
Super Huey	14.95

ADVENTURE AND MIND GAMES

Amazon	23.95
Adventure Construction Set	31.95
Below the Root	18.95
Bridge 4.0	17.95
Cutthroats	28.95
Dragonworld	23.95
Fahrenheit 451	23.95
Felony	23.95
Hitchhikers Guide	24.95
Infidel	24.95
Master of the Lamps	20.95
Micro Astrologer	15.95
Mind Prober	21.95
Mind Wheel	27.95
Monty Plays Scrabble	27.95
Murder By The Dozen	23.95
Party Quiz	49.95
Party Quiz Refills	17.95
Phantasie	27.95
Rendevous With Rama	23.95
Robots of Dawn	27.95
Scott Adams Adventures	22.95
Seven Cities of Gold	26.95
Starcross	26.95
Suspended	26.95
Swiss Family Robinson	18.95
Ultima IV	43.95
Wishbringer	24.95
Witness	24.95
Zork II & III	19.95

EDUCATIONAL PROGRAMS

Agent USA	21.95
Bank Street Storybook	27.95
Build a Book	28.95
Cave of the Word Wizard	24.95
CBS Success w/Math Series ..... ea.	18.95
Evelyn Wood's Dynamic Reader	35.95
Gertrude's Secrets	20.95
Halley Project	33.95
Harcourt Brace Computer SAT	54.95
Kidwriter	19.95
Muppet Learning Keys	49.95
Reader Rabbit	26.95
Sky Travel	27.95
Spell It	33.95
Story Machine	22.95
Typing Tutor III	28.95
Word Attack	33.95

WORD PROC.-DATA BASES-SPREADSHEETS

Batteries Included Home Organizers	
Audio/Video	13.95
Checkbook	13.95
Electronic Address Book	13.95
Home Inventory	13.95
Mail List	13.95
Photos/Slides/Movies	13.95
Recipes	13.95
Batteries Included Home Pak	34.95
B-Graph	19.95
Calc Result Advanced	52.95
Complete Personal Accountant	54.95
Consultant (Delphi's Oracle)	68.95
Cut and Paste	36.95
Financial Cookbook	31.95
Fleet Systems II	56.95
Magic Desk I	35.95
Mirage Data Base Manager	38.95
Mirage Word Processor	38.95
Net Worth	55.95
Paperclip w/Spellpack	82.95
PFS File	55.95
Practi Calc II	35.95
Practi File	34.95
Silent Butler	23.95
Superbase 64	52.95

GRAPHICS

Animation Station	63.95
Doodle	28.95
Koala Pad	65.95
Koala Printer	17.95
Micro Illustrator	23.95
Newsroom	CALL
Perspectives	38.95
Print Shop	31.95
Print Shop Graphics Library	17.95
Screen Dumper 64	15.95

MISCELLANEOUS

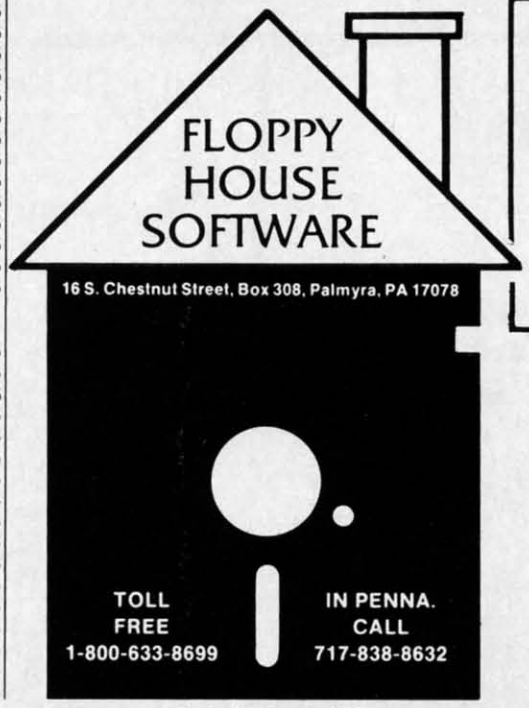
Bank Street Music Writer	34.95
Bulk Discs - Quantities of 10	9.95
Clone Machine	37.95
Compuserve Starter Kit	21.95
Copy II 64	27.95
Facsimile	17.95
Fast Load	27.95
Isepic	52.95
Mach 5	24.95
Maxell Disks - Box of 10	15.95
Ortho Complete Gardening	37.95
Pro-Tech Vinyl Covers	5.35
Simon's Basic	38.95
Tac II Joystick	9.95
Vidtex	21.95
VIP Terminal	29.95

HARDWARE AND PERIPHERALS

B.I. 80 Column Display	134.95
Cardco +G Interface	43.95
Cardco Numeric Keypad	36.95
Commodore Hardware	CALL
Indus Disk Drive	245.95
Okimate 10 w/Plug and Print	175.00
Okimate 10 Blk & Wht Ribbon	4.75
Okimate 10 Color Ribbon	5.25
Panasonic KXP1091 Printer	279.00
PPI Interface	39.95
Prowriter 7500 Printer	229.00
Prowriter 8510 AP Printer	319.00
Prowriter 8510 Ribbons	7.95
Tac II Joystick	9.95
Total Telecommunications Modem	42.95

LEROY'S CHEATSHEETS $2.95

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



SPECIALS
Compuserve Starter Kit/Vidtex
39.95
S-Games
(Formerly Hes Games)
11.95

MAILING CHARGE ON LEROY'S CHEATSHEETS
$1.00

ORDERING & TERMS

* VISA & M.C. accepted with no charge in the continental U.S.A.
* $2.50 shipping charge on orders under $100.00
* Pa. Residents add 6% Sales Tax
* Prices subject to change w/o notice
* 48 hour shipping for cashier check - Money order - charge

Reader Service No. 247

www.commodore.ca

ERRATA

FASTNEW

An earlier version of *Fastnew*, which *does* bang the R/W head against the stop, accidentally saw print. In order to upgrade the published version of the program to the no-bang version, follow these steps:

Load in *Fastnew* as published. Replace lines 12 and 50 with the following:

```
12 REM          V0316/85
50 FOR I=0 TO 509:READ A:POKE I+9472,A
```

In line 1007, change the string V010585 to V031685. Replace the following lines as listed below.

```
5010 DATA 28,165,34,133,74,198,74,32
5020 DATA 212,6,198,74,208,249,162,0,32,
219
5040 DATA 36,240,12,32,72,5,48,7
5050 DATA 230,34,32,202,6,240,238,173
5270 DATA 141,3,28,32,245,6,169,0
5440 DATA 133,77,32,178,6,162,10,80
5470 DATA 105,10,133,48,76,134,6,169
5480 DATA 244,133,75,96,32,178,6,160
5570 DATA 32,205,6,174,0,28,232,76
5580 DATA 219,6,32,215,6,174,0,28
5630 DATA 160,32,208,240
```

Also lines 60000, 60001, 62000-62004 can all be deleted. They were useful during development of the program but are no longer needed.

AHOY! DOCK

Lines 1070, 1110 and 9520 each contained a GOSUB that had misprinted and was spelled GOSUSB. Ignore the extra S. Line 9550 was also misprinted. That line should read:

```
9550 F=.:FORJ=1TOC-1
```

START & END FILE ADDRESS

Due to an error in our listing generator program, line 10 is incorrect. It should read:

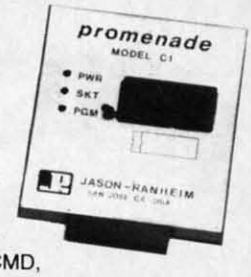
```
10 Q=24:P=Q*256:S=INT(P/256)
```

SOLITAIRE 64

Lines 6 and 7 contain a [008] within quotes. It should be replaced in each line by CNTRL H inside the quote marks.

Program Your Own EPROMS

▶ VIC 20
▶ C 64 **$99.50**



PLUGS INTO USER PORT.
NOTHING ELSE NEEDED.
EASY TO USE. VERSATILE.

promenade™

- Read or Program. One byte or 32K bytes!

OR Use like a disk drive. LOAD, SAVE, GET, INPUT, PRINT, CMD, OPEN, CLOSE—**EPROM FILES!**

Our software lets you use familiar BASIC commands to create, modify, scratch files on readily available EPROM chips. Adds a new dimension to your computing capability. Works with most ML Monitors too.

- Make Auto-Start Cartridges of your programs.
- The *promenade*™ C1 gives you 4 programming voltages, 2 EPROM supply voltages, 3 intelligent programming algorithms, 15 bit chip addressing, 3 LED's and NO switches. Your computer controls everything from software!
- Textool socket. Anti-static aluminum housing.
- EPROMS, cartridge PC boards, etc. at extra charge.
- Some EPROM types you can use with the *promenade*™

2758	2532	462732P	27128	5133	X2816A*
2516	2732	2564	27256	5143	52813*
2716	27C32	2764	68764	2815*	48016P*
27C16	2732A	27C64	68766	2816*	

▶ *Commodore Business Machines — *Denotes electrically eraseable types

Call Toll Free: 800-421-7731 **or 408-287-0259**
In California: 800-421-7748

JASON-RANHEIM
580 Parrott St., San Jose, CA 95112

VISA MasterCard

Reader Service No. 112

PROFESSIONAL HANDICAPPING SYSTEMS

• PRESENTED BY PROFESSOR JONES •

GLD. Thoroughbred "Gold" Edition™
A "Full" featured thoroughbred analysis designed for the professional and the serious novice. **$159.95** complete

EGLD. Enhanced "Gold" Edition™
"Gold" Edition with complete Master Bettor™ system integrated onto the same disk. This powerful program will transfer all horses and scores to the bet analysis with a "single keystroke." (Master Bettor™ included) **$199.95** complete

GLTD. Limited "Gold"™
Enables **Professional Handicappers** to assign specific values to the racing variables "they" feel are important. Create program weight based on a particular track and fine tune it for **maximum** win percentage. This program is designed for "ease of use". The user needs **no programming experience.**
(contains Integrated Bettor™) **$299.95** complete

GD. Gold Dog Analysis™
The "ONLY" professional greyhound analysis available that evaluates ALL variables. **$149.95** w/Integrated Bettor **$199.95** Limited Version **$299.95**

MHH. Master Harness Handicapper™
Professional software designed to provide a thorough analysis of all trotter and pacer races in North America and Canada. **$159.95** complete **$199.95** w/Integrated Bettor Limited **$299.95**

Professor Pix Football™
Complete STATISTICAL ANALYSIS on Data Base allowing "Designated" previous games to be evaluated. Statistical Series **$99.95** w/Win-Loss Power Ratings **$149.95**

NBA. Basketball™ NBA **$99.95** w/college **$129.95** w/power ratings **$149.95**

LOT. Lottery Analysis™
Statistical comparison program designed to detect subtle patterns in winning numbers and digits. Lottery (3-4 digit) **$79.95** w/Lotto (Max. 99 Digit) **$99.95**

PC-3 Portable Computer (4k) with choice of Thoroughbred, Greyhound or Trotter™ **$249.95** (Includes portable computer and program.)

M-100 Portable (24k) w/choice of Thoroughbred, Greyhound, or Trotter. **$649.95** (Includes portable computer and program.)

BROCHURE AVAILABLE

IBM™
APPLE™
TRS-80™
CPM™
COMMODORE™

Prof. Jones
1940 W. State St.
Boise, ID 83702

VISA MasterCard

48 HR. FREE SHIPPING

CALL
208-342-6939

TERMS: FREE SHIPPING ALL SOFTWARE. Add $6.00 hardware / $6.00 C.O.D. / UPS Blue $6.00 / Out of Country $9.00 / ID Residents 4% / 3 weeks personal checks / Cash price only, add 2% Visa, MC / Prices subject to change.

Reader Service No. 108

www.commodore.ca

Now Commodore
giant leap forward

Introducing Viewtron. The new service that's easier to use, and faster to operate. At a price you may never see again.

Viewtron represents the newest, most advanced generation of on-line services. And it's less expensive to buy, less expensive to use. It's a complete service that includes information, shopping, home banking, messaging, air fares, schedules, and much more.

Viewtron is so much easier to use than other services because its sophisticated navigational structure uses everyday English keywords rather than arbitrary meaningless codes. Our system makes it faster getting around. So it saves you time. *And that saves you money.*

For instance, just type "stocks," "encyclopedia," or "news" when you want the latest stock market data, educational material, or up-to-the-minute news. You can even create your own keywords.

A single button can get you what you need.

With the touch of one button, you access a keyword file that lets you quickly review information which is particularly important to you. Like the current price of your securities, or your favorite team's stats and scores, or news in your career field.

Special services exclusively for Commodore owners.

Viewtron lets you "Ask the Experts" for answers to Commodore questions. Or order bargains on hot-selling Commodore software. And you can bid on software and hardware in our electronic auction, where prices keep dropping instead of rising. Plus you get other services like Commodore software reviews. With Viewtron, you can also download free software.

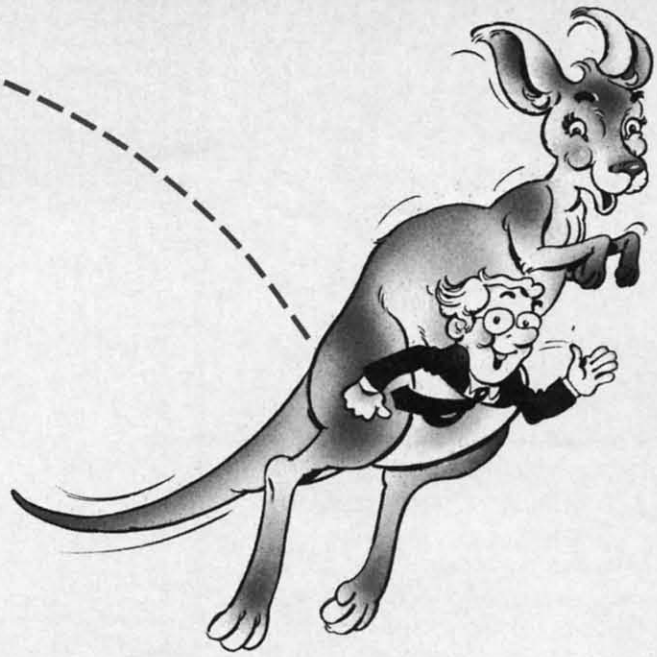
What's more, Viewtron is the only complete on-line service that *runs in color* on Commodore machines.

Pay only for what you use.

Viewtron charges no subscription fee and no monthly minimum. You pay only for what you use. Just 9 cents a minute weekday nights (after 6 pm) and weekends; 22 cents a minute weekdays.* And unlike other services, we don't charge extra if you use a 1200-baud modem.

You can access Viewtron on any of three communications networks, UNINET,® GTE Telenet,® or TYMNET.® (A local call for most people.)

www.commodore.ca



owners can take a in on-line services.

Limited trial offer!
FREE HOUR of Viewtron
when you buy a Starter Kit at low introductory price. Only $9.95.

Just mail the coupon or call. You'll get a free hour of Viewtron.** Plus communications software and an easy-to-understand user's guide. Just $9.95. With money back guarantee.

If you need a modem – rejoice! Get Viewtron and a modem for just $49.95. (A $109 value.)

An unbeatable deal. An unbelievably low price for a modem, and the software is free. Get Viewtron, plus a 300-baud Westridge 6420 by Anchor Automation. The modem plugs into your Commodore and amazingly enough at this low price it has Auto-Dial and Auto-Answer. Satisfaction guaranteed. Order now.

LEAP ON THIS DEAL WHILE YOU CAN.

Call 1 (800) 543-5500. Dept. 9001. Or mail coupon:

Viewdata Corporation of America, Inc., Dept. 9001, P.O. Box 317678, Cincinnati, Ohio 45231.

NAME _____ ADDRESS _____

CITY/STATE/ZIP _____ ( ) PHONE NUMBER _____

Credit cards only; sorry, no checks. Charge to: ☐ MasterCard ☐ VISA ☐ American Express

Card Number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiration Date ☐☐ ☐☐

SIGNATURE: _____

Yes, send me one of the Viewtron offers checked below.

VIEWTRON STARTER KIT
☐ for my Commodore 64 or 128. Only $9.95.

MODEM PACKAGES FOR COMMODORE 64 or 128
☐ 300 Baud Westridge 6420 by Anchor Automation with FREE Viewtron Starter Kit. Only $49.95.
☐ 1200 Baud Volksmodem 12 by Anchor Automation with FREE Viewtron Starter Kit. Only $189.95.

SATISFACTION GUARANTEED or your money back. All modem offers include a cable (where necessary) to connect the modem to your computer. If for any reason you are not satisfied, send back the merchandise within 30 days to receive a full refund. Sorry, no C.O.D.s. Allow four weeks for delivery. Shipping included. Florida, Illinois and New York residents will have the appropriate sales tax added to their order.

Viewtron®

A leap forward in on-line services.

To order, mail the coupon or call 24 hours a day, 7 days a week, 1 (800) 543-5500. Ask for Dept. 9001. Satisfaction Guaranteed.

Viewtron® is a service of Viewdata Corporation of America, Inc., which is a wholly-owned subsidiary of Knight-Ridder Newspapers, Inc., one of America's largest communications companies. Commodore 64™ is a trademark of Commodore Electronics, Ltd.

*Viewtron is not available in all cities. Prices may vary in some cities.

**Your free hour is available weekdays after 6 pm and weekends.

Reader Service No. 129

www.commodore.ca

DISCOUNT SOFTWARE

ABBY'S CARRIES A FULL SELECTION OF SOFTWARE FOR YOUR COMMODORE COMPUTER. CALL FOR CURRENT PRICES.

ABBY'S CHRISTMAS STOCKING STUFFERS

ATARISOFT VIC-20 (ROM)
Pac Man, Donkey Kong, Centipede, and Defender
All Four Titles for only **$12** PLUS Receive **FREE** Gortek & Michrochips

ATARISOFT C-64 & C-128 (ROM) ..... **$8 Each**
Robotron 2084, Defender, Dig Dug

ATARISOFT APPLE (Disk) ..... **$7 Each**
Stargate, Jungle, Defender, Pac Man, Donkey Kong, Robotron, Dig Dug, Centipede

SPINNAKER C-64 & C-128 (ROM) ..... **$16 Each**
Adventure Creator, Aegean Voyage, Alf in the Color Caves, Alphabet Zoo, Bubble Burst, Delta Drawing, Facemaker, Fraction Fever, Jukebox, Kids on Keys, Kindercomp, Up For Grabs

FISHER PRICE C-64 & C-128 (ROM) ..... **$13 Each**
Dance Fantasy, Memory Manor, Linking Logic, Logic Levels, Alpha Build, Hop Along Counting, Up & Add 'Em, Sea Speller, Number Tumbler

We Carry A Complete Line of Software. The Following Is Just a Sample of our Products

ABACUS

Ada	$45
Fortran	$39
C-Basic	$39
Ultrabasic	$27
TAS-64	$69
Basic-64	$36
Assembler-Editor	$36
Power-Plan	$49
Anatomy of the Commodore (Book)	$19.95
Anatomy of the Commodore (Disc)	$14.00
Anatomy of the 1541 (Book)	$19.95
Anatomy of the 1541 (Disc)	$14.00
Super Pascal	$48.00
Super C Compiler	$59.00

CBS

Felony	$27
Murder By Dozen	$27
Success With Math Series	Each $21

MICROPROSE

Air Rescue I	$24
F-15 Strike Eagle	$25
Kennedy Approach	$24
NATO Commander	$24
Decision In The Desert	$25
Crusade In Europe	$25

EPYX

Fast Load (R)	$28
Summer Games II	$28
Ball Blazer	$28
Rescue on Fraculus	$28
Robots of Dawn	$28
Jet Combat Simulator	$28
Winter Games	$28
Breakdance	$28
Impossible Mission	$25
Scrabble	$28

ELECTRONIC ARTS

Music Construction Set	$17
Realm of Impossibility	$17
7 Cities of Gold	$24
Adv. Construction Kit	$28
Archon	$17
Archon II	$24
Financial Cookbook	$28
One On One	$24
Pinball Construction	$17
Murder/Zinderneuf	$17
Moviemaker	CALL
Sky Fox	$27
Reach For Stars	$39
Europe Ablaze	$39
Carriers At War	$39

INFOCOM

Deadline	$35
Enchanter	$27
Hitch Hiker	$27
Infidel	$29
Sorcerer	$29
Witness	$27
Invisaclues	each $7
Wishbringer	$29
ZORK I	$25
ZORK II & III	$29

SSI

Kampfgruppe	$49
Field of Fire	$29
Operation Market Garden	$37
Computer Quarterback	$29
Battle for Normandy	$29
Broadsides	$29
Rails West	$29
Computer Baseball	$29
Baltic 85	$27
Combat Leader	$29
Gemstone Warrior	$27
Wings of War	CALL
Imperium Galactum	$29
Six Gun Shootout	$29
Phantasie	$29

MINDSCAPE

Show Director	$29.00
Perfect Score: Sat.	$54.00
Crossword Magic	$39.00
Keyboard Cadet	$29.00
Bank St. Music Writer	$39.00
Color Me: The Computer Coloring Kit	$24.00
To Use With Color Me:	
Hugga Bunch	$8.00
Rainbow Brite	$8.00
Shirt Tales	$8.00
Supply Box	$8.00
Tinka's Mazes	$19.00
Tinka's Adventure	$19.00
Tonk in the Land of Buddy-Bots	$19.00
Halley Project	$29.00

TOP HITS

Fleet System II	$65
Paperclip/Speller	$85
Homepak	$37
B/Graph	$29
Cave of the Word Wizards	$24
Beachhead II	$29
CSM 1541 Disk Alignment	$39
Print Shop	$31
Indiana Jones	$24
Home Accountant	$59
Net Worth	$71
Adv. Writer	$32
Code Writer	$69
Super Base	$59
Ghostbusters	$25
Star League Baseball	$25
Star League Football	$25
Doodle	$29
Dow Jones	$27
Compuserve	$24
Micro League Baseball	$31
Team Disk (for above)	$16
Master Type (R)	$29
Flight Sim. II (D or C)	$39
Mr. Nibble	$34
Mach 5	$29
VIP Terminal II	$36
Super Bowl Sunday	CALL
Vizastar	$85
Calkit	$39
Amazon	$26
Farenheit 451	$26
Xyphus	$29
I Am The C-64	$21
Spy vs. Spy	$21
Below The Root	$19
Swiss Family Robinson	$19
Mindwheel	$29
Spy Hunter	$34
Tapper	$34
Frogger II	$26
Super Zaxxon	$26

HARDWARE

1670 Modem	CALL
C-128 Computer	CALL
1571 Disk Drive	CALL
1572 Dual Drive	CALL
1902 Monitor	CALL
Okimate 10 W/Interface	CALL
Complete Line of Epson Printers and Star Printers	CALL
AMIGA	**CALL**
Toshiba 5400 VCR	$399
Toshiba 5000 VCR	$299

ACCESSORIES & BOOKS

Print Shop Paper Refill	$16
Disk Notcher	$ 6
Ribbon Cartridge For Okimate 10	$ 6
128 K Upgrade for C-128	CALL
384 K Upgrade for C-128	CALL
Covers for All Computers & Accessories	$8 & Up
G-WIZ	$59
Wabash (Box of 10)	$10

ALL TITLES ON DISK UNLESS MARKED (R) FOR ROM CARTRIDGE

Order Line
1-800-282-0333
ORDERS ONLY! Sales Personnel Do Not Have Information on Previous Orders or Product Specs.

M-F 10 a.m.-7 p.m.; Sat. 10-3 ET

37 S. Broad Street, Fairborn, OH 45324

Customer Service
1-513-879-9699

AD #AH-115

VISA

Prepaid orders over $50 receive free shipping, UPS, continental US. No waiting period when paid by credit card, certified check or money order. Add $2 shipping and handling on orders under $50. Add $5 for COD orders. $15 Minimum for VISA/MC. Hardware requires additional freight charges. Ohio residents add 5.5% sales tax. All items subject to availability and price change.

MasterCard

PLEASE CITE AD NUMBER WHEN ORDERING. Reader Service No. 278

Commodore.ca

CHOPPER FLIGHT FOR THE C-64

BY MIKE BUHIDAR, JR. AND KEVIN WORAM

In *Chopper Flight,* you are a civilian helicopter pilot who has been called upon by his country to fly a rescue mission. A group of ruthless terrorists have captured two adjacent apartment buildings in your city and are holding all the tenants hostage. If their demands are not met in one hour, they threaten to blow up the building, killing themselves and all of the inhabitants. The nearest military helicopter base is too far away to send help in time, so you have been asked to use your own personal chopper to rescue the hostages. However, the terrorists don't appreciate your sense of duty and they start to five anti-aircraft shells at you. Shortly before a shell is fired, you will hear a warning noise. If you hurry, you can guide your chopper onto a ledge where you will have some measure of safety. When the terrorists realize that you could ruin their plans, they send for a heat-seeking missile which automatically homes in on your chopper. However, before the missile is fired, the border of the screen will turn orange and then red. This is a warning that your time is almost up.

As the game begins, you are asked to enter the difficulty of the mission. The skill level ranges from one to six, one being the easiest, six being the hardest. The height of the building and the amount of time before the heat-seeking missile is fired depend on the skill level you select. Also, your maneuverability varies with the difficulty of your mission.

To maneuver your chopper, use a joystick plugged into control port 2. To control your rate of descent, hold down the fire button on your controller. After a short period of holding down the button, your helicopter will begin to hover. At this point you can maneuver your craft between the ledges to rescue the hostages. Releasing the fire button will cause the chopper to begin descending again. Movement in all eight directions is possible. However, do not try to move your chopper off the top or bottom of the screen; once a section of the building has scrolled off the top of the screen, it is gone and you cannot go back to it. Likewise, the only way to descend to the lower parts of the building is to allow your chopper to drop by releasing the fire button.

On every fourth floor of one of the buildings, you will see a hostage standing on a ledge. To save him, slow your chopper's downward fall to a hover and maneuver your chopper between the ledges. As soon as you touch the hostage, he is replaced by a heart symbol to show that he is safely on board. If you allow a hostage to scroll off the top of the screen, he is beyond help and you cannot save him.

If you are able to reach the bottom without crashing into the building, being hit by a shell, or running out of time, you will receive one hundred points for every man you have rescued. Also, extra points are awarded for the difficulty of your mission and for completing it in the least amount of time. □

SEE PROGRAM LISTING ON PAGE 136

SYMBOL MASTER™

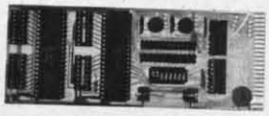
MULTI-PASS SYMBOLIC DISASSEMBLER FOR COMMODORE 64* & 128*

(Includes C-128 native 8502 mode) disassembles any 6502/6510/8502 machine code program into beautiful source

- Learn to program like the experts!
- Adapt existing programs to your needs!
- Automatic LABEL generation.
- Outputs source code files to disk fully compatible with your MAE,* PAL,* CBM,* Develop-64,* LADS* or Merlin* assembler, ready for re-assembly and editing!
- Outputs formatted listing to screen and printer.
- Automatically uses NAMES of Kernal jump table routines and all operating system variables.
- Disassembles programs regardless of load address. Easily handles autorun "Boot" programs.
- Generates list of equates for external addresses.
- Generates complete cross-referenced symbol table.
- Recognizes instructions hidden under BIT instructions.
- 100% machine code for speed. Not copy protected.

ORDER NOW! $49.95 postpaid USA. Disk only.

*MAE is a trademark of Eastern House. PAL is a trademark of Pro-Line. Commodore 64 & 128 are trademarks and CBM is a registered trademark of Commodore. Develop-64 is a trademark of French Silk. LADS is a trademark of Compute! Publications. Merlin is a trademark of Roger Wagner Publishing.



UNLEASH THE POWER OF YOUR COMMODORE 64 WITH THE ULTIMATE INTERFACE

- Control and monitor your home—appliances, lights and security system
- Intelligently control almost any device
- Connect to Analog-to-Digital Converters
- Control Robots
- Perform automated testing
- Acquire data for laboratory and other instrumentation applications
- Many other uses

Don't make the mistake of buying a limited capability interface. Investigate our universally applicable Dual 6522 Versatile Interface Adapter (VIA) Board, which plugs into the expansion connector and provides: Four 8-bit fully bidirectional I/O ports & eight handshake lines • Four 16 bit timer/counters • Full IRQ interrupt capability • Four convenient 16-pin DIP socket interface connections • Expandability up to four boards & sixteen ports.

ORDER NOW! Price $169, postpaid USA. Extensive documentation included. Each additional board $149.

COMMODORE 64™ SOURCE CODE!

"What's Really Inside the Commodore 64"

• Most complete available reconstructed assembly language source code for the C-64's Basic and Kernal ROMs, all 16K. • You will fully understand calls to undocumented ROM routines, and be able to effectively use them in your own programs. • Uses LABELS. Not a mere one-line disassembly. All branch targets and subroutine entry points are shown. • TABLES are fully sorted out and derived. • Completely commented, no gaps whatsoever. You will see and understand the purpose of every routine and every line of code! • Complete listing of equates to external label references. • Invaluable fully cross-referenced symbol table. Order C-64 Source. $29.95 postpaid USA.

PROFESSIONAL UTILITIES:

We personally use and highly recommend these two:

- PTD6510 Symbolic Debugger for C-64. An extremely powerful tool with capabilities far beyond a machine-language monitor. $49.95 postpaid USA.
- MAE64. Fully professional macro editor/assembler. $39.95 postpaid USA.

All orders shipped from stock within 24 hours via UPS. VISA/MasterCard welcomed. There will be a delay of 15 working days on orders paid by personal check.

SCHNEDLER SYSTEMS

1501 N. Ivanhoe, Dept. A-11, Arlington, VA 22205. Information/Telephone Orders (703) 237-4796

Reader Service No. 291

www.commodore.ca

FLOTSAM

Cheryl Peterson's guest editorial in the August Ahoy!, *concerning software piracy and its effect on the home computer industry, has drawn a wave of responses from both sides of the gangplank. A sampling is presented in this month's edition of* Flotsam.

Congratulations to Cheryl Peterson for her excellent guest editorial in the August issue of *Ahoy!* What she had to say about stealing programs is, I am afraid, all too true. As a teacher I have contact with a considerable number of students who use or would like to learn how to use computers. I have frequently been asked by students to make, or show them how to make, copies of copyrighted software. Many of these students, and other friends, think that there is nothing wrong with "sharing" special programs. I am amazed at the number of people who really don't understand that pirating software is a crime. I recently asked one young hacker if he would go into a local computer shop and walk out with a program under his jacket. He was shocked when I told him that making a copy of my *Print Shop* would be the same as taking it from the store.

A real challenge exists here for people who write for *Ahoy!*, or any of the other fine computer magazines, and teachers like myself, to educate computer users about the law and everyone's responsibility to respect the rights of others.

—Robert H. Croswell
Trappe, MD

To get straight to the point, I am a pirate. I have purchased programs at near full retail with absolutely no regret as to the price. The documentation was by itself well worth the money. I have better than one thousand pirated programs that I have traded for or copied using commercial copiers (most which were pirated themselves) or, using a machine language monitor, de-protected and copied. Several friends of mine and I trade programs between one another and have access to several networks that we can trade with. Several days ago I received 57 disks from our source in the midwest and another 35 from our west coast source. Some of these disks have as many as 6 complete programs. I am 39 years old and the average age of our group is 45, with members as young as 12. We are not in this for the profit, but rather as a means to build inexpensive libraries.

I have several suggestions that I would like to discuss with you. First of all, I wish to make an analogy to the record industry. A rock group spends as much as $250,000 to cut an album. Then they spend a tremendous amount of time and money promoting the album. The finished product is presented to the public for between $6 and $10. I think this is reasonable, and I have an extensive LP library to back this up. I could tape albums from my friends, but rarely want the entire album on one tape. The price is affordable, so I purchase what I want, and tape from that what I wish to listen to. The software industry, on the other hand, purchases many of its programs from pirate developers. They should not have any more expenses involved than does the record industry, but charge 3 to 10 times more for a program. Business programs require extensive documentation and I consider the price fair with all considerations. (If you own an IBM PC or equivalent, you get to pay a surcharge for the privilege of owning that brand.) Game programs, on the other hand, are sometimes of very poor quality and barely worth the price of a blank disk, much less $30. A lot of the time a person buys blindly, as most store clerks have trouble finding the price much less knowing anything about the content.

My solution is simple. Dongle protect the programs and keep the price around the $10 range for games, and under $50 for business programs. If the software industry is so greedy as to try to make their millions overnight, then they have to deal with the threat that people will steal their programs and give them away free as we do. A fair deal is a two way street. It must be profitable and affordable at the same time. Whatever happened to worth and value?

—Dave, a Pirate
Spokane, WA

To point the finger of guilt at the pirates and the home users who accept pirated software is legally fair, but ethically limited and based on naiveté. Let's point the finger at the real culprit, POPULAR COMPUTER MAGAZINES which advertise, for profit, "copy-all" programs. In the same issue of *Ahoy!* in which Cheryl's editorial appeared, I quickly found five (5) advertisements for disk copy-all programs, and two (2) advertisements for cartridge copiers. Isn't this the real nightmare? This practice is really setting the stage for and actively promoting the practice of piracy! How can a publisher justify running advertisements for expensive commercial software on one page and on the other page run advertisements for copy-all programs which will copy and pirate the commercial software on the previous page? The publishers will justify their actions by saying, "There is nothing wrong with advertising legitimate backup utilities!" However, software producers sell backup copies of their software at very reasonable costs.

So, Cheryl, let's shift the guilt from the bored and inquisitive pirate to the medium which has supplied the pirate with his tools. Hindsight is so clear...If computer magazine publishers had exercised better judgment a few years ago perhaps your "nightmare" would only be a mild concern and the home market would still be expanding.

—Richard N. Dawson
Marysville, MI

www.commodore.ca

I refuse to make or accept copies of commercial programs. My position is based upon knowledge of the effort, time, and money involved in developing software. Regrettably, the stealing, and it is stealing, is just one more example of the 'take care of number one' attitude so prevalent these days. What I'm trying to say is, the problem is not limited to stealing software. This is not to excuse the low level of morality but to put the problem in proper perspective.

The solution is some kind of method of physically preventing the making of copies, *including potential damage to the disk drive.* I realize that this is strong medicine, but the disease calls for such a cure. Obviously, this would create a potential problem in making legitimate copies for backup purposes, but certainly vendors can address this by offering two disks within a purchase. Such an arrangement is necessary to protect the purchaser in the event that the vendor drops the product or goes out of business.

—Charles A. Pocatille
Clifton, NJ

To friends who own Commodores and myself, this trend towards ignoring the home user is puzzling. Each of us spends a great deal of time on the computer. Admittedly, most of the programs in my library are ones I typed in from magazines. I can't afford to spend $40 or more on computer programs very often. But when I find something that I really want, I save my money and buy it.

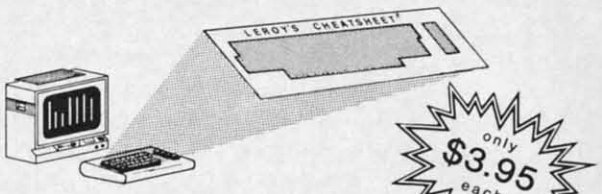
The small town computer shop near my home is selling out its Commodore software and will not restock when it is gone. The owner says that no one is buying it anymore. What is really happening is that no one is buying the games and no one can afford $75 for a utility. All this says to me is that the game players are played out and the only ones left are more serious users, who are beginning to turn to other sources for functional programs.

As long as magazines like *Ahoy!* are around, I say the heck with commercial software makers. They will be left in the dust of the wave of the future. A future with a chicken in every pot, and a computer in every home.

—John J. Hyland, III
Glassboro, NJ

One of the major problems we who live in small towns have is finding out about and locating new programs and products. We must rely on magazines and friends in other cities to pass new information on to us. Lately it seems that we are seeing fewer advertisements for products. What the problem is I really don't know, but if the dealers and software manufacturers would only find a way to let us know what products they have and provide us with information about them to allow us to make a choice on whether or not to purchase it, they and we would be better off. If we knew what was available, knew where to order it, and if vendors had a good return policy, we would probably be able to purchase more. Is the lack of advertisements because the publishers have raised their

LEROY'S CHEATSHEET®
KEYBOARD OVERLAYS



FOR COMMODORE 64™
(VIC-20 also available)

LEROY'S CHEATSHEETS® are plastic laminated keyboard overlays designed for use with popular software and hardware for Commodore's VIC-20 & C-64 computers.

These cut-it-out yourself overlays are designed to fit over the keyboard surrounding the keys with commands and controls grouped together for easy references.

LEROY'S CHEATSHEETS®
make life easier for you

WORD PROCESSORS
- ☐ Easy Script
- ☐ HES Writer
- ☐ Paper Clip
- ☐ Quick Brown Fox
- ☐ Script 64
- ☐ Speedscript (2.0 & 3.0 GAZETTE)
- ☐ Wordpro 3/PLUS
- ☐ Omniwriter
- ☐ Wordpro 64 (By Steve Punter)
- ☐ Write Now!/64
- ☐ Fleet System 2

SPREADSHEETS
- ☐ Calc Result (advanced)
- ☐ Calc Result (easy)
- ☐ Easy Calc
- ☐ Multiplan
- ☐ PractiCalc 64
- ☐ PRACTICALC II
- ☐ VIZASTAR

DATA BASES
- ☐ The Consultant (Delphi's Oracle)
- ☐ The Manager (CBM)
- ☐ SuperBase 64

PRINTERS
- ☐ Printer, (CBM) 1525, MPS 801, MPS 803
- ☐ Printer, (CBM) 1526, MPS 802
- ☐ Printer, (Epson) RX & FX 80 & 100
- ☐ Printer, (Gemini) 10X - 15X
- ☐ Printer, Okidata 92 & 93

AHY 11

MISCELLANEOUS
- ☐ Blanks (3 each - NOT Laminated)
- ☐ For The Beginner (using the 64)
- ☐ Sprites Only
- ☐ Flight Simulator II
- ☐ Doodle
- ☐ 1541 Disk
- ☐ Sky Travel

LANGUAGES & UTILITIES
- ☐ COMAL
- ☐ SIMON'S BASIC
- ☐ Basic
- ☐ Hesmon 64
- ☐ LOGO (sheet 1)
- ☐ LOGO (sheet 2)
- ☐ Pilot (CBM)

TERMINALS
- ☐ Vidtex
- ☐ The Smart 64 Terminal
- ☐ VIP Terminal
- ☐ Term 64

Qty.___ X $3.95 $ _____
Shipping & handling $ 1.00
6% sales tax (PA residents only) $ _____
TOTAL $ _____
U.S. FUNDS
NO COD's PLEASE

Dealer inquiries welcome

Name _____
Address _____
City _____ State _____ Zip _____

CHEATSHEET PRODUCTS INC
P.O. Box 111368 Pittsburgh PA. 15238
VISA master charge
(412) 781-1551

☐ Charge My VISA or Master Card NUMBER _____ EXPIRATION DATE ___/___

. . . OR SEE YOUR LOCAL DEALER !

Reader Service No. 279

www.commodore.ca

rates too high?

Yes, we do have members who copy programs for others. I would have to say that a large part of this problem would disappear if we had better access to the product and the cost was not so excessive. This copy problem is not as widespread as some magazine articles would have us believe, but it does exist. Relying on mail order almost exclusively is very difficult for us diehard Commodore users.

—Charles T. Cragg
Havasu Commodore Users Group
Lake Havasu City, AZ

Let's assume a small percentage of home computer users have to have everything they can get their hands on. I bet you that they will only play their favorites or use what they need. Is their illegally acquired, unused software really a threat or a loss to the manufacturer? That leaves a majority who are specifically inclined, meaning they only pirate what they want. Sadly I must admit there are also those who steal because they want to and have no intention of spending their own money. To those I say, you get back what you give.

Not all software is for everyone. I have pirated software that I didn't keep, and I have purchased software I wish I could get back my money for. It would be wonderful to try software before making the investment or have a 'satisfaction guaranteed or your money back' stipulation. Granted, some companies are service-oriented, but they are a minority.

Then there are the data management problems. Specifically, contingency planning. Most, but not all, companies make it an unnecessary inconvenience to acquire backup copies. Who wants to send a broken diskette away and then wait for a new one to be sent, especially if you're in the middle of a project?

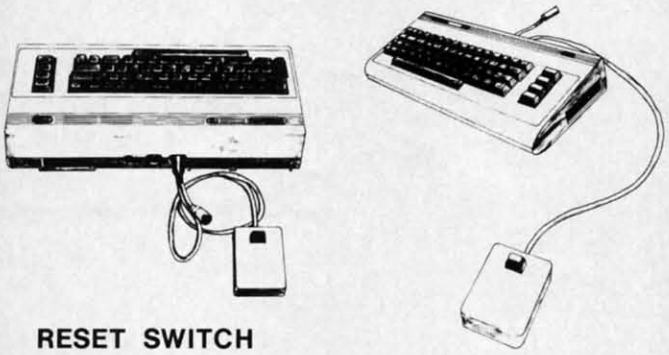
Non-customer service companies and dishonest people are escalating this piracy problem. I suggest my own policy, which is: if you have to, 'borrow' software temporarily, see if it meets your needs, and then purchase it legitimately and/or discard the pirated copy.

I must disagree with your negative prognosis of the Home Computer Industry. Just because the manufacturers aren't socking away a million plus dollars a year?

—R. Scot Derrer
Walnut Creek, CA

Why let the controversy end here? We'd be pleased to read, and possibly publish, your *views on software piracy and the future of the home computer industry—or on any other subject of interest to Commodore users. Address your correspondence to* Flotsam, *c/o* Ahoy!, *Ion International Inc., 45 West 34th St.—Suite 407, New York, NY 10001. While space limitations prevent us from publishing more than a small fraction of the letters we receive, we read each one, and answer personally where appropriate.*

PANIC BUTTON



RESET SWITCH

Only $19.95

BASIC RECOVER PROGRAMS INCLUDED
UNLOCKS "FROZEN" PROGRAMS
PLUGS INTO THE SERIAL AND USERPORT
ALLOWS FULL USE OF BOTH PORTS
EXTENDS THE LIFE OF YOUR COMPUTER
COMPATIBLE WITH COMMODORE 64, 128 AND VIC 20
1 YEAR WARRANTY

Mail Check Or Money Order To:
MICROVATIONS
5333 MISSION CENTER ROAD
SUITE 345
SAN DIEGO, CA 92108

CA Residents Add 6% Sales Tax
Shipping: Add $1.75 For Ground
COD $3.00 2nd Day Air

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____

Dealers Welcomed Phone (619) 291-2722

Reader Service No. 113

Ahoy! Kids America

WNYC pioneering radio program for children ages 5-12, "Small Things Considered," has gone national. The corporation for Public Broadcasting (CPB) announced that it will fund the nationwide production, broadcast and market testing of *Kids America* (formerly "Small Things Considered").

Kids America is a live, daily children's radio program produced by WNYC-AM, New York City. "Hopefully, *Kids America* will do for radio what Sesame Street and Mr. Rogers' Neighborhood did for public television," said Mary Perot Nichols, director WNYC.

AHOY! salutes WNYC management for this innovative program, C.P.B. for their foresight in supporting this project nationwide and Tom Trocco for his imaginative and unique teaching method used during his computer lesson segments (did you know that looping in computer programming is similar to playing hopscotch?).

The following areas will test market *Kids America:*

San Mateo, CA	KCSM—FM91	Cincinnati, OH	WUXU—FM91.7
Ames, IA	WOI—AM640	Rochester, NY	WXXI—FM91.5
Minneapolis, MN	WSJN—AM1330	Buffalo, NY	WBFO—FM88.7
Milwaukee, WI	WUWM—FM89.7	Boston, MA	WBUR—FM90.9
Macomb, IL	WIUM—FM91.3	New York, NY	WNYC—AM83.7

Time: 6:30 to 8:00 PM EST
Tune in and turn on to *Kids America.*

www.commodore.ca

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	

www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** This and the preceding 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  DE
·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   NE
·5010 DATA177,251,208,237,165,253,41,240,74,74  ME
·5011 DATA74,74,24,105,65,32,210,255,165,253    EF
·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       EF
·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
```


www.commodore.ca

```
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):GOTO3
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
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 121 and 122 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

MONKEY BUSINESS

FROM PAGE 37

STRICTLY RANDOM

```
·2 REM RUPERT REPORT #23:MONKEY BUSINESS  GL
·3 REM      >> STRICTLY RANDOM  <<        JJ
·4 REM                                    JD
·5 REM SIMULATE A MONKEY AT A NORMAL      JG
·6 REM 'ONE KEY PER CHARACTER'            FM
·7 REM TYPEWRITER KEYBOARD.               CE
·8 REM                                    JD
·10 A$=" ABCDEFGHIJKLMNOPQRSTUVWXYZ"      PC
·20 N=27*RND(0)+1 : PRINT MID$(A$,N,1);   CM
·30 GOTO 20                               OK
```

WEIGHTED KEYBOARD

```
·2 REM RUPERT REPORT #23:MONKEY BUSINESS  GL
·3 REM    >> WEIGHTED KEYBOARD <<         CB
·4 REM                                    JD
·5 REM SIMULATE A MONKEY AT A SPECIAL     CK
·6 REM KEYBOARD WITH NUMBERS OF KEYS      HN
·7 REM IN PROPORTION TO THE FREQUENCY OF  ML
·8 REM EACH LETTER'S USAGE IN ENGLISH.    CH
·9 REM                                    JD
·100 DATA 275," ",130,E,92,T,79,N         HL
·101 DATA 76,R,75,O,74,A,74,I,61,S        FF
·102 DATA 42,D,36,L,34,H,31,C,28,F        BK
·103 DATA 27,P,26,U,25,M,19,Y,16,G        MO
·104 DATA 16,W,15,V,10,B,5,X,3,Q          CG
·105 DATA 3,K,2,J,1,Z,-1,*                LG
·185 REM -- PACK STRINGS WITH LETTERS --  AK
·190 PRINT CHR$(147) "TOTAL = 0"          MM
·200 CT=0 : IX=0 : TTL=0                  EH
·210 READ N,C$ :NCT=0 :IF N=-1 THEN 300   IG
·220 A$(IX)=A$(IX)+C$                     DK
·230 NCT=NCT+1 : CT=CT+1 : TTL=TTL+1      AA
·240 IF CT=255 THEN CT=0 : IX=IX+1        FE
·250 IF NCT<N THEN 220                    KA
·260 PRINT CHR$(19)TAB(7) TTL : GOTO 210  FG
·295 REM   -- PICK AND PRINT LETTERS --   EI
·300 K=INT(RND(0)*TTL)                    OB
·310 X=INT(K/255) : CH=K-X*255+1          KF
·320 L$=MID$(A$(X),CH,1)                  JA
·330 PRINT L$;                            EJ
·340 GOTO 300                             BP
```

GETTING GRAPHIC

FROM PAGE 91

BALLBOUNCE.BAS

```
·10 REM **** BALLBOUNCE.BAS ****          IF
·20 PRINT CHR$(147):REM CLEAR SCREEN      EG
·30 BALL=81:SPACE=96:RULE=99:REM CODES TO
 PRINT THINGS ON THE SCREEN               GM
·40 FOR L=55616 TO 55975:POKE L,2:NEXT L:
 REM MAKE BALL RED                        NM
·50 FOR L=55976 TO 56015:POKE L,7:NEXT L:
 REM MAKE FLOOR YELLOW                    JI
·60 POKE 53281,0:POKE 53280,6:REM BLACK B
 ACKGROUND, BLUE BORDER                   NB
·70 PRINT CHR$(5):REM WHITE TEXT          JM
·80 PRINT:PRINT:PRINT "[5" "]FOLLOW THE B
 OUNCING BALL . . ."                      JG
·90 FOR L=1704 TO 1743:POKE L,RULE:NEXT L
 :REM DRAW FLOOR                          FK
·100 PSN=1664:CT=1:REM STARTING POSITION
 AND FRAME COUNTER                        FK
·110 FOR INC=1 TO 8:GOSUB 210:REM THIS LO
 OP DRAWS THE BALL GOING UP               EH
·120 PSN=PSN-40+1:REM THE BALL GOES UP    HL
·130 IF CT>40 THEN PSN=1344:CT=1:GOTO 150
 :REM BALL OFF SCREEN--BACK TO BEGINNING  FI
·140 NEXT INC                             GD
·150 FOR DEC=1 TO 8:GOSUB 210:REM THIS LO
 OP DRAWS THE BALL COMING DOWN            MJ
·160 PSN=PSN+41:REM THE BALL COMES DOWN   HK
·170 IF CT>40 THEN 100:REM BALL OFF SCREE
 N -- LOOP BACK                           FA
·180 NEXT DEC                             EN
·190 GOTO 110:REM DONE--START AGAIN       KC
·200 REM **** PRINT BALL ON SCREEN ****   PI
·210 POKE PSN,BALL                        MH
·220 FOR L=1 TO 50:NEXT L                 OH
·230 POKE PSN,SPACE                       CC
·240 CT=CT+1:RETURN                       HB
·250 END                                  IC
```

BLACKBOARD.BAS

```
·10 REM *** "BLACKBOARD.BAS" *******      OJ
·20 BASE=2*4096:POKE 53272,PEEK(53272)OR8
 :REM PUT HIGH-RES MAP AT 8192            CP
·30 POKE 53265,PEEK(53265)OR32:REM ENTER
 HIGH-RES BIT-MAP MODE                    NJ
·40 FOR I=BASE TO BASE+7999:POKE I,0:NEXT
 :REM CLEAR BIT MAP                       KJ
·50 FOR I=1024 TO 2023:POKE I,16:NEXT I:R
 EM BLACK BACKGROUND, WHITE LINE          NI
·60 GOTO 200                              BO
·80 REM ****** PLOT ROUTINE **********    KH
·90 CHAR=INT(HPSN/8)                      PA
·100 ROW=INT(VPSN/8)                      LI
·110 LINE=VPSN AND 7                      PM
·120 BYTE=BASE+ROW*320+8*CHAR+LINE        KG
·130 BIT=7-(HPSN AND 7)                   OP
·140 POKE BYTE,PEEK(BYTE) OR (2[UPARROW]B
 IT)                                      DO
·150 RETURN                               IM
·200 REM **** DRAW VERTICAL LINE *******  DE
·220 FOR VPSN=0 TO 199:REM PLOT LINE FROM
  TOP TO BOTTOM OF SCREEN                 IG
```


www.commodore.ca

```
·225 FOR HPSN=159 TO 160                                    EE
·230 GOSUB 80                                               PM
·240 NEXT HPSN:NEXT VPSN                                    NP
·245 REM *** DRAW HORIZONTAL LINE ******                    NA
·250 VPSN=100:REM HALFWAY DOWN SCREEN                       LK
·260 FOR HPSN=0 TO 319:REM PLOT LINE FROM
  LEFT SIDE TO RIGHT SIDE OF SCREEN                         HB
·270 GOSUB 80                                               PM
·280 NEXT HPSN                                              LL
·290 GOTO 290                                               CH
```

INSTANT BUG REPELLENT

FROM PAGE 73

```
·10 SA=49152                                                FI
·20 FORI=100TO148:ZZ=0:FORJ=1TO8:READA:PO
 KESA,A:SA=SA+1:ZZ=ZZ+A:NEXTJ                               FO
·30 READA:IF ZZ=ATHEN CK=CK+ZZ:NEXT I:GOT
 050                                                        KN
·40 PRINT" ERROR IN LINE #";I:END                           AM
·50 IF SA-CK=1377 THEN SYS49152:NEW                         EL
·100 DATA 169,90,141,2,3,169,192,141,907                    KI
·101 DATA 3,3,160,0,185,37,192,240,820                      NG
·102 DATA 6,32,210,255,200,208,245,169,13
 25                                                         LF
·103 DATA 1,133,251,169,8,133,252,165,111
 2                                                          DG
·104 DATA 0,141,167,2,96,18,32,73,529                       ND
·105 DATA 78,83,84,65,78,84,32,66,570                       FP
·106 DATA 85,71,32,82,69,80,69,76,564                       LG
·107 DATA 76,69,78,84,32,13,0,40,392                        AK
·108 DATA 104,170,104,168,104,76,131,164,
 1021                                                       EJ
·109 DATA 160,0,177,251,240,1,96,200,1125                   IJ
·110 DATA 177,251,240,1,96,104,104,76,104
 9                                                          IN
·111 DATA 63,192,72,152,72,138,72,8,769                     EH
·112 DATA 165,157,208,3,76,63,192,169,103
 3                                                          PE
·113 DATA 1,133,251,169,8,133,252,169,111
 6                                                          DO
·114 DATA 0,133,254,133,255,165,20,133,10
 93                                                         LM
·115 DATA 254,165,21,133,255,32,72,192,11
 24                                                         KM
·116 DATA 160,2,177,251,197,254,240,6,128
 7                                                          ND
·117 DATA 32,105,193,24,144,239,200,177,1
 114                                                        HB
·118 DATA 251,197,255,240,6,32,105,193,12
 79                                                         FL
·119 DATA 24,144,226,169,1,133,254,169,11
 20                                                         BD
·120 DATA 0,133,255,165,251,24,105,4,937                    IC
·121 DATA 133,251,165,252,105,0,133,252,1
 291                                                        CH
·122 DATA 169,0,160,0,169,0,141,167,806                     KI
·123 DATA 2,240,38,133,255,177,251,208,13
 04                                                         KI
·124 DATA 27,165,255,41,240,74,74,74,950                    JJ
·125 DATA 74,24,105,129,141,2,4,165,644                     IO
·126 DATA 255,41,15,24,105,129,141,3,713                    DL
·127 DATA 4,76,15,193,230,254,32,84,888                     FE
·128 DATA 193,165,255,160,0,170,177,251,1
 371                                                        BD
·129 DATA 201,34,208,8,173,167,2,73,866                     KA
·130 DATA 255,141,167,2,173,167,2,208,111
 5                                                          IO
·131 DATA 12,177,251,201,32,208,6,198,108
 5                                                          IH
·132 DATA 254,138,76,187,192,138,113,251,
 1349                                                       MA
·133 DATA 69,254,170,138,76,187,192,169,1
 255                                                        EG
·134 DATA 160,141,0,4,169,186,141,1,802                     LA
·135 DATA 4,160,0,185,127,193,240,6,915                     CE
·136 DATA 153,4,4,200,208,245,234,165,121
 3                                                          AJ
·137 DATA 20,133,99,165,21,133,98,162,831                   KH
·138 DATA 144,56,32,73,188,32,221,189,935                   PL
·139 DATA 162,0,189,1,1,240,10,41,644                       NJ
·140 DATA 63,9,128,157,12,4,232,208,813                     AL
·141 DATA 241,169,160,157,12,4,32,91,866                    NH
·142 DATA 193,76,63,192,230,251,208,2,121
 5                                                          CG
·143 DATA 230,252,96,173,134,2,160,0,1047                   JH
·144 DATA 153,0,216,200,192,21,144,248,11
 74                                                         KG
·145 DATA 96,160,0,177,251,170,32,84,970                    DG
·146 DATA 193,177,251,133,252,138,133,251
 ,1528                                                      KO
·147 DATA 96,0,0,0,0,0,0,160,256                            AN
·148 DATA 140,137,142,133,160,163,160,0,1
 035                                                        HH
```

INTERRUPTING YOUR WAY TO FAST MOTION

FROM PAGE 18

BORDER INTERRUPT

```
·1 REM PROGRAMMING IN THE IRQ INTERRUPT                     GH
·7 REM RUN THIS PROGRAM, THEN TYPE CHARAC
 TERS IN UPPER LEFTHAND CORNER OF SCREEN                    AB
·8 REM THE BORDER COLOR WILL CHANGE, DEPE
 NDING ON SCREEN CODE OF CHARACTER                          DB
·9 REM 13-BYTE PROGRAM SETS UP THE INTERR
 UPT: 11-BYTE PROGRAM RUNS IT                               FE
·10 FOR I=5011 TO 5023:READ A:POKE I,A:NE
 XT                                                         CK
·20 FOR I=5000 TO 5010:READ A:POKE I,A:NE
 XT                                                         NK
·30 SYS 5011                                                GH
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 121 and 122 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·40 END                                                    IC
·47 REM  *** SYS CALL TO SET INTERRUPT VE
CTOR                                                       JA
·48 REM BLOCK INTERRUPTS, SET VECTOR ADDR
ESS (LOW, HIGH), ENABLE INTERRUPTS                         GI
·49 REM  SEI LDA#136 STA 788  LDA#19 STA
789 CLI RTS                                                BH
·50 DATA 120,169,136,141,20,3,169,19,141,
21,3,88,96                                                 EH
·57 REM  *** ACTUAL INTERRUPT ROUTINE                      EI
·58 REM GET FIRST SCREEN CHARACTER; USE L
OW NYBBLE TO SET BORDER COLOR                              NG
·59 REM LDA 1024 AND#15 STA 53280 JMP 599
53                                                         NJ
·60 DATA 173,0,4,41,31,141,32,208,76,49,2
34                                                         MG
```

STARSHIP

```
·1 REM      ***  STARSHIP  ***                             OE
·2 REM MACHINE LANGUAGE SPRITE MOVEMENT                    GK
·4 REM ** MOVE TOP OF MEMORY TO MAKE ROOM
 FOR VIDEO BLOCK AT 32768                                  CA
·5 POKE 55,255:POKE 56,127:POKE 643,255:P
OKE 644,127:PRINT "[CLEAR]"                                KE
·6 REM ** DECLARE PRIMARY VARIABLES                        GG
·7 F$=" ":C0%=0:C1%=0:C2%=0:C3%=0:C4%=0:C
5%=0:C6%=0:SP%=0                                           NO
·8 DEF FN PG(X)=INT(X/256):DEF FN LO(X)=X
-256*(INT(X/256))                                          EF
·9 REM      *** SET-UP SUBROUTINES ***                     OF
·10 GOSUB 90:GOSUB 600:REM (VIDEO MEMORY)                  PO
·12 PRINT "[CLEAR]";:GOSUB 95:GOSUB 700:G
OSUB 2000:REM (MACHINE LANGUAGE AND SPRI
TES)                                                       PH
·13 GOSUB 900:REM (SPRITE POSITIONS AND G
AME SCREEN)                                                HN
·14 SYS 38067:REM (START INTERRUPT!)                       BM
·15 REM ** ENABLE SPRITES                                  OD
·16 FOR I=0 TO 29:NEXT:POKE ES,31:REM (SP
RITES 0-4)                                                 NF
·17 GOTO 100                                               CF
·19 REM ** LOAD SUBROUTINE                                 GD
·20 FOR I=XB TO XE:READ A:POKE I,A:NEXT:P
RINT "[RVSON].[RVSOFF]";:RETURN                            MG
·89 REM TURN OFF SCREEN                                    GF
·90 POKE 53265,0:RETURN                                    MO
·94 REM TURN ON SCREEN (AND EXTENDED BACK
GROUND COLOR TEXT MODE)                                    JD
·95 POKE 53265,91:RETURN                                   PH
·98 REM      *** ACTION LOOP ***                           KJ
·100 K=PEEK(653):IF K=7 THEN 300                           OF
·110 FOR I=1 TO 4:SP%=I:SYS 38336:NEXT                     AN
·115 PRINT "[HOME][5" "][HOME][DOWN][6" "
][HOME][DOWN][DOWN][6" "][HOME][3"[DOWN]
"][5" "]"                                                  JC
·120 ON C0% GOSUB 200:C0%=0:C4%=0:C5%=0:C
6%=0                                                       LK
·125 FOR I=1 TO 4:SP%=I:SYS 38336:NEXT                     AN
·130 IF C1%>0 THEN GOSUB 220:C1%=0                         KG
·135 IF C2%>0 THEN GOSUB 240:C2%=0                         OC
·140 IF C3%>0 THEN GOSUB 260:C3%=0                         LC
·196 GOTO 100                                              CF
·199 REM ** FIREBUTTON ROUTINE GOES HERE                   FE
·200 PRINT "[HOME]F"                                       DJ
·201 IF C4%=1 THEN PRINT "[HOME][RIGHT][R
IGHT]!"                                                    BG
·202 IF C5%>1 THEN PRINT "[HOME][4"[RIGHT
]"]S"                                                      JA
·203 IF C6%=1 THEN PRINT "[HOME][6"[RIGHT
]"]F"                                                      IC
·205 RETURN                                                IM
·220 PRINT "[HOME][3"[DOWN]"]EW"STR$(C1%)
:RETURN                                                    DD
·239 REM ** SPR/SPR COLLISION ROUTINE                      NC
·240 PRINT "[HOME][DOWN]CS"STR$(C2%):RETU
RN                                                         BD
·259 REM ** SPR/FORE COLLISION ROUTINE                     AG
·260 PRINT "[HOME][DOWN][DOWN]CF"STR$(C3%
):RETURN                                                   CH
·296 RETURN                                                IM
·298 REM    *** END HANDLING ***                           AB
·299 REM ** PUT VIDEO MEMORY BACK TO FIRS
T BLOCK, AND SCREEN MEMORY TO 1024                         EG
·300 GOSUB 90:POKE ES,0:REM DISABLE SPRIT
ES                                                         LO
·304 REM RESTORE VIDEO/SCREEN MEMORY                       AK
·305 POKE 56578,PEEK(56578)OR3:POKE 56576
,(PEEK(56576)AND 252)OR 3                                  MB
·306 I=PEEK(53272):POKE 53272,20:K=PEEK(6
48):POKE 648,4                                             JJ
·310 GOSUB 95:PRINT "[CLEAR]QUIT? ([RVSON
][s Y][RVSOFF] OR [RVSON][s N][RVSOFF])"
:PRINT:PRINT                                               EP
·315 GET A$:IF A$="" THEN 315                              HO
·320 IF A$="Y" THEN PRINT "[HOME]SO LONG,
 STAR PILOT!":GOTO 370                                     AL
·325 GOSUB 90:POKE 56578,PEEK(56578)OR3:P
OKE 56576,(PEEK(56576)AND 252)OR 1                         PN
·330 POKE 53272,I:POKE 648,K:POKE ES,31:G
OSUB 95:GOTO 100                                           FK
·370 FOR I=0 TO 599:NEXT                                   MF
·379 REM ** REENABLE SHIFT/COMMODORE AND
RUN-STOP/RESTORE                                           FB
·380 POKE 657,0:POKE 792,71:POKE 808,237                   PK
·390 SYS 65126                                             KI
·598 REM    *** ARRANGE MEMORY ***                         KB
·599 REM USE THIRD VIDEO BLOCK (32768 TO
49151), SO ROM CHARACTER SET IS USABLE.                    AL
·600 VB=32768:POKE 56578,PEEK(56578)OR3:P
OKE 56576,(PEEK(56576)AND 252)OR 1                         CN
```


www.commodore.ca

```
·601 REM ** TELL VIC-2 WHERE SCREEN IS WI THOUT CHANGING CHARACTER SET LOCATION
·602 SB=0:POKE 53272,(SB*16)+4:SB=VB+1024 *SB
·603 REM ** TELL BASIC WHERE SCREEN IS
·604 BB=SB/256:POKE 648,BB
·608 REM     *** REGISTER ADDRESSES ***
·611 REM ** SPRITE COLOR TABLE
·612 CT(0)=53287:FOR I=1 TO 7:CT(I)=CT(I-1)+1:NEXT
·613 REM ** SPRITE HORIZONTAL POSITION TABLE (LOW BYTES)
·614 HT(0)=53248:FOR I=1 TO 7:HT(I)=HT(I-1)+2:NEXT
·615 REM ** SPRITE VERTICAL POSITION TABLE
·616 VT(0)=53249:FOR I=1 TO 7:VT(I)=VT(I-1)+2:NEXT
·617 REM ** SPRITE HORIZONTAL HIGH-BIT REGISTER
·618 HR=53264
·619 REM ** SPRITE ENABLE REGISTER
·620 ES=53269
·621 REM ** VERTICAL EXPANSION REGISTER (1=DOUBLE HEIGHT)
·622 VE=53271
·623 REM ** HORIZONTAL EXPANSION REGISTER (1=DOUBLE WIDTH)
·624 HE=53277
·625 REM ** SPRITE PRIORITY REGISTER (1=SPRITE IS IN FRONT OF FOREGROUND)
·626 PR=53275
·627 REM ** MULTICOLOR ENABLE REGISTER (1=MULTI-COLOR ENABLED)
·628 EM=53276
·629 REM ** SPRITE MULTICOLOR COLOR REGISTERS
·630 MR=53285:REM ('01' REGISTER: ADD 1 TO MR FOR '11' REGISTER)
·633 REM ** SET-BIT AND CLEAR-BIT VALUES
·634 BS(0)=1:FOR I=1 TO 7:BS(I)=2*BS(I-1):NEXT
·635 FOR I=0 TO 7:BC(I)=255-BS(I):NEXT
·638 REM     *** INITIALIZE VALUES ***
·639 REM ** FOREGROUND COLOR
·640 POKE 53281,0:PRINT "[CLEAR][c 7]";:REM (LIGHT BLUE)
·641 REM ** BACKGROUND COLOR
·642 POKE 53281,0 :REM (BLACK)
·643 REM ** BORDER COLOR
·644 POKE 53280,0:REM (BLACK)
·645 REM ** SPRITE COLORS (DEFAULTS: WHI, RED,L-GRN,PUR,GRN,BLU,YEL,M-GRAY)
·646 POKE CT(0),7:POKE CT(1),5:POKE CT(2),2:POKE CT(3),6:POKE CT(4),12
·647 REM ** SET PRIORITY
·648 POKE PR,0:REM (ALL IN FRONT)
·649 REM ** SET HORIZONTAL SIZES
·650 POKE HE,0:REM (ALL SMALL)
·651 REM ** SET VERTICAL SIZES
·652 POKE VE,0:REM (ALL SMALL)
·653 REM ** ENABLE SPRITES
·654 POKE ES,0:REM (LEAVE THEM OFF FOR NOW)
·655 REM ** ENABL MULTICOLOR FOR SPR 1-4
·656 POKE EM,30
·657 REM ** SET MULTI-COLORS 1 AND 3 (1=LIGHT GREY, 3=YELLOW)
·658 POKE MR,15:POKE MR+1,7
·659 REM       *** ML TABLE SETUP ***
·660 REM ** ANIMATION TIMER (1=FASTEST)
·661 POKE 37920,4:POKE 37921,4
·662 REM ** ANIMATION COUNTER (ALWAYS 1)
·663 POKE 37922,1
·664 REM ** ANIMATE SPRITE 0?  (1=YES)
·665 POKE 37923,0
·666 REM ** MOVEMENT TIMER (NUMBER OF INTERRUPTS BETWEEN MOVES [1=FASTEST])
·667 POKE 37924,1:POKE 37925,1
·668 REM ** ALL SPRITES WRAP AT SCREEN EDGE? (1=YES)
·669 POKE 37936,1
·670 REM ** SPRITE 0 BOUNCE OFF SPRITES? (1=YES)
·671 POKE 37940,1
·672 REM ** SPRITE 0 BOUNCE OFF FOREGROUND? (1=YES)
·673 POKE 37941,0
·674 REM ** GO-SPEED TIMER (NUMBER OF SPRITE 0 MOVES PER INTERRUPT [1=SLOWEST])
·675 POKE 37926,3:POKE 37928,3
·676 REM CLEAR FLAGS
·677 POKE 37927,0:POKE 37935,0:POKE 37943,0
·678 REM ** EXTENDED BACKGROUND COLORS
·679 POKE 53282,1:POKE 53283,7:POKE 53284,9
·683 REM ** SPRITE 0-7 BIT TABLE
·684 X=1:FOR I=37962 TO 37969:POKE I,X:X=X*2:NEXT
·693 REM     *** SAFETY PROCEDURES ***
·694 POKE 657,128:REM DISABLE SHIFT/COMMODORE CHARACTER SET SWITCH
·695 REM POKE 808,234:POKE 792,193:REM DISABLE STOP AND STOP/RESTORE
·696 RETURN
·699 REM *** INTRO SCREEN ***
·700 PRINT "[CLEAR][5"[DOWN]"]"TAB(12)"[s S][s T][s A][s R][s S][s H][s I][s P][S S][s C][s A][s P][s T][s A][s I][s N]"
·710 PRINT:PRINT TAB(6)"[RVSON][s Y][s O][s U][s R][SS][s C][s R][s A][s F][s T][SS][s I][s S][SS][s B][s E][s I][s N][s G][SS][s P][s R][s E][s P][s A][s R][s E][s D][RVSOFF]":PRINT:PRINT
·796 RETURN
```


www.commodore.ca

```
·898 REM     *** SPRITE POSITIONS ***          KA
·899 REM ** POSSIBLE POSITIONS DIM'ED          MK
·900 FOR I=1 TO 4:POKE HT(I),20+INT(RND(9
)*220)                                         DK
·901 POKE VT(I),50+INT(RND(9)*190):NEXT        JM
·902 POKE HR,0                                 IJ
·909 REM ** PUT STARS ON THE SCREEN            MF
·910 PRINT "[CLEAR]";:FOR I=0 TO 49:POKE
 VB+INT(RND(9)*1024),46:NEXT                   EO
·915 FOR I=0 TO 8:POKE VB+INT(RND(9)*1024
),42:NEXT                                      IK
·919 REM ** STARSHIP POSITION                  LP
·920 POKE 53248,175:POKE 53249,150             AE
·921 REM ** STARSHIP DIRECTION                 CN
·922 POKE VB+1016,16                           ED
·946 RETURN                                    IM
·1998 REM    ***  MACHINE LANGUAGE  ***       OA
·1999 REM ** STARTUP SYS ROUTINE               JO
·2000 POKE 37888,PEEK(788):POKE 37889,PEE
K(789)                                         LB
·2002 XB=38067:XE=38079:GOSUB 20               BF
·2003 REM BLOCK INTERR, SET VECTOR TO ANI
MATION SHELL,ENABLE INTERR                     CG
·2004 REM SEI LDA #0 STA 788 LDA #149 STA
 789 CLI RTS                                   CM
·2005 DATA 120,169.0,141,20,3,169,149,141
,21,3,88,96                                    HC
·2019 REM   ***  ANIMATION SHELL  ***          KJ
·2020 XB=38144:XE=38176:GOSUB 20               DD
·2021 REM SEE IF TIMER CALLS FOR ANIMATIO
N OR MOVEMENT                                  CM
·2022 REM DEC 37920 BEQ+3 JMP(37898)           EG
·2023 DATA 206,32,148,240,3,108,10,148         BE
·2024 REM RESET ANIMATION TIMER                BA
·2025 REM LDA 37921 STA 37920                  IJ
·2026 DATA 173,33,148,141,32,148               ED
·2027 REM GET NEXT STEP IN ANIMATION SEQU
ENCE                                           ON
·2028 REM DEC 37922 BNE+5 LDA#8 STA 37922
 LDX 37922 DEX                                 CG
·2029 DATA 206,34,148,208,5,169,8,141,34,
148,174,34,148,202                             OK
·2030 REM IF CALLED FOR, ANIMATE #0            GE
·2031 REM LDA 37923 BEQ+6                      MH
·2032 DATA 173,35,148,240,6                    LH
·2033 REM ANIMATE ALL SPRITES                  KN
·2034 REM LDA ANIM.SEQ.TAB,X STA SPRITE.S
HAPE.TAB                                       FC
·2035 A=192:B=248:FOR I=38177 TO 38219 ST
EP 6:POKE I,189:POKE I+1,A:POKE I+2,148        MN
·2036 POKE I+3,141:POKE I+4,B:POKE I+5,13
1:A=A+8:B=B+1:NEXT                             BG
·2037 REM EXIT THROUGH MOVEMENT HANDLER        GA
·2038 REM JMP (37898)                          PF
·2039 POKE 38225,108:POKE 38226,10:POKE 3
8227,148                                       KK
·2049 REM   *** MOVEMENT COUNTER ***           HI
·2050 XB=38272:XE=38288:GOSUB 20               EL
·2051 X=38272:POKE 37896,FN LO(X):POKE 37
897,FN PG(X):REM SET COUNTER ADDRESS           AA
·2052 POKE 37898,FN LO(X):POKE 37899,FN P
G(X):REM MOVE.VECT.=JOYDIR                     CD
·2053 REM DECREMENT TIMER; IF NOT 0, GO F
INISH UP                                       LI
·2054 REM DEC 37924 BEQ+3 JMP(37900)           BI
·2055 DATA 206,36,148,240,3,108,12,148         GC
·2056 REM RESET TIMER AND JUMP TO READ RO
UTINE THROUGH VECTOR SET FROM BASIC            CM
·2057 REM LDA 37925 STA 37924 JMP(37890)       KP
·2058 DATA 173,37,148,141,36,148,108,2,14
8                                              EA
·2059 REM ** BITSET SUBROUTINE **              EI
·2060 XB=38314:XE=38323:GOSUB 20               BM
·2061 REM GET BITMASK AND PUT IT IN HORIZ
ONTAL HI-BIT REGISTER                          PE
·2062 REM LDA 37962,Y ORA 53264 STA 53264
 RTS                                           EL
·2063 DATA 185,74,148,13,16,208,141,16,20
8,96                                           EH
·2069 REM ** BITCLEAR SUBROUTINE **            KG
·2070 XB=38324:XE=38335:GOSUB 20               CK
·2071 REM GET BITMASK, REVERSE IT, AND PU
T IT IN HORIZONTAL HI-BIT REGISTER             KM
·2072 REM LDA 37962,Y EOR#255 AND 53264 S
TA 53264 RTS                                   JP
·2073 DATA 185,74,148,73,255,45,16,208,14
1,16,208,96                                    CF
·2099 REM      *** XMOVE ***                   EL
·2100 XB=38400:XE=38467:GOSUB 20               CJ
·2101 REM TEST FOR UPMOVE                      JG
·2102 REM LDA#1 AND 37963,Y BEQ+3 JSR 385
28                                             PF
·2103 DATA 169,1,57,75,148,240,3,32,128,1
50                                             GE
·2107 REM TEST FOR DOWNMOVE                    DO
·2108 REM LDA#2 AND 37963,Y BEQ+3 JSR 385
92                                             IJ
·2109 DATA 169,2,57,75,148,240,3,32,192,1
50                                             CM
·2117 REM TEST FOR LEFTMOVE AND HI-BIT         ME
·2118 REM LDA#4 AND 37963,Y BEQ+17 LDA 37
962,Y AND 53264 BEQ+6                          JB
·2119 DATA 169,4,57,75,148,240,17,185,74,
148,45,16,208,240,6                            KM
·2120 REM DO EITHER LEFTSET OR LEFTCLEAR       AM
·2121 REM JSR 38656 JMP 38444 JSR 38720        AH
·2122 DATA 32,0,151,76,44,150,32,64,151        FE
·2126 REM TEST FOR RIGHTMOVE AND HI-BIT        AN
·2127 REM LDA#8 AND 37963,Y BNE+1 RTS LDA
 37962,Y AND 53264 BEQ+4                       FD
·2128 DATA 169,8,57,75,148,208,1,96,185,7
4,148,45,16,208,240,4                          HN
·2129 REM DO EITHER RIGHTSET OR RIGHTCLEA
R                                              JL
·2130 REM JSR 38784 JMP 38515 JSR 38848 R
TS                                             CH
```


www.commodore.ca

```
·2131 DATA 32,128,151,96,32,192,151,96                    OB
·2139 REM   *** UPMOVE SUBROUTINE ***                     NE
·2140 XB=38528:XE=38561:GOSUB 20                          BL
·2141 REM GET VERT.LOC.,DECREMENT,CHECK E
 DGE,STORE NEW VERT.LOC.                                  BH
·2142 REM LDX 53249,Y DEX TXA CMP 37978,Y
  BNE+3 JSR 38546 TXA STA 53249,Y RTS                     FB
·2143 DATA 190,1,208,202,138,217,90,148,2
 08,3,32,146,150,138,153,1,208,96                         BB
·2144 REM * TOPCHECK                                      BC
·2145 REM LDA 37936 BNE+4 INX JMP 38391 L
 DX 37979,Y DEX JMP 38391                                 PK
·2146 DATA 173,48,148,208,4,232,76,247,14
 9,190,91,148,202,76,247,149                              OB
·2159 REM   *** DOWNMOVE SUBROUTINE ***                   PI
·2160 XB=38592:XE=38625:GOSUB 20                          CJ
·2161 REM GET VERT.LOC.,INCREMENT,CHECK E
 DGE,STORE NEW VERT.LOC.                                  FD
·2162 REM LDX 53249,Y INX TXA CMP 37979,Y
  BNE+3 JSR 38610 TXA STA 53249,Y RTS                     FK
·2163 DATA 190,1,208,232,138,217,91,148,2
 08,3,32,210,150,138,153,1,208,96                         NF
·2164 REM * BOTTOMCHECK                                   PG
·2165 REM LDA 37936 BNE+4 DEX JMP 38391 L
 DX 37978,Y INX JMP 38391                                 HD
·2166 DATA 173,48,148,208,4,202,76,247,14
 9,190,90,148,232,76,247,149                              OM
·2179 REM *** LEFTMOVE (HI BIT SET) ***                   KO
·2180 XB=38656:XE=38669:GOSUB 20                          GC
·2181 REM GET HORIZ LOC, DECREMENT, CHECK
  CROSSOVER, STORE AND RETURN                             DJ
·2182 REM LDX 53248,Y DEX BPL+3 JSR 38324
  TXA STA 53248,7 RTS                                     IE
·2183 DATA 190,0,208,202,16,3,32,180,149,
 138,153,0,208,96                                         DC
·2199 REM *** LEFTMOVE (HI BIT CLR) ***                   KJ
·2200 XB=38720:XE=38756:GOSUB 20                          LH
·2201 REM GET HORIZONTAL POSITION, DECREM
 ENT; IF EDGE, MOVE AND LEAVE                             BC
·2202 REM LDX 53248,Y DEX TXA CMP 37994,Y
  BNE+3 JSR 38738 TXA STA 53248,Y RTS                     OC
·2203 DATA 190,0,208,202,138,217,106,148,
 208,3,32,82,151,138,153,0,208,96                         FL
·2204 REM *** LEFT CHECK                                  IA
·2205 REM LDA 37936 BNE+4 INX JMP 38391 L
 DX 37995,Y DEX JSR 38314 JMP 38391                       BE
·2206 DATA 173,48,148,208,4,232,76,247,14
 9,190,107,148,202,32,170,149,76,247,149                  PA
·2219 REM *** RIGHTMVE (HI BIT SET) ***                   CM
·2220 XB=38784:XE=38820:GOSUB 20                          CL
·2221 REM GET HORIZONTAL POSITION, INCREM
 ENT; IF EDGE, MOVE AND LEAVE                             DI
·2222 REM LDX 53248,Y INX TXA CMP 37932,Y
  BNE+3 JSR 38802 TXA STA 53248,Y RTS                     EM
·2223 DATA 190,0,208,232,138,217,107,148,
 208,3,32,146,151,138,153,0,208,96                        BE
·2224 REM *** RIGHT CHECK                                 OB
·2225 REM LDA 37936 BNE+4 DEX JMP 38391 L
 DX 37994,Y INX JSR 38324 JMP 38391                       DG
·2226 DATA 173,48,148,208,4,202,76,247,14
 9,190,106,148,232,32,180,149,76,247,149                  MI
·2239 REM *** RIGHTMVE (HI BIT CLR) ***                   LP
·2240 XB=38848:XE=38861:GOSUB 20                          CL
·2241 REM GET HORIZ LOC, INCREMENT, CHECK
  CROSSOVER, STORE AND RETURN                             ND
·2242 REM LDX 53248,Y INX BNE+3 JSR 38314
  TXA STA 53248,Y RTS                                     DI
·2243 DATA 190,0,208,232,208,3,32,170,149
 ,138,153,0,208,96                                        ND
·2399 REM *** BASIC MOVEMENT HANDLER ***                  AN
·2400 XB=38336:XE=38346:GOSUB 20                          BL
·2401 REM SET Y TO OFFSET OF SELECTED SPR
 ITE                                                      DD
·2402 REM LDY#59 LDA(45)Y TAX LDY 37944,X
  JMP 38400                                               AF
·2403 DATA 160,59,177,45,170,188,56,148,7
 6,0,150                                                  OI
·2469 REM ** REPORT NON-SPRITE-0 WRAPS AN
 D EDGES TO BASIC                                         PI
·2470 XB=38391:XE=38399:GOSUB 20                          EC
·2471 REM WAS IT SPRITE 0? IF NOT, REPORT
  WRAP                                                    OF
·2472 REM CPY#0 BNE+1 RTS STY 37943 RTS                   OL
·2473 DATA 192,0,208,1,96,140,55,148,96                   OO
·2499 REM   *** READ JOYSTICK ***                         BO
·2500 XB=38912:XE=38972:GOSUB 20                          FC
·2501 REM SET READ VECTOR TO POINT TO JOY
 STICK ROUTINE                                            MO
·2502 X=38912:POKE 37890,FN LO(X):POKE 37
 891,FN PG(X):REM (REM THIS TO USE KEYB.)                 DC
·2506 REM GET JOYSTICK 2; STORE IT, CHECK
  FOR FIREBUTTON, REPORT IF PRESSED                       ON
·2507 REM LDA 56320 STA 37933 AND#16 BNE+
 8 LDA#1 STA 37927 JSR 39168                              JH
·2508 DATA 173,0,220,141,45,148,41,16,208
 ,8,169,1,141,39,148,32,0,153                             AD
·2509 REM IS MOVEMENT CALLED FOR? IF SO,
 SET JOYDIR, SET SHAPE 0                                  AC
·2510 REM LDA 37933 AND#15 CMP#15 BNE+3 J
 MP(37900) EOR#15 STA 37963 JSR 39040                     LK
·2511 DATA 173,45,148,41,15,201,15,208,3,
 108,12,148,73,15,141,75,148,32,128,152                   AN
·2512 REM MOVE ONCE, THEN POINT MOVE.VECT
 .TO MOVEMENT HANDLER                                     HN
·2513 REM LDY#0 JSR 38400 JSR 39072 LDA 3
 7892 STA 37898                                           OG
·2514 DATA 160,0,32,0,150,32,160,152,173,
 4,148,141,10,148                                         OM
·2515 REM LDA 37893 STA 37899 JMP(37900)                  MK
·2516 DATA 173,5,148,141,11,148,108,12,14
 8                                                        EH
·2529 REM *** INTERR.MOVE.HANDLER ***                     PG
·2530 XB=38976:XE=39009:GOSUB 20                          PM
·2531 POKE 37892,FN LO(XB):POKE 37893,FN
 PG(XB)                                                   IF
·2532 REM SET UP FOR SPRITE 0, GO MOVE, C
```


www.commodore.ca

```
HECK COLLISIONS, REDO IF NECESSARY                    DP
·2533 REM LDY#0 JSR 38400 JSR 39072                  KH
·2534 DATA 160,0,32,0,150,32,160,152                 BJ
·2535 REM DO IT AGAIN? IF NOT, RESET TIME
R                                                    NP
·2536 REM DEC 37928 BNE-13 LDA 37926 STA
37928                                                LI
·2537 DATA 206,40,148,208,243,173,38,148,
141,40,148                                           LH
·2538 REM RESET MOVEMENT VECTOR, QUIT                FF
·2539 REM LDA 37896 STA 37898 LDA 37897 S
TA 37899 JMP(37900)                                  NJ
·2540 DATA 173,8,148,141,10,148,173,9,148
,141,11,148,108,12,148                               FG
·2549 REM *** SET SHAPE 0 ***                        DB
·2550 XB=39040:XE=39057:GOSUB 20                     BC
·2551 REM USING JOYDIR AS INDEX, GET SPRI
TE SHAPE CODE FROM SPRITE 0 DIREC.TABLE              CA
·2552 REM LDA 37923 BNE+9 LDY 37963 LDA 3
7951,Y STA 33784 JSR 39232 RTS                       MA
·2553 DATA 173,35,148,208,9,172,75,148,18
5,63,148,141,248,131,32,64,153,96                    FI
·2569 REM ** SPRITE 0 COLLISION ROUTINE              AM
·2570 XB=39072:XE=39111:GOSUB 20                     DC
·2571 REM DO WE BOUNCE OFF SPRITES? IF SO
, DO IT                                              DG
·2572 REM LDA 37940 BEQ+16 LDA 53278 STA
37929 AND#1 BEQ+6                                    CH
·2573 DATA 173,52,148,240,16,173,30,208,1
41,41,148,41,1,240,6                                 LG
·2574 REM JSR 39248 JMP UNMOVE                       GD
·2575 DATA 32,80,153,76,208,152                      FI
·2576 REM DO WE BOUNCE OFF FOREGROUND? IF
 SO, DO IT AND RETURN                                FK
·2577 REM LDA 37941 BEQ+13 LDA 53279 AND#
1 BEQ+6 JSR 39280 JMP UNMOVE RTS                     CP
·2578 DATA 173,53,148,240,13,173,31,208,4
1,1,240,3,32,112,153,76,208,152,96                   CD
·2579 REM ** UNMOVE **                               AI
·2580 XB=39120:XE=39148:GOSUB 20                     PK
·2581 REM STOP GOSPEED, REVERSE JOYDIR AN
D STORE IT                                           AN
·2582 REM LDA#1 STA 37928 LDY 37963 LDA 3
8009,Y STA 37963                                     KH
·2583 DATA 169,1,141,40,148,172,75,148,18
5,121,148,141,75,148                                 OB
·2584 REM GO MOVE, REVERSE JOYDIR AND RES
TORE IT, QUIT                                        KE
·2585 REM LDY#0 JSR 38400 LDY 37963 LDA 3
8009,Y STA 37963 RTS                                 IO
·2586 DATA 160,0,32,0,150,172,75,148,185,
121,148,141,75,148,96                                OC
·2599 REM   *** READ KEYBOARD ***                    MN
·2600 REM XB=00000:XE=00000:GOSUB 20                 KC
·2601 REM SET READ VECTOR TO POINT TO KEY
BOARD ROUTINE (REMOVE REM TO USE)                    LE
·2602 REM X=38912:POKE 37890,FN LO(X):POK
E 37891,FN PG(X):REM (NOT IN FORCE)                  PF
·2699 REM ** BASIC VARIABLE SUBROUTINES              DD
·2700 XB=39168:XE=39174:GOSUB 20                     KD
·2701 REM ** REPORT FIREBUTTON - C0%                 LD
·2702 REM LDY#10 LDA#1 STA(45),Y RTS                 EA
·2703 DATA 160,10,169,1,145,45,96                    PP
·2709 REM ** REPORT SPRITES TO BASIC                 PN
·2710 XB=39184:XE=39222:GOSUB 20                     CJ
·2711 REM C1%=EDGEWRAP, C2%=S/S COLLIS.              JP
·2712 REM LDY#17 LDA 37943 BEQ+2 STA(45)Y
 LDY#24 LDA 37937 BEQ+2 STA(45)Y                     BB
·2713 DATA 160,17,173,55,148,240,2,145,45
,160,24,173,49,148,240,2,145,45                      LI
·2714 REM C3%=S/FQREG.COLLIS.,ERASE,RET.             KM
·2715 REM  LDY#31 LDA 37938 BEQ+2 STA(45)
Y                                                    AD
·2716 DATA  160,31,173,50,148,240,2,145,4
5                                                    CL
·2717 REM LDA#0 STA 37937 STA 37938 STA 3
7943 RTS                                             EP
·2718 DATA 169,0,141,49,148,141,50,148,14
1,55,148,96                                          KJ
·2719 REM ** REPORT MOVEMENT                         BK
·2720 XB=39232:XE=39241:GOSUB 20                     BC
·2721 REM C4%=PLAYER HAS TRIED TO MOVE               GH
·2722 REM LDY#38 LDA#1 STA(45)Y STA 37935
 RTS                                                 PN
·2723 DATA 160,38,169,1,145,45,141,47,148
,96                                                  NK
·2729 REM ** REPORT SPRITE 0 BOUNCE/S                MP
·2730 XB=39248:XE=39257:GOSUB 20                     AC
·2731 REM C5%=SPRITE 0 BOUNCED/SPRITE:               CD
·2732 REM LDY#45 LDA 37929 AND#254 STA(45
)Y RTS                                               AI
·2733 DATA 160,45,173,41,148,41,254,145,4
5,96                                                 DH
·2739 REM ** REPORT SPRITE 0 BOUNCE/F                MA
·2740 XB=39280:XE=39289:GOSUB 20                     CF
·2741 REM C6%=SPRITE 0 BOUNCED FOREGROU.             PH
·2742 REM LDY#52 LDA#1 STA(45)Y STA 37930
 RTS                                                 KM
·2743 DATA 160,52,169,1,145,45,141,42,148
,96                                                  LP
·2898 REM   *** WRAPUP ***                           HC
·2899 REM ALL ROUTINES EXIT THROUGH HERE             AJ
·2900 XB=40704:XE=40721:GOSUB 20                     MD
·2901 REM ** SET COLLISION VECTOR                    FI
·2902 X=40704:POKE 37900,FN LO(X):POKE 37
901,FN PG(X)                                         HB
·2906 REM GET COLLISION DATA AND REPORT I
T TO BASIC VARIABLES                                 DG
·2907 REM LDA 53278 STA 37937 LDA 53279 S
TA 37938 JSR 39184                                   IJ
·2908 DATA 173,30,208,141,49,148,173,31,2
08,141,50,148,32,16,153                              ED
·2915 REM RETURN TO HOUSEKEEPING                     EL
·2916 REM JMP(37888)                                 MI
·2917 DATA 108,0,148                                 FH
·2989 REM ROUTINES ADDED BEFORE FINAL JMP
```


www.commodore.ca

```
DD  () WILL EXECUTE EVERY INTERR.
KD ·2998 REM  *** SET UP SPRITE SHAPES ***
LD ·2999 REM ** PLANETS
EA ·3000 FOR I=34304 TO 34816 STEP 512:FOR J
PP =0 TO 448 STEP 64:FOR K=0 TO 20
PN ·3001 READ A:POKE I+J+K,A:NEXT
CJ ·3002 FOR K=21 TO 63:POKE I+J+K,0:NEXT:NE
JP XT:GOSUB 3985:NEXT
Y  ·3009 REM ** PLANET 1 SHAPE DATA
BB ·3010 DATA 0,40,0,0,175,0,82,191,192,82,1
5  75,128,2,191,128,0,171,0,0,40,0
LI ·3011 DATA 0,40,0,0,190,0,2,255,128,5,190
KM ,128,5,254,128,0,175,0,0,40,0
)  ·3012 DATA 0,40,0,0,250,0,3,254,128,2,90,
AD 128,3,90,128,0,190,0,0,40,0
4  ·3013 DATA 0,40,0,0,234,0,3,250,128,3,229
CL ,128,3,229,128,0,250,0,0,40,0
3  ·3014 DATA 0,40,0,0,170,0,3,234,80,3,170,
EP 80,3,170,128,0,234,0,0,40,0
4  ·3015 DATA 0,40,0,0,170,20,3,170,212,2,17
KJ 0,128,2,170,192,0,170,0,0,40,0
BK ·3016 DATA 0,40,0,0,170,64,2,171,192,2,17
BC 0,192,2,171,192,0,170,0,0,40,0
GH ·3017 DATA 0,40,0,5,171,0,6,175,192,2,171
5  ,192,2,175,192,0,170,0,0,40,0
PN ·3019 REM ** PLANET 2 SHAPE DATA
3  ·3020 DATA 0,40,0,3,170,192,62,170,148,16
NK 2,170,143,42,255,252,2,255,192,0,40,0
MP ·3021 DATA 0,40,0,3,106,192,61,106,188,24
AC 1,106,143,62,171,252,2,171,192,0,40,0
CD ·3022 DATA 0,24,0,3,90,192,61,90,188,81,9
   0,143,63,234,188,3,234,128,0,24,0
AI ·3023 DATA 0,20,0,3,86,192,61,86,188,241,
   86,138,61,255,168,1,255,128,0,20,0
DH ·3024 DATA 0,20,0,3,85,128,61,85,104,241,
MA 85,74,63,223,252,3,223,192,0,20,0
CF ·3025 DATA 0,20,0,3,149,192,62,149,124,24
PH 2,149,79,63,253,252,3,253,192,0,20,0
   ·3026 DATA 0,40,0,2,169,192,62,169,124,24
KM 2,169,79,63,255,252,3,255,192,0,40,0
   ·3027 DATA 0,40,0,2,170,0,42,170,188,162,
LP 170,143,63,255,212,3,255,192,0,40,0
HC ·3058 GOSUB 3985
AJ ·3059 REM ** SPRITE ANIMATION TABLES
MD ·3060 FOR I=38088 TO 38112 STEP 8:FOR J=0
FI   TO 7:READ A:POKE I+J,A:NEXT:NEXT
   ·3061 REM ** ANIMATION TABLE DATA
HB ·3062 DATA 24,25,26,27,28,29,30,31
   ·3063 DATA 32,33,34,35,36,37,38,39
DG ·3064 DATA 27,26,25,24,31,30,29,28
   ·3065 DATA 37,36,35,34,33,32,39,38
IJ ·3098 REM    ***   SPRITE 0 SHAPES   ***
   ·3099 REM ** SPRITE 0 DIRECTION TABLE
ED ·3100 FOR I=37952 TO 37961:READ A:POKE I,
EL  A:NEXT:GOSUB 3985
MI ·3101 DATA 16,20,0,22,23,21,0,18,17,19
FH ·3102 REM ** SPRITE 0 ANIMATION TABLE
   ·3103 FOR I=0 TO 7:POKE 38080+I,16+I:NEXT
   ·3104 REM ** SPRITE 0 SHAPES
   ·3105 FOR I=33792 TO 34240 STEP 64:FOR J=
     0 TO 18 STEP 3
   ·3106 READ A:POKE I+J,A:POKE I+J+1,0:POKE
     I+J+2,0:NEXT
   ·3107 FOR J=21 TO 63:POKE I+J,0:NEXT:NEXT
    :GOSUB 3985
   ·3109 REM ** SPRITE 0 DATA
   ·3110 DATA 8,28,28,28,54,34,0
   ·3111 DATA 6,14,28,120,240,48,32
   ·3112 DATA 0,224,62,31,62,224,0
   ·3113 DATA 32,48,240,120,28,14,6
   ·3114 DATA 0,34,54,28,28,28,8
   ·3115 DATA 4,12,15,30,56,112,96
   ·3116 DATA 0,7,124,248,124,7,0
   ·3117 DATA 96,112,56,30,15,12,4
   ·3298 REM ** SPRITE MOVEMENT DATA
   ·3299 REM ** LOCATION REGISTER OFFSET TAB
    LE
   ·3300 X=0:FOR I=37944 TO 37951:POKE I,X:X
    =X+2:NEXT
   ·3304 REM ** SPRITE BITMASK TABLE
   ·3305 X=1:FOR I=37962 TO 37976 STEP 2:POK
    E I,X:X=X*2:NEXT
   ·3307 REM ** SPRITE JOYDIR TABLE
   ·3308 REM SPRITE DIRECTIONS: 1=UP 2=DN 4=
    LF 8=RT  ADD FOR DIAGONALS
   ·3309 REM 0=NO MOVEMENT (CAN BE CHANGED F
    ROM BASIC)
   ·3310 FOR I=37963 TO 37977 STEP 2:READ A:
    POKE I,A:NEXT
   ·3311 DATA 0,5,6,10,9,0,0,0
   ·3312 REM ** SPRITE BORDERS TABLE **
   ·3313 REM EACH SPRITE'S SCREEN MARGINS AR
    E INDIVIDUALLY SET
   ·3314 REM SPRITES ARE CONSIDERED IN NUMER
    ICAL ORDER
   ·3315 REM ** TOP EDGES (0-7)
   ·3316 FOR I=37978 TO 37992 STEP 2:READ A:
    POKE I,A:NEXT
   ·3317 DATA 50,43,43,43,43,43,43,43
   ·3318 REM ** BOTTOM EDGES (0-7)
   ·3319 FOR I=37979 TO 37993 STEP 2:READ A:
    POKE I,A:NEXT
   ·3320 DATA 243,250,250,250,250,250,250,25
    0
   ·3321 REM ** LEFT EDGES (0-7)
   ·3322 FOR I=37994 TO 38008 STEP 2:READ A:
    POKE I,A:NEXT
   ·3323 DATA 23,0,0,0,0,0,0,0
   ·3324 REM ** RIGHT EDGES (0-7)
   ·3325 FOR I=37995 TO 38009 STEP 2:READ A:
    POKE I,A:NEXT
   ·3326 DATA 80,87,87,87,87,87,87,87
   ·3327 REM ** JOYSTICK REVERSAL TABLE
   ·3328 FOR I=38010 TO 38019:READ A:POKE I,
    A:NEXT
   ·3329 DATA 2,1,0,8,10,9,0,4,6,5
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 121 and 122 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·3985 PRINT "[RVSON]![RVSOFF]";                    NG
·3990 RETURN                                       IM
```

SLITHER
FROM PAGE 35

```
·1 REM ALL TIME HIGH 43,840                        JO
·10 POKE53281,0:POKE53280,0:POKE52,56:POK
 E56,56:I=RND(-TI):PRINTCHR$(142)                  HP
·20 J0=127:J2=56320:S=1024:SI=54272:C=SI:
 S1=SI+1:S4=SI+4:S5=SI+5:S6=SI+6:SV=SI+24          PK
·22 BH=33:BS=34:BC=32:NS=40:C$=CHR$(96)            HE
·30 FORI=SITOSV:POKEI,0:NEXT:POKES6,242:P
 OKESV,15                                          IP
·32 POKE56334,0:POKE1,51                           KE
·34 A=14336:B=53248:FORI=0TO1487:POKEI+A,
 PEEK(I+B):NEXT                                    HC
·36 POKE1,55:POKE56334,1:POKE53272,30              PP
·38 FORI=0TO30:READP:POKE49152+I,P:NEXT            AG
·40 FORI=14600TO14631:READP:POKEI,P:NEXT           IE
·42 FORI=15064TO15159:READP:POKEI,P:NEXT           KH
·44 FORI=14720TO14799:READP:POKEI,P:NEXT           OA
·60 DIMJD(32),L$(NS),FC(12),FS(13),BC(5)           IE
·64 JD(1)=-40:JD(2)=40:JD(4)=-1:JD(8)=1            KO
·66 FORJ=1TONS:READL$(J):NEXT                      EC
·68 FORI=1TO12:READFC(I):NEXT                      JN
·70 FORI=1TO13:READFS(I):NEXT                      IO
·72 FORI=1TO5:READCB(I):NEXT                       DJ
·99 :                                              DI
·100 PRINT"[CLEAR][6"[DOWN]"]"                     NG
·120 PRINT"[PURPLE]  [5"[s C]"] [s C][5"
 "][s C] [5"[s C]"] [s C][3" "][s C] [5"[
 s C]"] [4"[s C]"] "                               EH
·122 PRINT"[BLUE]  [s C][5" "][s C][5" "]
 [s C][3" "][s C][3" "][s C][3" "][s C] [
 s C][5" "][s C][3" "][s C]"                       DN
·124 PRINT"[c 7][3" "][3"[s C]"]  [s C][5
 " "][s C][3" "][s C][3" "][5"[s C]"] [4"
 [s C]"]  [4"[s C]"]"                              KN
·126 PRINT"[CYAN][6" "][s C] [s C][5" "][
 s C][3" "][s C][3" "][s C][3" "][s C] [s
  C][5" "][s C]  [s C]"                            HE
·127 PRINT"[GREEN]  [4"[s C]"]  [5"[s C]"
 ] [s C][3" "][s C][3" "][s C][3" "][s C]
  [5"[s C]"] [s C][3" "][s C]"                     CK
·130 FORI=1TO200:NEXT                              KN
·132 PRINT"[HOME][4"[DOWN]"][WHITE]"TAB(1
 2)"DIFFICULTY 12345":P=S+183                      OB
·134 J=PEEK(J2):IFJ=119THENP=P+1:IFP>S+18
 7THENP=P-1                                        PG
·136 IFJ=123THENP=P-1:IFP<S+183THENP=P+1           DM
·138 IFJ=111THEN160                                FE
·140 POKEP,PEEK(P)OR128:FORI=1TO30:NEXT:P
 OKEP,PEEK(P)AND127:GOTO134                        HJ
·160 DI=P-(S+182):DU=DI:W=7:SC=0:L=1               IJ
·162 POKE53270,216:POKE53282,6:POKE53283,
 5                                                 GO
·190 GOSUB700                                      CH
·192 P=S+165:PD=1                                  GN
·199 :                                             DI
·200 J=PEEK(J2):IFJ=J0THEN250                      BC
·202 J=J0-J:IFJD(J)THENPD=JD(J)                    DH
·250 Q=PEEK(P+PD):IFQ<>BCTHEN300                   EK
·252 POKEP,BS:P=P+PD:POKEP+C,CB:POKEP,BH           CK
·260 FORD=1TO50:NEXT:GOTO200                       MO
·298 :                                             DI
·299 : * COLLISION *                               FP
·300 IFQ<91THEN400                                 DI
·302 Q=Q-90:SC=SC+FS(Q):POKEP,BS:FORI=Q*1
 2+10TO0STEP-Q/2:POKES1,I:POKES4,17:NEXT           IK
·304 P=P+PD:POKEP,BH:FORI=0TOQ*12+10STEPQ
 /2:POKES1,I:POKES4,17:NEXT:POKES4,16              BB
·310 POKEP+C,CB:GOSUB350:DU=DU-1:IFDU=0TH
 EN800                                             LN
·312 GOTO200                                       BO
·348 :                                             DI
·349 : * STATUS LINE *                             JC
·350 PRINT"[HOME][WHITE]SCORE  ";:N=SC:GO
 TO940                                             BB
·360 PRINT"[HOME]"TAB(19)"[BLUE]("CHR$(48
 +W)")"TAB(28)"[GREEN]HIGH  ";:N=SH:GOSUB
 940                                               MC
·362 PRINT"[WHITE]":RETURN                         KJ
·398 :                                             DI
·399 : * CRASHED *                                 JA
·400 POKES1,5:POKES4,129:FORI=15TO0STEP-1
 :POKESV,I:POKEP+C,I:POKE53281,I                   OO
·402 POKE53280,I:POKE53282,15-I:POKE53283
 ,I:FORJ=1TO20:NEXT:NEXT                           OG
·406 POKE53282,6:POKE53283,5:POKESV,15:PO
 KES4,0:W=W-1:GOSUB360:IFW=0THEN410                CK
·408 POKEP,BC:SYS49152:FORI=1TO800:NEXT:G
 OTO192                                            LG
·410 PRINT"[HOME][11"[DOWN]"]":PRINTTAB(1
 5)"GAME  OVER"                                    BO
·412 IFSC>SHTHENSH=SC:GOSUB450                     FB
·420 PRINTTAB(7)"[WHITE]PRESS [RED]BUTTON
 [WHITE] TO PLAY AGAIN":GOSUB360                   MK
·430 IFPEEK(J2)<>111THEN430                        IA
·440 GOSUB3100:GOTO130                             LK
·450 FORJ=1TO15:PRINTTAB(15)"[UP][RVSON][
 RED]HIGH SCORE":FORI=1TO20:NEXT:POKES1,3
 8:POKES4,33                                       FJ
·452 PRINTTAB(15)"[UP][YELLOW]HIGH SCORE"
 :FORI=1TO20:NEXT:POKES1,40:POKES4,33:NEX
 T                                                 ON
·454 POKES4,32:RETURN                              HI
·598 :                                             DI
·599 : * PLACE PRIZES *                            NC
·600 R=INT(RND(1)*19)*40+INT(RND(1)*36)+1
 22+S:IFPEEK(R)<>BCORR=S+165THEN600                LE
```


www.commodore.ca

```
·610 L1=L:IFL>12THENL1=12                                    PI
·620 POKER,L1+90:POKER+C,FC(L1):RETURN                       DO
·698 :                                                       DI
·699 : * SET UP SCREEN *                                     PF
·700 IFL=1THENGOSUB900                                       KD
·701 PRINT"[CLEAR][DOWN][DOWN][c 8][40"$"                    GD
 ]";
·702 FORI=1TO21:PRINT"$[38" "]$";:NEXT                       GE
·704 PRINT"[39"$"]";:POKE2023,36:POKE5629                    PN
 5,15
·705 L1=L:IFL>12THENL1=12                                    PI
·706 PRINT"[HOME][DOWN][WHITE]"TAB(17-LEN                    IF
 (L$(L))/2)"#"MID$(STR$(L),2)"-"L$(L)" ";
·707 POKE646,FC(L1):PRINTCHR$(L1+218)                        DN
·710 READA,B:IFA=0THEN720                                    FK
·711 N=1:IFB-A>39THENN=40                                    GF
·712 FORI=A+STOB+SSTEPN:POKEI,36:POKEI+C,                    OH
 15:NEXT:GOTO710
·720 FORJ=1TODU:GOSUB600:NEXT:GOSUB350:GO                    PA
 SUB360:CB=CB(DI):FORI=1TO400:NEXT
·732 RETURN                                                  IM
·798 :                                                       DI
·799 : * COMPLETED A SCREEN *                                KN
·800 DU=DI:FORI=1TO200:NEXT:L1=L:IFL>12TH                    CA
 ENL1=12
·810 PRINT"[HOME][10"[DOWN]"]"                               JL
·812 PRINTTAB(15)"BONUS  [RED]"RIGHT$("                      EM
 "+STR$(FS(L1+1)),3)
·814 SC=SC+FS(L1+1):GOSUB350:FORI=1TO800:                    PC
 NEXT:L=L+1:IFL<=NSTHEN190
·820 K=0:L=1:GOSUB3100:PRINT"[CLEAR]":POK                    LH
 ES6,240
·822 A=80:B=960:N=40:GOSUB960:A=961:B=999                    LO
 :N=1:GOSUB960:A=959:B=119:N=-40:GOSUB960
·824 A=118:B=81:N=-1:GOSUB960                                KH
·830 GOSUB350:GOSUB360:M=0:A$="LEVEL"+C$+                    JP
 CHR$(DI+112)+C$+C$+"BONUS"
·832 POKES6,251:FORJ=1TO14:POKES1,J+70:PO                    JI
 KES4,21:POKES4,20
·834 FORI=1196TO1476STEP40:POKEI+J-40,32:                    OO
 POKEI+J,ASC(MID$(A$,J,1))-64:NEXT:NEXT
·836 FORI=1TO800:NEXT:POKES6,242:FORI=1TO                    AH
 12:POKEI+1597,I+90:POKEI+55869,FC(I)
·838 POKES1,I+5:POKES4,129:FORJ=1TO10:NEX                    LL
 T:M=M+FS(I)*DI:POKES4,128
·840 FORJ=1TO20:NEXT:PRINT"[HOME][RED][12                    HG
 "[DOWN]"]"TAB(18)RIGHT$("[3" "]"+STR$(M)
 ,4)
·842 SC=SC+FS(I)*DI:GOSUB350:NEXT:DI=DI+1                    PA
 :DU=DU+1:IFDI>5THENDI=5:DU=5
·854 FORI=1TO200:NEXT:PRINT"[5"[DOWN]"]"T                    JL
 AB(12)DI"[BLUE] BONUS LIVES":W=W+DI:IFW>
 9THENW=9
·860 GOSUB360:PRINT"[DOWN]"TAB(7)"[WHITE]                    PA
 GET READY FOR DIFFICULTY[RED]"DI
·870 FORI=1TO2000:NEXT:GOTO190                               EP
·899 :                                                       DI
·900 PRINT"[CLEAR][8"[DOWN]"][RED]"TAB(9)                    FA
 NS"[WHITE]SCREENS THIS LEVEL":POKES6,252                    IK
 :POKES1,75
·902 POKES1+14,DI*15+30:POKES4,21:POKES4,                    DO
 20:FORI=1TO2300:NEXT:POKES6,242:RETURN
·940 PRINTRIGHT$("[5" "]"+STR$(N),6);:RET                    NK
 URN
·960 FORI=A+STOB+SSTEPN:POKEI,42:POKEI+C,                    ND
 2+K:K=NOTK:POKES1,5+K:POKES4,129
·962 FORJ=1TO4:NEXT:POKES4,128:FORJ=1TO2:                    DI
 NEXT:NEXT:RETURN
·998 :                                                       CC
·999 : * ML DATA *                                           CD
·1000 DATA169,0,133,253,169,4,133,254,160                    ML
 ,0,162,4
·1002 DATA177,253,201,34,208,4,169,32,145                    BI
 ,253,200,208,243,230,254,202,208,238,96
·1099 : * CHARACTER DATA *                                   BH
·1100 DATA40,158,158,190,190,158,158,40                      MH
·1102 DATA40,190,190,190,190,190,190,40                      DO
·1104 DATA36,254,36,36,36,254,36,0                           CO
·1106 DATA85,85,85,85,85,85,85,85                            DI
·1119 :                                                      DF
·1120 DATA130,40,60,255,255,60,60,0                          LK
·1122 DATA130,40,255,255,255,60,60,0                         GD
·1124 DATA0,0,56,124,254,124,56,0                            FJ
·1126 DATA60,40,170,190,170,190,170,40                       EF
·1128 DATA160,160,60,247,255,255,60,0                        HK
·1130 DATA3,3,7,14,14,28,56,96                               BJ
·1132 DATA60,60,60,60,60,60,247,247                          BJ
·1134 DATA60,60,60,60,60,60,247,247                          CD
·1136 DATA28,193,3,1,3,1,3,1                                 FC
·1138 DATA20,20,60,195,195,195,60,0                          FC
·1140 DATA20,20,60,195,195,195,60,0                          LE
·1142 DATA0,56,131,194,131,194,44,0                          DI
·1159 :                                                      GD
·1160 DATA254,198,198,198,198,198,254,0                      FA
·1162 DATA6,6,6,6,6,6,6,0                                    JG
·1164 DATA254,6,6,254,192,192,254,0                          IG
·1166 DATA254,6,6,254,6,6,254,0                              LN
·1168 DATA198,198,198,254,6,6,6,0                            AC
·1170 DATA254,192,192,254,6,6,254,0                          EB
·1172 DATA192,192,192,254,198,198,254,0                      NA
·1174 DATA254,6,6,6,6,6,6,0                                  PK
·1176 DATA254,198,198,254,198,198,254,0                      CC
·1178 DATA254,198,198,254,6,6,6,0                            AP
·1199 : * NAMES *                                            JN
·1201 DATA"THE COURTYARD"                                    HP
·1202 DATA"THE GATEHOUSE"                                    DN
·1203 DATA"THE BARRACKS"                                     HE
·1204 DATA"THE GARDEN"                                       HI
·1205 DATA"THE CHAPEL"                                       KC
·1206 DATA"THE PIT"                                          DO
·1207 DATA"GALLERY I"                                        OI
·1208 DATA"THE LIBRARY"                                      CI
·1209 DATA"THE SNAKE"                                        BH
·1210 DATA"THE COLONADE"                                     KM
·1211 DATA"THE WINE CELLAR"                                  DI
·1212 DATA"THE THREE ROOMS"
```


www.commodore.ca

```
·1213 DATA"THE DOUBLE-E ROOM"                AH
·1214 DATA"GALLERY II"                       IO
·1215 DATA"THE OVAL ROOM"                    EC
·1216 DATA"THE GREENHOUSE"                   LM
·1217 DATA"THE MESH ROOM"                    FP
·1218 DATA"THE PRISON"                       MC
·1219 DATA"THE I-J-K ROOM"                   ID
·1220 DATA"THE CUBICLES"                     CK
·1221 DATA"TRIPLE-T"                         AD
·1222 DATA"THE VESTIBULE"                    JB
·1223 DATA"THE ARENA"                        DN
·1224 DATA"DEATHTRAP I"                      OO
·1225 DATA"THE S-CHAMBER"                    DE
·1226 DATA"THE SPIRAL HALL"                  OO
·1227 DATA"THE WEB"                          KJ
·1228 DATA"THE MAZE"                         PB
·1229 DATA"THE PIT II"                       GL
·1230 DATA"THE ATRIUM"                       LB
·1231 DATA"THE CLOAKROOMS"                   MF
·1232 DATA"THE DUNGEON"                      OB
·1233 DATA"THE SHIFTING HALL"                GO
·1234 DATA"THE SUITE"                        GG
·1235 DATA"THE PANTRY"                       LP
·1236 DATA"GALLERY [3"I"]"                   NN
·1237 DATA"THE WINDBREAK"                    FL
·1238 DATA"DEATHTRAP II"                     CE
·1239 DATA"THE WALL"                         PC
·1240 DATA"DEATHTRAP [3"I"]"                 JB
·2999 : * MISCELLANEOUS *                    IH
·3010 DATA10,12,7,15,10,7,9,15,9,9,15,10     FJ
·3020 DATA10,25,50,75,100,125,150,175,200
 ,250,300,400,500                            AI
·3030 DATA15,10,12,14,13                     AL
·3099 : * COORDINATE DATA *                  IG
·3100 POKE65,PEEK(61):POKE66,PEEK(62):RET
 URN                                         GI
·3110 DATA0,0                                BC
·3120 DATA135,455,495,498,145,465,502,505
 ,0,0                                        AG
·3130 DATA139,379,140,380,410,429,650,669
 ,699,939,700,940,0,0                        PP
·3140 DATA401,420,659,678,0,0                EO
·3150 DATA330,730,350,750,521,536,543,558
 ,0,0                                        FO
·3160 DATA250,269,250,810,269,829,811,816
 ,822,828,455,458,461,464,455,655            JD
·3162 DATA504,664,656,663,0,0                PF
·3170 DATA130,410,650,930,149,429,669,949
 ,339,739,340,740,0,0                        KC
·3180 DATA287,767,312,792,290,309,410,429
 ,530,549,650,669,770,789,0,0                LM
·3190 DATA285,314,325,405,445,474,474,594
 ,605,634,645,725,765,794,0,0                LE
·3200 DATA446,606,250,450,610,930,454,614
 ,458,618,461,621,465,625,149,469            JI
·3202 DATA629,949,473,633,0,0                CP
·3210 DATA210,229,361,374,385,398,530,549
 ,681,694,705,718,850,869,0,0                CE
·3220 DATA415,424,521,530,549,558,570,650
 ,850,930,455,655,855,935,464,664            HP
·3222 DATA864,944,589,669,869,949,0,0        FF
·3230 DATA244,257,262,275,524,532,547,555
 ,804,817,822,835,297,777,302,782,0,0        KH
·3240 DATA241,250,254,265,269,278,418,421
 ,765,770,789,794,290,730,294,774            FI
·3242 DATA458,778,461,781,305,785,309,789
 ,0,0                                        IA
·3250 DATA378,381,415,417,422,424,453,454
 ,465,466,492,493,506,507,531,532            JF
·3252 DATA547,548,691,692,707,708,732,733    FK
·3254 DATA746,747,773,774,785,786,815,817
 ,822,824,858,861,0,0                        CF
·3260 DATA205,234,405,434,605,634,805,834
 ,245,785,0,0                                BN
·3270 DATA325,925,250,730,335,935,140,740
 ,345,945,150,750,355,955,0,0                EB
·3280 DATA130,410,650,930,149,429,669,949
 ,334,345,374,385,414,428                    FF
·3282 DATA651,665,694,705,734,745,0,0        AE
·3290 DATA286,304,330,730,766,774,339,739
 ,777,779,344,784,545,546                    LF
·3292 DATA312,312,351,351,390,390,429,429
 ,468,468,507,507,587,587,628,628,669,669    EA
·3294 DATA710,710,751,751,792,792,0,0        NI
·3300 DATA135,255,140,260,145,265,150,270
 ,155,275,805,925,810,930,815,935            OI
·3302 DATA820,940,825,945,830,950,835,955
 ,0,0                                        KO
·3310 DATA286,486,580,940,314,514,521,530
 ,535,545,550,558,0,0                        JB
·3320 DATA326,338,341,353,526,553,726,738
 ,741,753,141,301,366,406,646,686            GJ
·3322 DATA393,433,673,713,778,938,0,0        NE
·3330 DATA137,142,253,257,262,266,410,413
 ,426,429,610,613,626,629,773,777,782        PB
·3332 DATA786,897,902,177,217,182,222,450
 ,570,469,589,817,857,822,862,0,0            NM
·3340 DATA243,843,290,810,297,937,142,822
 ,309,829,316,876,248,257,267,276            KC
·3342 DATA843,852,862,871,0,0                KI
·3350 DATA241,263,416,423,576,583,736,743
 ,896,918,936,958,456,536,623,703,0,0        PN
·3360 DATA216,224,376,381,540,544,696,701
 ,860,864,145,865,416,936,0,0                PA
·3370 DATA325,725,136,456,616,936,347,747
 ,158,478,638,958,521,530,542,552            KD
·3372 DATA331,341,353,357,731,741,753,757
 ,0,0                                        HI
·3380 DATA404,412,427,435,452,932,140,820
 ,467,947,0,0                                AP
·3390 DATA410,416,423,429,655,668,821,824
 ,450,930,695,735,895,935,861,941            IP
·3392 DATA469,949,0,0                        JD
·3400 DATA321,329,335,337,342,344,350,358
 ,526,536,543,553,766,776,783,793            OO
·3402 DATA376,496,383,503,566,726,593,753
```


www.commodore.ca

```
,816,936,823,943,0,0                                        OB
·3410 DATA241,246,250,256,263,275,454,456
,463,465,614,616,623,625,804,829                            LJ
·3412 DATA833,838,290,770,296,416,656,776
,303,423,663,783,309,789,0,0                                GP
·3420 DATA244,275,284,315,655,664,695,704
,324,924,325,925,339,739,340,740                            FO
·3422 DATA899,939,900,940,354,714,355,715
,0,0                                                        FB
·3430 DATA361,376,384,398,681,696,704,718
,297,457,617,777,140,300,460,620                            BJ
·3432 DATA780,940,303,463,623,783,0,0                       NK
·3440 DATA408,413,648,653,425,431,665,671
,295,297,302,304,775,777,782,784                            JK
·3442 DATA134,414,654,934,145,425,665,945
,0,0                                                        CC
·3450 DATA325,925,330,930,335,935,145,745
,150,750,155,755,336,338,742,744,0,0                        AE
·3460 DATA241,255,264,278,801,815,824,838
,288,528,535,775,259,819,260,820                            AL
·3462 DATA311,551,544,784,0,0                               IG
·3470 DATA565,925,490,850,415,775,340,700
,265,625,190,550,155,475,0,0                                OA
·3480 DATA281,292,295,307,533,536,543,546
,772,784,787,798,332,652,812,932                            MP
·3482 DATA339,419,340,420,659,739,660,740
,147,267,427,747,0,0                                        HM
·3490 DATA523,556,0,0                                       KC
·3500 DATA260,278,444,454,459,463,616,620
,625,635,801,819,135,615,464,944,0,0                        FK
```

COMPULOAN
FROM PAGE 97

```
·10 PRINT "COPYRIGHT 1985 CHERYL PETERSON
"                                                           EB
·15 PRINT"LOAN PAYMENT CALCULATOR"                          GF
·20 PRINT"[RVSON]S[RVSOFF]CREEN DISPLAY O
R [RVSON]P[RVSOFF]RINTOUT?"                                 KB
·30 GET S$:IF S$=""GOTO30                                   CG
·35 IF S$="S" OR S$="P" THEN 40                             MJ
·36 PRINT"WAS THAT AN [RVSON]S[RVSOFF] OR
 A [RVSON]P[RVSOFF]?":GOTO20                                NO
·40 PRINT"VIEW/PRINT PAYBACK SCHEDULE ([R
VSON]Y[RVSOFF] OR [RVSON]N[RVSOFF])?"                       MK
·50 GET V$:IF V$=""THEN50                                   IM
·55 IF V$="Y" OR V$="N" THEN 210                            CE
·60 PRINT"WAS THAT A [RVSON]Y[RVSOFF] OR
AN [RVSON]N[RVSOFF]?":GOTO50                                MF
·210 INPUT"ENTER CASH VALUE BEING FINANCE
D";A                                                        EP
·215 K=A:GOSUB 1000                                         NI
·216 A=K                                                    EH
·220 INPUT"ENTER NUMBER OF PAYMENTS PER Y
EAR";N                                                      AK
·225 K=N:GOSUB 1000                                         NN
·226 N=K                                                    EI
·230 INPUT"ENTER TOTAL NUMBER OF PAYMENTS
 TO MAKE";T                                                 KP
·235 K=T:GOSUB 1000                                         NL
·236 T=K                                                    FG
·240 INPUT"ENTER YEARLY INTEREST RATE";I                    LN
·241 K=I:GOSUB 1000                                         NA
·242 I=K                                                    EP
·243 II=I:I=I/100                                           GG
·250 IF S$="S"THEN 260                                      IF
·251 OPEN1,4,7:CMD1                                         KP
·260 PRINT CHR$(147)                                        FG
·270 PRINT"LOAN PAYMENT CALCULATOR"                         GF
·275 GOSUB 4030                                             FO
·280 PRINT"AMOUNT FINANCED $";A$                            AO
·285 PRINT"INTEREST RATE";II;"%."                           AC
·290 PRINT"# OF PAYMENTS PER YEAR: ";N                      NA
·300 PRINT"TOTAL NUMBER OF PAYMENTS IS ";
T                                                           DE
·400 P=A*(I/N)/(1-(1/(1+(I/N))[UPARROW](T
)))                                                         HO
·410 Z=P*T:C=Z-A                                            NO
·600 GOSUB 3000                                             FE
·610 PRINT"EACH PAYMENT IS";P$                              PF
·620 PRINT"TOTAL INTEREST PAID IS $";C$                     PP
·630 PRINT"TOTAL AMOUNT PAID IS $";Z$                       IB
·640 PRINT                                                  JJ
·690 IFV$<>"Y"THEN845                                       DJ
·691 C=A*I/N                                                DN
·695 J=P-C                                                  DM
·697 A=A+C                                                  CL
·700 IF S$="S" THEN 705                                     JL
·701 PRINT SPC(7);"BALANCE";SPC(9);"PAYME
NT";SPC(8);                                                 HP
·702 PRINT"INTEREST";SPC(7);"PRINCIPLE":G
OTO 710                                                     JD
·705 PRINT" BALANCE";SPC(3);"PAYMENT";SPC
(2);"INTEREST";SPC(1);"PRINCIPLE"                           JE
·710 X=1                                                    DM
·720 GOSUB 3000                                             FE
·725 GOSUB 4000                                             FB
·730 IF X=50 AND X2<1THEN PRINT CHR$(12):
X=1:X2=X2+1:GOTO 700                                        HK
·740 IF X2>0 AND X=60 THEN PRINT CHR$(12)
:X=1:GOTO 700                                               ML
·750 IF LEN(A$)<=8 THEN A1=9-LEN(A$)                        OE
·751 IF LEN(P$)<=8 THEN P1=9-LEN(P$)                        AH
·752 IF LEN(C$)<=8 THEN C1=9-LEN(C$)                        CG
·753 IF LEN(J$)<=8 THEN J1=9-LEN(J$)                        BJ
·760 IF S$="S" THEN 765                                     KJ
·761 PRINT SPC(A1+6);A$;SPC(P1+6);P$;SPC(
C1+6);C$;SPC(J1+6);J$:GOTO 770                              ID
·765 PRINT SPC(A1);A$;SPC(P1);P$;SPC(C1);
C$;SPC(J1);J$                                               CL
·770 A=A+C-P:C=A*I/N:J=P-C                                  PC
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 121 and 122 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·820 IF A+C<=P THEN P=A+C:J=A                    NO
·825 IF A<=.01 THEN 845                          PL
·830 X=X+1                                       DJ
·840 GOTO 720                                    CJ
·845 IF S$="S"THENEND                            HI
·850 PRINT#1:CLOSE 1:END                         ID
·999 END                                         IC
·1000 K=ABS(K):RETURN                            DE
·2000 B=INT(B*100+.001):B$=STR$(B):B$=LEF
 T$(B$,LEN(B$)-2)+"."+RIGHT$(B$,2)               JM
·2090 RETURN                                     IM
·3000 REM TRUNCATE P,C AND Z                     NA
·3010 B=P:GOSUB 2000                             NB
·3020 P$=B$                                      IO
·3030 B=C:GOSUB 2000                             MI
·3040 C$=B$                                      IH
·3050 B=Z:GOSUB 2000                             OD
·3060 Z$=B$                                      JE
·3090 RETURN                                     IM
·4000 REM TRUNCATE A AND J                       FH
·4010 B=J:GOSUB 2000                             ND
·4020 J$=B$                                      IE
·4030 B=A:GOSUB 2000                             LG
·4040 A$=B$                                      IJ
·4090 RETURN                                     IM
```

CHOPPER FLIGHT

FROM PAGE 117

```
·2 REM CHOPPER FLIGHT BY MIKE BUHIDAR JR.
  AND KEVIN WORAM                                PD
·4 CN=0:POKE832,0:BC=53280:GOTO12                BL
·6 PRINTTAB(10);                                 AD
·8 D=D-8:POKECP,D:POKEPL,192:FORW=1TO12:N
 EXT:POKEPL,193                                  LC
·10 GOSUB122:FORW=1TO12:NEXT:RETURN              ML
·12 M1=0:GOSUB258:GOSUB132:FA=3:D=215            IF
·14 POKEV,164:POKEV+31,0:GOSUB302                OK
·16 PRINT"[CLEAR][WHITE]ENTER SKILL LEVEL
  (1-6)"                                         BG
·18 RM=14:WS=11:GOSUB258                         EH
·20 GETSL$:IFVAL(SL$)<1ORVAL(SL$)>6THEN20        ED
·22 GOSUB132                                     CG
·24 POKEV+31,0:LS=192:Q=4:M=88:U=83              MH
·26 BR$="[RIGHT][RIGHT][c 2][RVSON][c R][
 c E][c R][c E][c R][c E][c R][c E][c R][
 c E]":B2$="[RIGHT][RIGHT][c 2][RVSON][c
 R][c E][c R][c E][c R][c E][c R][c E][c
 R][c E]":HD=4-SL/3:GH=HD-1:MD=INT(GH*30)        IP
·28 WN$="[RIGHT][RIGHT][c 2][RVSON][c E][
 c R][c E][RVSOFF] [RVSON][c R][c E][c R]
 [RVSOFF] [RVSON][c E][c R][RVSOFF][c 5][
 s 0][c Y][c V]":W2$="[c 5][c C][c Y][s P
 ][c 2][RVSON][c E][c R][c E][RVSOFF] [RV
 SON][c R][c E][c R][RVSOFF] [RVSON][c E]
 [c R][RVSOFF]":
 IFHD>3THENHD=3                                  GD
·30 TR$="[RIGHT][RIGHT][c 2][RVSON][c E][
 c R][c E][c 5][4"[c Y]"][c 2][c R][c E][
 c R][RVSOFF]":OS$="[RIGHT][RIGHT][c 2][R
 VSON][c R][c E][c R][RVSOFF][4" "][RVSON
 ][c E][c R][c E][RVSOFF]"                       KL
·32 F$="[RIGHT][c 5][s @][RVSON][10" "][R
 VSOFF][s L][RVSOFF]":SB=13+(2*SL):POKEZ,
 PEEK(Z)AND239:TD=(7-SL)*10                      BN
·34 PRINT"[CLEAR][5"[DOWN]"]":RF$="[RIGHT
 ][c 5][s P][RVSON][10"[c P]"][RVSOFF][s
 O]":PRINTRF$SPC(RM)RF$:POKEV+1,Y:POKEV,X        EG
·36 GOSUB374                                     CO
·38 PRINT"[UP]"BR$SPC(N)BR$                      PL
·40 FORP=1TO5:PRINTWN$SPC(WS)W2$:PRINTBR$
 SPC(N)BR$:PRINTB2$SPC(N)B2$:NEXT                OF
·42 POKEZ,PEEK(Z)OR16                            MD
·44 TI$="[6"0"]":FORL4=1TOSB:FORK=1TOQ           PL
·46 PRINTWN$SPC(WS)W2$:GOSUB80                   LM
·48 PRINTBR$SPC(N)BR$:GOSUB80                    GH
·50 PRINTB2$SPC(N)B2$:GOSUB80:NEXT               OO
·52 MP=INT(RND(1)*2)*RM:PRINTTAB(13+MP)"[
 c 1][UP][s X]":NEXT                             PA
·54 PRINTWN$SPC(WS)W2$:GOSUB80                   LM
·56 PRINTTR$SPC(15)TR$:GOSUB80:FORK=1TO3:
 PRINTOS$SPC(15)OS$:GOSUB80:NEXT                 DH
·58 PRINTOS$SPC(5)"[c 5][6"[c P]"]"SPC(4)
 OS$                                             DI
·60 PRINTF$SPC(4)"[c 5][6"[s W]"]"SPC(4)F
 $;                                              FI
·62 PRINT"[c 5] [39"[c Y]"]":GOSUB80             IL
·64 CP=833:FORP=PEEK(V+1)TO211STEP2:D=D+8
 :POKEV+1,P:GOSUB8:NEXT:POKEPL,193               ML
·66 PRINT"[HOME][YELLOW][18"[DOWN]"][15"[
 RIGHT]"]YOU DID IT!":FORM=1TO2000:NEXT           OD
·68 ZZ=VAL(TI$):FORHH=0TO1000:NEXT:POKEV,
 0:GOSUB358                                      CI
·70 PRINT"[WHITE][6" "]PRESS TRIGGER TO P
 LAY AGAIN."                                     IH
·72 B=PEEK(JL)AND16:IFB=0THEN76                  OK
·74 GOTO72                                       PD
·76 CLR:GOTO16                                   MC
·78 REM JOYSTICK ROUTINE                         EM
·80 FR=(PEEK(JL)AND16)/16+1:ONFRGOTO110,1
 16                                              EP
·82 SP=192:XD=HD:YD=0:RETURN                     JD
·84 SP=194:XD=-HD:YD=0:RETURN                    EG
·86 SP=LS:XD=0:YD=0:RETURN                       BN
·88 SP=LS:YD=-HD:XD=0:RETURN                     IL
·90 SP=LS:YD=HD:XD=0:RETURN                      HH
·92 SP=194:XD=-HD:YD=-HD:RETURN                  PI
·94 SP=194:XD=-HD:YD=HD:RETURN                   HP
·96 SP=192:XD=HD:YD=-HD:RETURN                   MH
·98 SP=192:XD=HD:YD=HD:RETURN                    AF
·100 RETURN                                      IM
```


www.commodore.ca

```
·102 POKEBC,8:RETURN                                    FJ
·104 POKEBC,2:RETURN                                    EL
·106 POKEHF,20:X1=X:POKEV+40,2:GOSUB284                 LJ
·108 REM SLOWER FALL                                    JN
·110 GOSUB122:FA=FA+2:IFFA>50THEN252                    PJ
·112 GOSUB240:RETURN                                    OL
·114 REM FASTER FALL                                    IC
·116 GOSUB122:IFFA<1THENFA=2                            HH
·118 FA=FA-2:GOSUB240:RETURN                            KO
·120 REM SOUND                                          BM
·122 POKEHF,7:POKELF,53:POKEHF,0:POKELF,0
  :POKEHF,7:POKELF,163:POKELF,0:POKEHF,0                II
·124 FORG1=0TO(50-FA):NEXT                              EE
·126 POKEHF,7:POKELF,53:POKEHF,0:POKELF,0
  :POKEHF,7:POKELF,163:POKELF,0:POKEHF,0                II
·128 RETURN                                             IM
·130 SPRITE INITIALIZATION                              DG
·132 PRINT"[CLEAR]":POKEBC,0:POKEBC+1,0                 NJ
·134 V=53248:PL=2040:POKEV+21,7:X=170:Y=1
  00:SP=192:POKEV+39,15:POKEPL,SP LOOP FOR              FB
·136 POKEPL+1,196:POKEV+40,12                           OO
·138 POKEPL+2,197:POKEV+28,4:POKEV+41,8:P
  OKEV+37,7:POKEV+38,2                                  HH
·140 POKEV+29,4:POKEV+23,4:PRINT"[CLEAR][
  WHITE]READING DATA[3"."]"                             CI
·142 IFPEEK(12660)=150THEN148                           FH
·144 R=12288:FORG=1TO6:FORI=1TO63:READA:D
  C=DC+A:POKER,A:R=R+1:NEXT:R=R+1:NEXT                  IP
·146 IFDC<>27628THENPRINT"[CLEAR]ERROR IN
   DATA. . .":STOP                                      LG
·148 JL=56320:N=15:Z=53265:CD=53269                     BD
·150 POKEZ,PEEK(Z)AND247:POKEZ,(PEEK(Z)AN
  D248)+7:RETURN                                        JP
·152 REM SPRITE DATA                                    IB
·154 DATA0,0,0,0,0,0,0,0,0,0,1,255,255,0,1,
  0,0,7,192,0,31,240                                    DD
·156 DATA192,63,136,224,63,4,255,255,2,25
  5,255,130,0,63,130,0,47,252,0,15,248                  FA
·158 DATA0,6,248,1,4,17,1,140,27,0,255,25
  4,0,0,0,0,0,0,0,0,0,0                                 JN
·160 DATA0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,
  7,192,0,31,240                                        HF
·162 DATA192,63,136,224,63,4,255,255,2,25
  5,255,130,0,63,130,0,47,252,0,15,248                  FA
·164 DATA0,6,248,1,4,17,1,140,27,0,255,25
  4,0,0,0,0,0,0,0,0,0,0                                 JN
·166 DATA0,0,0,0,0,0,0,0,0,0,255,255,128,0,
  128,0,3,224,0,15,248,0                                JB
·168 DATA17,252,3,32,252,7,64,255,255,65,
  255,255,65,252,0,63,248,0,31,240,0                    DN
·170 DATA12,96,0,136,32,128,216,49,128,12
  7,255,0,0,0,0,0,0,0,0,0,0                             BK
·172 DATA0,0,0,0,0,0,0,0,0,0,0,128,0,0,128,
  0,3,224,0,15,248,0                                    FH
·174 DATA17,252,3,32,252,7,64,255,255,65,
  255,255,65,252,0,63,248,0,31,240,0                    DN
·176 DATA12,96,0,136,32,128,216,49,128,12
  7,255,0,0,0,0,0,0,0,0,0,0                             BK
·178 DATA0,0,0,0,62,0,0,119,0,0,239,128,0
  ,207,128,0,255,128,0,0,0                              JO
·180 DATA0,127,0,0,127,0,0,127,0,0,62,0,0
  ,62,0,0,62,0,0,28,0                                   HA
·182 DATA0,127,0,0,235,128,0,193,128,1,12
  8,192,1,128,192,1,0,64,1,0,64                         HA
·184 DATA0,20,0,0,85,0,1,150,64,5,105,80,
  5,170,80,38,170,152,42,170,168                        GK
·186 DATA46,170,184,91,190,229,122,255,17
  3,119,255,221,90,255,165,27,190,232                   EP
·188 DATA46,170,184                                     GP
·190 DATA42,170,168,5,170,80,5,105,80,1,1
  50,64,0,85,0,0,20,0,0,0,0                             FC
·192 REM SPRITE-DATA COLLISION                          BF
·194 XP=X-24:YP=Y-54:CX=INT(XP/8):CY=INT(
  YP/8):BB=1104+CX+(40*CY)                              GK
·196 P1=PEEK(BB):P2=PEEK(BB+1):P3=PEEK(BB
  +2):P4=PEEK(BB+3)                                     NI
·198 IFP1=UORP2=UORP3=UORP4=UTHENRETURN                 LE
·200 IFP1=MORP2=MORP3=MORP4=MTHEN204                    DN
·202 GOTO206                                            CE
·204 POKEH1,50:FORL=0TO49:NEXT:POKEH1,0                 CD
·206 IFP1=(M)THEN226                                    KK
·208 IFP2=(M)THEN228                                    KD
·210 IFP3=(M)THEN230                                    KN
·212 IFP4=(M)THEN232                                    KK
·214 REM PUT EXPLOSION HERE                             EN
·216 E$="CRASHED INTO A BUILDING,"                      IP
·218 POKEHF,7:X=X-12:Y=Y-10:POKEV+4,X:POK
  EV+5,Y:FORWA=1TO255:POKELF,WA:NEXT                    EE
·220 POKEV+21,0                                         FF
·222 POKEHF,0:POKELF,0:GOSUB268                         LM
·224 REM PICK UP MAN HERE                               GN
·226 MS=MS+1:POKEBB,U:RETURN                            HC
·228 MS=MS+1:POKEBB+1,U:RETURN                          DC
·230 MS=MS+1:POKEBB+2,U:RETURN                          EN
·232 MS=MS+1:POKEBB+3,U:RETURN                          EI
·234 POKEHF,7:X=X-12:Y=Y-10:POKEV+4,X:POK
  EV+5,Y:FORWA=1TO255:POKELF,WA:NEXT                    EE
·236 POKEHF,0:POKELF,0:RETURN                           CI
·238 REM MOVEMENT ROUTINE                               CF
·240 JV=N-(PEEK(JL)ANDN)+1:TL=INT((VAL(TI
  $))/TD)+1:GOSUB292                                    MJ
·242 IFPEEK(V+31)ANDX=XTHENGOSUB192                     AL
·244 ONTLGOSUB100,100,100,102,104,106                   GH
·246 ONJVGOSUB86,88,90,100,84,92,94,100,8
  2,96,98:LS=SP                                         CI
·248 POKEPL,SP:X=X+XD:POKEV,X:Y=PEEK(V+1)
  +YD:POKEV+1,Y:POKEPL,SP+1:RETURN                      PO
·250 REM HOVERING ROUTINE                               DC
·252 FR=(PEEK(JL)AND16)/16+1:IFFR=2THENFA
  =48:GOTO110                                           GI
·254 GOSUB240                                           CG
·256 GOSUB122:GOTO252                                   II
·258 REM SOUND INITIALIZATION                           DL
·260 S=54272:FORQ=STOS+24:POKEQ,0:NEXT                  GE
·262 POKES+24,15:POKES+5,18:POKES+6,33                  LD
·264 POKES+4,129:HF=54273:LF=54272                      BL
```


www.commodore.ca

```
·266 POKES+12,255:POKES+13,255:POKES+11,1
 7:H1=54280:L1=54279:RETURN                DO
·268 REM END ROUTINE                       IP
·270 POKEBC,0:CN=PEEK(832):CN=CN+1         GF
·272 PRINT"[DOWN][CLEAR][WHITE]CHOPPER V
 -"CN;E$                                   MD
·274 PRINT"[DOWN]SEND FOR REPLACEMENT PIL
 OT IMMEDIATELY![DOWN]":POKEV+5,0:POKE832
 ,CN                                       FK
·276 FORK=0TO999:NEXT:ZZ=VAL(TI$):GOSUB35
 8                                         OF
·278 GOTO70                                PF
·280 REM MISSILE MOVING ROUTINE            JD
·282 POKEHF,20:X1=INT(RND(0)*68)+144       FC
·284 FORY1=255TO0STEP-10:POKELF,(255-Y1)   EL
·286 POKEV+2,X1:POKEV+3,Y1:IFPEEK(V+30)AN
 D1=1THENIFPEEK(V+30)AND2=2THEN300         JJ
·288 NEXT:POKEV+31,0:M1=0:RETURN           HI
·290 REM MISSILE SELECT                    KF
·292 M1=M1+1:IFM1<MDTHENRETURN             GK
·294 POKEHF,5:POKELF,5                     EJ
·296 IFM1>MD+10THEN282                     JP
·298 RETURN                                IM
·300 POKEV+2,0:E$="DESTROYED BY ENEMY FIR
 E,":GOTO218                               JP
·302 REM TITLE SCREEN                      NG
·304 FORL=1TO22:PRINT:NEXT                 GG
·306 FORK=1TO3:PRINTTAB(19)"[c 5][s B]":N
 EXT                                       CB
·308 T$(1)="[CYAN][s U][3"[s C]"][s I][s
 U][s I][s U][s I][s U][s C][s C][s I][s
 U][s C][s C][s I][s U][s C][s C][s I][s
 U][s C][s C][s I][s U][3"[s C]"][s I]"    HK
·310 T$(2)="[s B] [s U][s I][6"[s B]"][s
 U][s I][s B][s B][s U][s I][s B][s B][s
 U][s I][s B][s B][s U][s C][s K][s B][s
 U][s I] [s B]"                            KO
·312 T$(3)="[s B] [s B][s J][s K][s B][s
 J][s K][6"[s B]"][s J][s K][s B][s B][s
 J][s K][s B][s B][s J][s C][s I][s B][s
 J][s K] [s B]"                            ND
·314 T$(4)="[s B] [s B][s U][s I][s B][s
 U][s I][6"[s B]"][s U][s C][s K][s B][s
 U][s C][s K][s B][s U][s C][s K][s B][s
 U][s I][s U][s K]"                        LP
·316 T$(5)="[s B] [s J][s K][6"[s B]"][s
 J][s K][3"[s B]"]  [s B][s B]  [s B][s J
 ][s C][s I][4"[s B]"]"                    GD
·318 T$(6)="[s J][3"[s C]"][s K][s J][s K
 ][s J][s K][s J][s C][s C][s K][s J][s K
 ]  [s J][s K]  [s J][s C][s C][s K][s J]
 [s K][s J][s K]"                          AL
·320 T$(7)="":T$(8)="":J$(1)="[s U][3"[s
 C]"][s I][s U][s I]  [s U][s C][s C][s I
 ][s U][s C][s C][s I][s U][s I][s U][s I
 ][s U][4"[s C]"][s I]"                    NE
·322 J$(2)="[s B][SS][s U][s C][s K][s B]
 [s B][SS][SS][s J][s I][s U][s K][s B][s
 U][s C][s K][4"[s B]"][s J][s C][s I][s
 U][s C][s K]"                             MO
·324 J$(3)="[s B][SS][s J][s C][s I][s B]
 [s B][3"[SS]"][s B][s B][SS][s B][s B][s
 U][s I][s B][s J][s K][s B][SS] [s B][s
 B][SS]"                                   DG
·326 J$(4)="[s B][SS][s U][s C][s K][s B]
 [s B][3"[SS]"][s B][s B][SS][5"[s B]"][s
 U][s I][s B][SS] [s B][s B][SS]"          GJ
·328 J$(5)="[s B][SS][s B][SS][SS][s B][s
 J][s C][s I][s U][s K][s J][s I][s B][s
 J][s K][5"[s B]"][SS] [s B][s B][SS]"     HF
·330 J$(6)="[s J][s C][s K][SS][SS][s J][
 s C][s C][s K][s J][s C][s C][s K][s J][
 s C][s C][s K][s J][s K][s J][s K][SS] [
 s J][s K][SS]"                            BM
·332 BL$="[s B][19" "][s B]":N1$="[s B][W
 HITE]MICHAEL BUHIDAR JR.[PURPLE][s B]"    KE
·334 TP$="[PURPLE][4" "][s U][11"[s C]"][
 s I]":DB$="[s U][3"[s C]"][s K][WHITE]DE
 SIGNED BY[PURPLE][s J][3"[s C]"][s I]"    HB
·336 AN$="[PURPLE][s B][8" "][WHITE]AND[P
 URPLE][8" "][s B]":N2$="[s B][WHITE][4"
 "]KEVIN WORAM[4" "][PURPLE][s B]"         GF
·338 CR$="[PURPLE][s B][WHITE]FOR AHOY! M
 AGAZINE [PURPLE][s B]":ED$="[s J][19"[s
 C]"][s K]"                                EN
·339 SP$="[PURPLE][s B][19" "][PURPLE][s
 B]"                                       MK
·340 CP=V+1:FORL=1TO7:GOSUB8:PRINTTAB(5)T
 $(L):NEXT                                 DK
·342 FORL=1TO6:GOSUB8:PRINTTAB(7)J$(L):NE
 XT                                        GC
·344 PRINT:GOSUB8                          JL
·346 GOSUB6:PRINTTP$:GOSUB6:PRINTDB$:GOSU
 B6:PRINTBL$:GOSUB6                        GL
·348 PRINTN1$:GOSUB6:PRINTAN$:GOSUB6:PRIN
 TN2$                                      PG
·350 GOSUB6:PRINTCR$:GOSUB6:PRINTSP$:GOSU
 B6:PRINTED$:GOSUB6                        JE
·352 PRINTTAB(10)"[WHITE]PRESS TRIGGER TO
  PLAY"                                    AB
·354 B=PEEK(JL)AND16:IFB=0THENRETURN        PP
·356 GOTO354                               CE
·358 REM FINAL SCORE                       IG
·360 POKEBC,0:FS=100*MS*SL+(500-ZZ):TS=(4
 0-(14+MS))/2                              EK
·362 IFMS=0THENPRINT"[CLEAR]":PRINTTAB(13
 )"[PURPLE]NO MEN RESCUED[WHITE]":GOTO370  AJ
·364 PRINT"[CLEAR][WHITE]":PRINTTAB(TS)"M
 EN RESCUED:  ";:FORMR=1TOMS:PRINT"[c 1][
 s X]";:NEXT:PRINT                         AJ
·366 PRINTTAB(14)"[WHITE]BONUS:  ";:FORHH
 =1TOMS:O=O+100:POKEH1,100:FORM=1TO150:NE
 XT                                        OD
·368 POKEH1,0:PRINT"[HOME][DOWN][DOWN]"TA
 B(21)O;:NEXT:PRINT                        DH
·370 PRINTTAB(11)"[8"[DOWN]"]YOUR SCORE:
```


www.commodore.ca

```
  "FS"[DOWN]"                                                    AE
·372 FORJJ=0TO1000:NEXT:RETURN                                   OG
·374 REM MOVEMENT CHECKER                                        BE
·376 RESTORE                                                     IO
·378 R=12288:FORG=1TO6:FORI=1TO63:READA:D
 C=DC+A:R=R+1:NEXT:R=R+1:NEXT                                    AP
·380 DATA173,1,208,201,50,144,7,201,229,1
 76,11,76,49,234,105,4,141,1,208,76                              HH
·382 DATA49,234,233,4,141,1,208,76,49,234                        MJ
·384 REM INTERRUPT SETUP                                         OG
·386 DATA120,169,0,141,20,3,169,192,141,2
 1,3,88,96                                                       DA
·388 FORP=49152TO49194:READA:POKEP,A:NEXT                        FM
·390 SYS49182:RETURN                                             GJ
```

FILE SCOUT
FROM PAGE 70

```
·1 REM ********************************                         DN
·2 REM *                              *                         OO
·3 REM *     FILE SCOUT V0619/85      *                         JA
·4 REM *        (RV 0316/84)          *                         GB
·5 REM *                              *                         OO
·6 REM ********************************                         DN
·7 REM                                                           JD
·10 POKE 53280,12:POKE 53281,0 :PRINT CHR
 $(147)CHR$(154);                                                CA
·16 H$="0123456789ABCDEF"                                        OA
·20 GOTO 1000                                                    FC
·50 REM LOCATE                                                   FC
·51 POKE 214,RW:POKE 211,CL:SYS 58640:RET
 URN                                                             AF
·60 REM SET ROW, COLUMNS FOR SCAN                                CG
·61 DIM RW(35),CL(35)                                            GE
·62 FOR I=0 TO 16:RW(I)=I+6:CL(I)=4:NEXT
 I                                                               MC
·63 FOR I=17 TO 34:RW(I)=(I+6)-17:CL(I)=1
 9:NEXT I                                                        AG
·65 RETURN                                                       IM
·99 :                                                            DI
·100 REM ** COLLECT ALL THE ACTIVE FILES
 AND THEIR FIRST TRACK & SECTORS **                              AC
·101 RW=5:CL=6 :GOSUB 50:PRINT "READING D
 IRECTORY";                                                      IM
·105 OPEN 15,8,15,"U;"                                           PH
·110 N1=144:DIM F$(N1),T(N1),S(N1),B(N1),
 FT$(N1)                                                         IA
·115 OPEN 2,8,2,"#2"                                             GP
·120 TD=18:TS=1:NF=0                                             DI
·130 PRINT#15,"U1:2,8,"TD,TS                                     IF
·140 GET#2,A$:TD=ASC(A$+CHR$(0)):GET#2,A$
 :TS=ASC(A$+CHR$(0))                                             EJ
·145 FOR K=0 TO 7                                                IJ
·150 GET#2,A$:IF ASC(A$+CHR$(0))=0 THEN G
 OSUB 300:GOTO 230                                               PJ
·151 A=ASC(A$):IF (A AND 64)=0 THEN FT$(N
 F)="[SS]":GOTO 153                                              PF
·152 FT$(NF)=">"                                                 NI
·153 A=(A AND(255-64))                                           CK
·155 IF A=129 THEN FT$(NF)=FT$(NF)+"SEQ"                         JD
·156 IF A=130 THEN FT$(NF)=FT$(NF)+"PRG"                         IH
·157 IF A=131 THEN FT$(NF)=FT$(NF)+"USR"                         NL
·160 GET#2,A$:GET#2,B$:T(NF)=ASC(A$+CHR$(
 0)):S(NF)=ASC(B$+CHR$(0))                                       PO
·165 FOR J=0 TO 15:GET#2,A$                                      HG
·170 IF A$=CHR$(160) THEN GOTO 190                               CE
·180 F$(NF)=F$(NF)+A$                                            HK
·190 NEXT J                                                      MM
·199 :                                                           DI
·200 FOR J=0 TO 8:GET#2,A$:NEXT J                                FB
·210 GET#2,A$:GET#2,B$:B(NF)=ASC(A$+CHR$(
 0))+ASC(B$+CHR$(0))*256:BA=BA+B(NF)                             NL
·224 NF=NF+1                                                     MG
·225 GET#2,A$:GET#2,A$                                           GE
·230 NEXT K                                                      MP
·240 IF TD<>18 THEN GOTO 245                                     PA
·241 IF TS<1 OR TS>19 THEN GOTO 245                              KA
·242 GOTO 130                                                    CA
·245 IF NF=0 THEN PRINT"NO FILES":END                            JD
·246 CLOSE2:BF=664-BA:RETURN                                     OK
·299 :                                                           DI
·300 FOR J=0 TO 30                                               KF
·310 GET#2,A$:NEXTJ                                              JP
·320 RETURN                                                      IM
·399 :                                                           DI
·400 REM SCAN FOR T & S                                          GP
·405 RW=24:CL=1:GOSUB 50:PRINT BL$;                              FK
·410 RW=23:CL=2:GOSUB 50:F$="":K1=-1:PRIN
 T CHR$(158);:INPUT "FILE NAME";F$                               DG
·411 IF F$="" THEN RW=23:CL=1:PRINT BL$;:
 RETURN                                                          AE
·412 LN=LEN(F$):IF RIGHT$(F$,1)="*" THEN
 LN=LEN(F$)-1:F$=LEFT$(F$,LN)                                    IF
·420 FOR K=0 TO NF                                               OP
·422 IF K1<>-1 THEN GOTO 430                                     CF
·425 IF F$=LEFT$(F$(K),LN) THEN K1=K                             MJ
·430 NEXT K                                                      MP
·431 IFK1=-1THENPRINT "[4" "]NOT FOUND";:
 FOR K=0TO2000:NEXTK:PRINTCHR$(154);:RETU
 RN                                                              II
·435 TR=T(K1):SC=S(K1)                                           EF
·440 BC=0:C=0:T4$="TRK SEC"                                      HB
·445 PRINT#15,"I0:":CLOSE2: OPEN 2,8,2,"#
 2"                                                              KM
·450 PRINT HD$;:RW=2:CL=1:GOSUB 50:PRINT
 F$(K1);                                                         MB
·455 RW=4:CL=6:GOSUB 50:PRINT T4$;                               AD
·460 RW=RW(C):CL=CL(C):GOSUB 50:PRINT BC;
 TR;SC                                                           KP
·465 GOSUB 590:IF TR=0 THEN GOTO 480                             HH
·470 BC=BC+1:C=C+1                                               OE
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 121 and 122 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·475 IF C<34 THEN GOTO 460                                    LO
·480 RW=24:CL=10:GOSUB 50:PRINT CHR$(5) "
 NEXT  PRINT  RETURN ";                                       KI
·482 GOSUB 700:PRINT A$CHR$(158);                             GM
·490 IF A$="N" AND BC<B(K1)-1 THEN C=0:GO
 TO 450                                                       AN
·492 IF A$="N" AND BC=B(K1)-1 THEN GOTO 4
 35                                                           MA
·495 IF A$="P" THEN GOSUB 500:GOTO 435                        FC
·496 IF A$="R" THEN CLOSE 2:PRINT CHR$(15
 4);:RETURN                                                   HJ
·497 GOTO 482                                                 CK
·499 :                                                        DI
·500 REM PRINT OUT FOR T & S                                  KJ
·505 OPEN 4,4:BC=0                                            ME
·510 PRINT#4,F$(K1)                                           EB
·515 TR=T(K1):SC=S(K1)                                        EF
·520 PRINT#4,CHR$(13)CHR$(13)T4$CHR$(13)C
 HR$(13)                                                      OA
·525 PRINT#4,BC,TR,SC                                         BM
·530 GOSUB 590:BC=BC+1                                        DB
·535 IF TR=0 THEN PRINT#4:CLOSE4:RETURN                       HA
·540 GOTO 525                                                 CK
·590 PRINT#15,"U1:2,8",TR,SC                                  FG
·595 GET#2,A$:TR=ASC(A$+CHR$(0))                              GK
·596 GET#2,A$:SC=ASC(A$+CHR$(0))                              NK
·597 RETURN                                                   IM
·599 :                                                        DI
·600 REM ** GET LOAD ADDR PRG FILES **                        IN
·605 DIM LA(N1),LH$(N1)                                       KI
·610 OPEN 2,8,2,"#2"                                          GP
·620 FOR I=0 TO NF-1                                          NI
·625 IF RIGHT$(FT$(I),3)<>"PRG" THEN LA(I
 )=-1:GOTO 650                                                EF
·630 PRINT#15,"U1:2,8,"T(I),S(I)                              DD
·632 PRINT#15,"B-P:"2,0                                       OL
·635 GET#2,A$:GET#2,A$:GET#2,A$:GET#2,B$                      AE
·636 A=ASC(A$+CHR$(0)):B=ASC(B$+CHR$(0))                      NC
·640 LA(I)=A +B*256                                           HH
·641 BH=INT(B/16):BL=B-(BH*16)                                OO
·642 AH=INT(A/16):AL=A-(AH*16)                                ND
·643 LH$(I)=MID$(H$,BH+1,1)+MID$(H$,BL+1,
 1)                                                           OF
·644 LH$(I)="$"+LH$(I)+MID$(H$,AH+1,1)+MI
 D$(H$,AL+1,1)                                                KB
·650 NEXT I                                                   MN
·655 CLOSE 2:RETURN                                           JP
·699 :                                                        DI
·700 REM GET CHAR FROM KB, WITH CURSOR                        DA
·705 POKE 198,0:POKE 204,0                                    HG
·710 GET A$:POKE 207,0:IF A$="" THEN GOTO
  710                                                         CK
·715 POKE 204,1:PRINT " "CHR$(157);:RETUR
 N                                                            KP
·725 CLOSE15:OPEN 15,8,15,"I0:"                               FK
·730 INPUT#15,A$,B$,C$,D$:CLOSE15                             IJ
·735 IF A$="00" THEN RETURN                                   HB
·740 PRINT:PRINT:PRINT"DISK ERROR":STOP                       EE
·749 :                                                        DI
·750 REM GET DISK NAME & ID                                   GL
·760 OPEN 15,8,15,"U;"                                        PH
·770 OPEN 2,8,2,"#2"                                          GP
·775 TD=18:TS=0:PRINT#15,"U1:2,8,"TD,TS                       PH
·780 PRINT#15,"M-R"CHR$(144)CHR$(5)CHR$(2
 0)                                                           LP
·782 FOR I=0 TO 19:GET#15,A$:DN$=DN$+A$:N
 EXT I                                                        IH
·795 CLOSE2:CLOSE15:RETURN                                    ON
·799 :                                                        DI
·800 T1$=DN$+NF$                                              HA
·801 T2$="BLOCKS ALLOC:[SS]"+STR$(BA)+" B
 LOCKS FREE: "+STR$(BF)+" "                                   MH
·802 T3$="FILE[12" "]TYPE TRK SEC BLK ADD
 R.D ADDR.H":S1$="[5" "]"                                     KP
·805 OPEN 4,4:CMD 4                                           EL
·808 PRINT S1$T1$:PRINT S1$T2$:PRINT:PRIN
 T S1$T3$:PRINT                                               JE
·810 FOR K=0 TO NF-1                                          MO
·820 PRINT S1$F$(K);:FOR J=0TO16-LEN(F$(K
 )):PRINT" ";:NEXT J:PRINT FT$(K);                            JM
·829 SP$="[6" "]"                                             HK
·830 A$=STR$(T(K)):A$=A$+RIGHT$(SP$,3-LEN
 (A$))                                                        JH
·831 B$=STR$(S(K)):B$=B$+RIGHT$(SP$,3-LEN
 (B$))                                                        BG
·832 C$=STR$(B(K)):C$=C$+RIGHT$(SP$,4-LEN
 (C$))                                                        JC
·840 PRINT A$SPC(1)B$SPC(1)C$SPC(1);:IF L
 A(K)<0 THEN PRINT SP$;:GOTO 845                              ID
·841 D$=STR$(LA(K)):D$=D$+RIGHT$(SP$,6-LE
 N(D$)):PRINT D$SPC(1)                                        KC
·845 PRINT LH$(K)                                             PI
·850 NEXT K                                                   MP
·860 PRINT#4,CHR$(13):PRINT#4:CLOSE4:RETU
 RN                                                           HJ
·899 :                                                        DI
·1000 REM ** MAIN ROUTINE **                                  JB
·1001 R0$=CHR$(146):R9$=CHR$(18):CL$=CHR$
 (147)                                                        KD
·1002 BL$="[38" "]"                                           OL
·1003 HD$=CL$+R9$+"[11" "]FILE SCOUT V061
 9/85[10" "]"+R0$                                             JL
·1004 PRINT HD$;:GOSUB 60                                     LP
·1005 RW=3:CL=1:GOSUB 50:PRINT "INSERT DI
 SK AND HIT A KEY ";                                          JN
·1006 GOSUB 700:GOSUB 725:GOSUB 750                           PE
·1010 GOSUB 100:GOSUB 600                                     IH
·1015 PRINT                                                   JJ
·1400 NF$=STR$(NF):L=LEN(NF$)                                 CN
·1401 FL$=RIGHT$(BL$,9-L)                                     HM
·1405 NF$= " # FILES : "+NF$+FL$                              IK
·1410 TH$=CL$+R9$+DN$+NF$                                     AB
```


www.commodore.ca

```
·1501 SP$="[9" "]":L=LEN(STR$(BA))+LEN(ST
 R$(BF)):FL$=RIGHT$(SP$,14-L)                IJ
·1502 TJ$=R9$+"BLOCKS ALLOC:"+STR$(BA)+"
 BLOCKS FREE:"+STR$(BF)+FL$                  IL
·1503 TL$=R9$+"FILE[13" "]TYPE TRK SEC BL
 K ADDR  "+R0$:I=0                           MB
·1505 PRINT TH$;:PRINT TJ$:PRINT TL$:PRIN
 T                                           HN
·1520 PRINT F$(I);:FOR J=0TO16-LEN(F$(I))
 :PRINT" ";:NEXT J:PRINT FT$(I);             FE
·1529 SP$="[6" "]"                           HK
·1530 A$=STR$(T(I)):A$=A$+RIGHT$(SP$,3-LE
 N(A$))                                      CN
·1531 B$=STR$(S(I)):B$=B$+RIGHT$(SP$,3-LE
 N(B$))                                      BM
·1532 C$=STR$(B(I)):C$=C$+RIGHT$(SP$,4-LE
 N(C$))                                      OI
·1540 PRINT A$SPC(1)B$SPC(1)C$;:IF LA(I)<
 0 THEN GOTO 1545                            AA
·1541 PRINT STR$(LA(I))                      ND
·1545 PRINT SPC(34)LH$(I)                    EO
·1546 I=I+1:IF I=NF THEN GOTO 1548           IN
·1547 IF I/8-INT(I/8)<>0 THEN GOTO 1520      CJ
·1548 RW=24:CL=5:GOSUB 50:PRINT CHR$(5)"
 NEXT SCOUT PRINT BOOT QUIT?[SS]";           HG
·1549 GOSUB 700:PRINT A$;:PRINT CHR$(154)
 ;                                           EN
·1550 IF A$="P" THEN GOSUB 800:PRINT CHR$
 (20);:GOTO 1548                             EL
·1555 IF A$="S" THEN GOSUB 400:GOTO 1580     MO
·1560 IF A$="Q" THEN STOP                    HE
·1565 IF A$="B" THEN CLR:GOTO 10             KH
·1570 IF A$<>"N" THEN GOTO 1548              PG
·1575 IF I<NF THEN GOTO 1505                 CP
·1580 I=0:GOTO 1505                          MN
·5999 :                                      DI
·6000 CLOSE15:OPEN 15,8,15,"I0:":PRINT#15
 ,"S0:FILE SCOUT V0619":CLOSE15              MI
·6005 SAVE "FILE SCOUT V0619",8:STOP         HN
·6100 CLOSE15:OPEN 15,8,15:INPUT#15,A$,B$
 ,C$,D$:CLOSE15                              CE
·6101 PRINT A$"[SS]"B$"[SS]"C$"[SS]"D$:ST
 OP                                          MG
```

THE KNIGHT'S TOUR

FROM PAGE 73

```
·100 REM * KNIGHT'S TOUR / COMMODORE 64 /
  RAMELLA                                    OD
·101 POKE 53280,0: POKE 53281,0              BH
·102 PRINT "[CLEAR][c 3]": V=53248: POKE
 V+34,4: POKE 53269,4: POKE 2042,13          JK
·103 FOR N=0 TO 62: READ A: POKE 832+N,A:
  NEXT: L=53: H=28                           LH
·105 PRINT"[WHITE]";SPC(33);"KNIGHT": PRI
 NT: PRINT SPC(33);"TOUR": PRINT "[RED]"     ML
·106 PRINT: PRINT: PRINT: PRINT: PRINT "[
 BLUE]":PRINT SPC(33);"SCORE:": PRINT "[H
 OME]";                                      HM
·107 PRINT "[RED]"                           PP
·110 T$="[s 0][3"[c Y]"][s 0][3"[c Y]"][s
  0][3"[c Y]"][s 0][3"[c Y]"][s 0][3"[c Y
 ]"][s 0][3"[c Y]"][s 0][3"[c Y]"][s 0][3
 "[c Y]"][c H]"                              KO
·112 N$="[c H][3" "][c H][3" "][c H][3" "
 ][c H][3" "][c H][3" "][c H][3" "][c H][
 3" "][c H][3" "][c H]"                      EA
·115 FOR X=1 TO 8: PRINT T$: PRINT N$: PR
 INT N$: NEXT                                EI
·116 PRINT"[UP][s L][3"[c P]"][s L][3"[c
 P]"][s L][3"[c P]"][s L][3"[c P]"][s L][
 3"[c P]"][s L][3"[c P]"][s L][3"[c P]"][
 s L][3"[c P]"]"                             NJ
·120 GOSUB 5000                              FK
·150 GETA$: IF A$<>"A" AND A$<>"Z" AND A$
 <>"," AND A$<>"." AND A$<>"K" THEN 150      HA
·152 IF A$="K" THEN 245                      GM
·155 IF H=252 AND A$="." THEN 150            JB
·160 IF A$="." THEN FOR H=H TO H+32 STEP
 4                                           IJ
·165 IF H=28 AND A$="," THEN 150             IH
·170 IF A$="," THEN FOR H=H TO H-32 STEP
 -4                                          GJ
·175 IF L=221 AND A$="Z" THEN 150            MD
·180 IF A$="Z" THEN FOR L=L TO L+24 STEP
 2                                           FE
·185 IF L=53 AND A$="A" THEN 150             GC
·190 IF A$="A" THEN FOR L=L TO L-24 STEP
 -2                                          EL
·200 GOSUB 5000: NEXT                        BL
·210 IF A$="." THEN H=H-4: P=P+4             FD
·220 IF A$="," THEN H=H+4: P=P-4             LH
·230 IF A$="Z" THEN L=L-2: P=P+120           OP
·235 IF A$="A" THEN L=L+2: P=P-120           AA
·240 GOTO 150                                CG
·245 S$="": H1=H: L1=L: P1=P                 OP
·250 GET A$: IF A$<>"A" AND A$<>"Z" AND A
 $<>"," AND A$<>"." THEN 250                 CI
·252 S$=S$+A$                                KE
·255 IF H=252 AND A$="." THEN 385            JL
·260 IF A$="." THEN FOR H=H TO H+32 STEP
 4                                           IJ
·265 IF H=28 AND A$="," THEN 385             IL
·270 IF A$="," THEN FOR H=H TO H-32 STEP
 -4                                          GJ
·275 IF L=221 AND A$="Z" THEN 385            OF
·280 IF A$="Z" THEN FOR L=L TO L+24 STEP
 2                                           FE
·285 IF L=53 AND A$="A" THEN 385             GG
·290 IF A$="A" THEN FOR L=L TO L-24 STEP
 -2                                          EL
·300 GOSUB 5000: NEXT                        BL
·310 IF A$="." THEN H=H-4: P=P+4             FD
·320 IF A$="," THEN H=H+4: P=P-4             LH
·330 IF A$="Z" THEN L=L-2: P=P+120           OP
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 121 and 122 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·335 IF A$="A" THEN L=L+2: P=P-120                    AA
·337 IF LEN(S$)=3 THEN 350                            FD
·340 GOTO 250                                         CD
·350 IF S$="AA." OR S$=".AA" OR S$="A.."
 OR S$="..A" OR S$="..Z" THEN W=1                     DF
·360 IF S$="Z.." OR S$="ZZ." OR S$=".ZZ"
 OR S$="ZZ," OR S$=",ZZ" THEN W=1                     JJ
·370 IF S$=",,Z"OR S$="Z,,"OR S$=",,A"OR
 S$="A,,"OR S$="AA,"OR S$=",AA"THEN W=1               GG
·375 IF PEEK(1024+P)=81 THEN 385                      ML
·380 IF W=1 THEN W=0: GOTO 400                        OM
·385 P=P1: H=H1: L=L1: GOSUB 5000: W=0: G
 OTO 245                                              DO
·390 GOTO 245                                         CJ
·400 FOR G=P TO P+80 STEP 40: FOR R=G TO
 G+3                                                  OL
·410 POKE 1024+R,81: POKE 55296+R,14: NEX
 T R,G                                                KM
·411 TL=TL+1: TL$=STR$(TL): FOR JF=2 TO L
 EN(TL$)                                              PH
·412 POKE 1537+JF,ASC(MID$(TL$,JF,1)): NE
 XT JF                                                BH
·420 GOTO 245                                         CJ
·5000 POKE V+4,H: POKE V+5,L: RETURN                  JK
·10000 DATA 0,31,128,0,255,224,1,255,240              IP
·10010 DATA 3,255,248,7,231,254,15,255,25
 2                                                    AM
·10020 DATA 31,255,252,63,255,255,127,255
 ,252                                                 HE
·10030 DATA 255,255,254,255,255,255,127,1
 59,252                                               CN
·10040 DATA 60,15,254,0,31,255,0,31,252               JM
·10050 DATA 0,63,254,0,255,255,1,255,252              GF
·10060 DATA 7,255,254,31,255,255,63,255,2
 55                                                   BD
·10070 END                                            IC
```

RHYTHMIC BITS
FROM PAGE 76

```
·40 POKE 53280,1:POKE53281,3                          BP
·50 POKE 650,255                                      AD
·100 GOSUB 2000                                       FL
·200 PRINT"[CLEAR]"                                   HH
·210 PRINT"[13" "]RHYTHMIC BITS"                      AD
·220 PRINT"[12" "]BY DAVID BARRON"                    OC
·230 PRINT"[6"[DOWN]"](F1) - DEFINE SOUND
 S"                                                   JA
·240 PRINT"[DOWN](F3) - ENTER RHYTHMIC PA
 TTERN"                                               HG
·250 PRINT"[DOWN](F5) - CHANGE FILTER"                LA
·255 PRINT"[DOWN](F6) - SET SPEED"                    OA
·260 PRINT"[DOWN](F7) - PLAY PATTERN"                 PC
·270 PRINT"[DOWN][DOWN][4" "]- ENTER YOUR
  SELECTION -"                                        AO
·280 GET T$:IF T$="" THEN 280                         JP
·290 IF T$="[F1]" THEN 1000                           PJ
·295 IF T$="[F3]" THEN 5500                           AP
·300 IF T$="[F5]" THEN 350                            MG
·305 IF T$="[F6]" THEN 500                            OF
·310 IF T$="[F7]" THEN 6000                           AL
·320 GOTO 280                                         CG
·350 PRINT "[CLEAR][12" "]FILTER CHANGE"              EP
·352 PRINT"[13"[DOWN]"](F1) - INCREASE"               AF
·353 PRINT"(F3) - DECREASE"                           HM
·354 PRINT"(F7) - QUIT"                               EF
·360 PRINT"[HOME][6"[DOWN]"]LOW:[31" "]:H
 IGH"                                                 EG
·380 PRINT"[HOME][5"[DOWN]"]"                         EP
·385 REM                                              JD
·390 PRINT TAB(5+FI/10);" [RVSON] [RVSOFF
 ] [UP]"                                              OC
·400 GET T$:IF T$="" THEN 400                         JB
·410 IF T$="[F1]" THEN FI=FI+1:IF FI>255
 THEN FI=255                                          FB
·420 IF T$="[F3]" THEN FI=FI-1:IF FI<0 TH
 EN FI=0                                              GP
·425 IF T$="[F7]" THEN 440                            NP
·430 GOTO 385                                         CK
·440 POKE SID+22,FI                                   FC
·450 GOTO 200                                         BO
·500 PRINT "[CLEAR][12" "]SPEED CHANGE"               CH
·552 PRINT"[13"[DOWN]"](F1) - DECREASE"               HB
·553 PRINT"(F3) - INCREASE"                           GK
·554 PRINT"(F7) - QUIT"                               EF
·560 PRINT"[HOME][5"[DOWN]"]FAST:[30" "]:
 SLOW"                                                NA
·580 PRINT"[HOME][4"[DOWN]"]"                         DC
·585 REM                                              JD
·590 PRINT TAB(5+SP/10);" [RVSON] [RVSOFF
 ] [UP]"                                              BM
·600 GET T$:IF T$="" THEN 600                         IL
·610 IF T$="[F1]" THEN SP=SP+1:IF SP>254
 THEN SP=254                                          IH
·620 IF T$="[F3]" THEN SP=SP-1:IF SP<0 TH
 EN SP=0                                              HH
·625 IF T$="[F7]" THEN 640                            NB
·630 GOTO 585                                         DE
·640 POKE 49164,SP+1                                  EI
·650 GOTO 200                                         BO
·1000 PRINT"[CLEAR]":POKE 650,255:IF V>7
 THEN V=0                                             GE
·1020 PRINT"[DOWN][DOWN]"                             PN
·1030 PRINT"[6"[DOWN]"]FREQUENCY:[DOWN]"              FE
·1040 PRINT      "DECAY[4" "]:[DOWN]"                 BF
·1050 PRINT"WAVE TYPE:"                               LL
·1051 PRINT "[DOWN]USE (F1) FOR FREQUENCY
  INCREASE"                                           OO
·1052 PRINT "USE (F2) FOR FREQUENCY DECRE
 ASE"                                                 NC
·1053 PRINT "USE (F3) FOR ATTACK INCREASE
 "                                                    BA
```


www.commodore.ca

```
·1054 PRINT "USE (F4) FOR ATTACK DECREASE
"                                          NB
·1055 PRINT "USE (F5) TO CHANGE WAVEFORM" KC
·1056 PRINT"USE (F7) TO CHANGE SOUND NUMB
ER"                                        BI
·1057 PRINT"USE (SPACEBAR) TO TEST SOUND" AH
·1058 PRINT"USE (X) TO EXIT"               OB
·1059 PRINT"[HOME][DOWN][DOWN][4" "]SOUND
 DEFINITION FOR VOICE #";V                 PD
·1060 PRINT"[HOME][10"[DOWN]"][10"[RIGHT]
"]";                                       PE
·1062 BASE=12*4096+256+V*4                 LJ
·1063 F=PEEK(BASE)+PEEK(BASE+1)*256        DH
·1064 D=PEEK(BASE+2):GOSUB 3000            FE
·1066 IF D>15 THEN D=15                    OH
·1067 IF S>2 THEN S=2                      KE
·1070 FOR X=1 TO (F/65535)*28              JK
·1080 PRINT "[RVSON] ";:NEXT X:PRINT"[RVS
OFF] ";                                    EK
·1090 PRINT"[DOWN]":PRINT"[10"[RIGHT]"]"; FO
·1100 FOR X=1 TO D*1.7                     PA
·1110 PRINT "[RVSON] ";:NEXT X:PRINT"[RVS
OFF]  ";                                   EC
·1120 PRINT"[DOWN]":PRINT"[10"[RIGHT]"]"; FO
·1130 IF S=0 THEN PRINT"TRIANGLE":S1=16    ND
·1140 IF S=1 THEN PRINT"SAWTOOTH":S1=32    AN
·1150 IF S=2 THEN PRINT"NOISE[5" "]":S1=1
28                                         MC
·1160 GET I$:IF I$="" THEN 1160            NG
·1170 IF I$="[F1]"THEN F=F+(200[UPARROW](
1.25+F/650000)):IF F>65535 THEN F=65535    IH
·1180 IF I$="[F2]"THEN F=F-(200[UPARROW](
1.25+F/650000)):IF F<0 THEN F=0            GK
·1190 IF I$="[F3]"THEN D=D+1:IF D>15THEN
D=D-1                                      EF
·1200 IF I$="[F4]"THEN D=D-1:IF D<0 THEN
D=D+1                                      MO
·1210 IF I$="[F5]" THEN GOSUB 2200         GB
·1220 IF I$="[F7]"THEN V=V+1:IF V>7 THEN
V=0                                        CE
·1225 IF I$="[F7]"THEN GOTO 1000           FP
·1230 IF I$=" " THEN POKE SID+4,S1:POKESI
D+4,S1+1                                   HF
·1235 IF I$="X" THEN 200                   IM
·1240 POKE SID,F-INT(F/256)*256            LO
·1250 POKE SID+1,INT(F/256)                AJ
·1260 POKE SID+6,D+240                     CG
·1270 POKE SID+4,S1                        BJ
·1280 POKE BASE,F-INT(F/256)*256           BB
·1290 POKE BASE+1,INT(F/256)               FI
·1300 POKE BASE+2,D:POKE BASE+3,S1         KM
·1500 GOTO 1059                            GE
·2000 REM INITIALIZE SID CHIP              JD
·2005 SID=54272:FI=128:SP=128              KD
·2010 FOR X=0 TO 28                        NM
·2020 POKE SID+X,0:NEXT X                  AE
·2030 POKE SID+21,7:POKE SID+23,7:POKE SI
D+22,FI                                    FA
·2040 POKE SID+24,31                       AF
·2050 RETURN                               IM
·2200 S=S+1:IF S>2 THEN S=0                AI
·2205 IF S=0 THEN S1=16                    AK
·2210 IF S=1 THEN S1=32                    BN
·2220 IF S=2 THEN S1=128                   DL
·2230 RETURN                               IM
·3000 S1=PEEK(BASE+3)                      DK
·3010 IF S1=16 THEN S=0:RETURN             NL
·3020 IF S1=32 THEN S=1:RETURN             OK
·3030 IF S1=128 THEN S=2:RETURN            BI
·3040 S1=16:S=0                            CK
·3050 RETURN                               IM
·5000 DATA 1,2,1,4,1,2,1,4,1,2,1,4,1,2,1,
4,1                                        GG
·5010 FOR X=49152+512 TO 49152+512+16      HG
·5020 READ A:POKE X,A:NEXT                 BA
·5500 PRINT "[CLEAR][13" "]PATTERN ENTRY" EK
·5510 PRINT"[4"[DOWN]"][17" "][10"1"][10"
2"][3"3"]"                                 IG
·5520 PRINT"[UP][8" "]123456789012345678901
23456789012"                               HP
·5525 PRINT"[UP][UP]"                      PN
·5530 FOR X=0  TO 7                        KA
·5550 PRINT "VOICE";X;"[LEFT]:"            BJ
·5560 NEXT X                               NK
·5570 PRINT "[40"[s *]"][UP]"              DB
·5580 GOSUB 5850                           GN
·5585 PRINT"[3"[DOWN]"]CURSOR KEYS MOVE C
URSOR"                                     IH
·5586 PRINT"USE (F7) TO EXIT"              KJ
·5587 PRINT"SPACEBAR TOGGLES MARKER"       MN
·5588 PRINT"USE (F3) TO CLEAR PATTERN"     AJ
·5600 V=0:X=0                              LL
·5610 LOC=40*7+V*40+8+X                    MD
·5620 POKE 1024+LOC,PEEK(1024+LOC)OR128:P
OKE55296+LOC,14                            EB
·5630 GET A$:IF A$="" THEN 5630            MC
·5640 IF A$=" " THEN IF PEEK(1024+LOC)=86
+128THEN POKE 1024+LOC,160:GOTO 5650       EA
·5645 IF A$=" " THEN IF PEEK(1024+LOC)=16
0THEN POKE 1024+LOC,86+128                 MK
·5650 IF A$="[RIGHT]" THEN X=X+1           AO
·5660 IF A$="[LEFT]" THEN X=X-1            FG
·5670 IF A$="[DOWN]" THEN V=V+1            OC
·5675 IF A$="[F3]" THEN FOR X=49152+512 T
O 49152+512+32:POKE X,0:NEXT X:GOTO 5500   JM
·5680 IF A$="[UP]" THEN V=V-1              GK
·5685 IF A$="[F7]" THEN GOTO 5750          JN
·5690 POKE 1024+LOC,PEEK(1024+LOC)AND127   CB
·5700 IF X<0 THEN X=0                      KM
·5710 IF X>31 THEN X=31                    AD
·5720 IF V<0 THEN V=0                      KI
·5730 IF V>7 THEN V=7                      ME
·5740 GOTO 5610                            GD
·5750 PRINT"WORKING.";                     HI
·5752 FOR X=0 TO 30                        MD
·5755 S=0                                  DM
```


www.commodore.ca

To enter Lightning Loader ...you must use our *Flankspeed* machine language entry program. Read the instructions for *Flankspeed* on page 122.

```
·5757 PRINT".";                                          EI
·5760 FOR V=0 TO 7                                       JO
·5770 LOC=40*7+V*40+8+X                                  MD
·5780 IF (PEEK(1024+LOC)AND127)=86 THEN S
 =S+2[UPARROW]V                                          AM
·5790 NEXT V                                             NI
·5800 POKE 49152+512+X,S                                 GL
·5810 NEXT X                                             NK
·5815 GOSUB 8000                                         FN
·5820 GOTO 200                                           BO
·5850 FOR X=0 TO 31                                      MC
·5860 FOR V=0 TO 7                                       JO
·5862 IF PEEK(49152+512+X)=0 THEN 5895                   FH
·5865 IF (PEEK(49152+512+X)AND(2[UPARROW]
 V))=0 THEN 5890                                         JD
·5870 LOC=40*7+V*40+8+X                                  MD
·5880 POKE 1024+LOC,86:POKE 55296+LOC,14                 HN
·5890 NEXT V                                             NI
·5895 NEXT X:RETURN                                      KH
·6000 SYS 49312:IF PEEK(197)=64 THEN 6000                JP
·6010 GOTO 200                                           BO
·8000 S=0:FOR V=0 TO 7                                   BN
·8010 IF (PEEK(40*7+V*40+39+1024)AND127)=
 86 THEN S=S+2[UPARROW]V                                 FN
·8020 NEXT V:POKE 49152+512+31,S:RETURN                  GO
·9999 GOTO 9999                                          HB
```

LIGHTNING LOADER
FROM PAGE 74

Starting address in hex: C900
Ending address in hex: CC00
SYS to start: 51456

```
C900: A2 05 BD 0B C9 95 73 CA 0E
C908: 10 F8 60 EA EA EA 4C 11 8F
C910: C9 84 B0 E6 7A D0 02 E6 2A
C918: 7B A0 00 B1 7A C9 51 F0 6C
C920: 05 A4 B0 4C 79 00 C8 B1 BA
C928: 7A C9 93 D0 F4 18 A5 7A FD
C930: 69 02 85 7A 90 02 E6 7B 90
C938: 20 D4 E1 A0 00 C4 B7 F0 1D
C940: 08 B1 BB 99 F0 CB C8 10 E4
C948: F4 A9 A0 99 F0 CB C8 C0 67
C950: 10 90 F8 20 0B CA 78 20 78
C958: C9 C9 A0 01 A2 08 A5 B9 97
C960: F0 06 AC 02 C8 AE 03 C8 49
C968: 84 AE 86 AF 20 93 C9 B0 FF
C970: 08 20 C9 C9 20 90 C9 90 37
C978: F8 58 A5 B9 D0 04 A6 AE 53
C980: A4 AF AD 00 C8 0D 01 C8 22
C988: D0 03 4C 9C E1 4C A1 E1 F6
C990: A2 02 2C A2 04 AD 00 C8 7E
C998: D0 19 AD 01 C8 F0 12 A0 9D
C9A0: 00 BD 00 C8 91 AE EC 01 55
C9A8: C8 F0 06 20 C2 C9 E8 D0 CE
C9B0: F0 38 60 A0 00 BD 00 C8 61
C9B8: 91 AE 20 C2 C9 E8 D0 F5 55
C9C0: 18 60 E6 AE D0 02 E6 AF 38
C9C8: 60 20 D8 C9 A0 00 20 D8 85
C9D0: C9 99 00 C8 C8 D0 F7 60 EE
C9D8: AD 12 D0 29 07 C9 03 90 F6
C9E0: F7 A9 27 8D 00 DD CD 00 E2
C9E8: DD D0 FB A9 07 8D 00 DD AF
C9F0: A2 04 EA CA 10 FC A2 03 FF
C9F8: AD 00 DD 0A 08 0A 26 B0 77
CA00: 28 26 B0 CA 10 F2 A5 B0 23
CA08: 49 FF 60 20 6D CA A9 49 FC
CA10: 20 A8 FF 20 AE FF A9 00 51
CA18: A0 CB 85 B0 84 B1 A9 00 9A
CA20: A0 03 85 B2 84 B3 A9 08 E5
CA28: 85 B4 20 83 CA A9 57 20 F1
CA30: A8 FF A5 B2 20 A8 FF A5 9F
CA38: B3 20 A8 FF A9 20 20 A8 47
CA40: FF A0 00 A2 1F B1 B0 20 25
CA48: A8 FF E6 B0 E6 B2 CA 10 FC
CA50: F4 20 AE FF C6 B4 D0 D2 33
CA58: 20 83 CA A9 45 20 A8 FF 7E
CA60: A9 70 20 A8 FF A9 03 20 10
CA68: A8 FF 4C AE FF 20 AE FF DA
CA70: A9 00 85 90 A5 BA 20 B1 62
CA78: FF A9 6F 20 93 FF A5 90 7B
CA80: D0 0E 60 20 6D CA A9 4D 0F
CA88: 20 A8 FF A9 2D 4C A8 FF 1D
CA90: A2 05 6C 00 03 C4 0D 0E 87
CA98: 4F 4E 2D 52 41 4D 20 0C 70
CAA0: 01 41 C4 0D 43 48 45 01 86
CAA8: 4B 53 55 4D 20 45 52 12 B3
CAB0: 4F D2 0D 46 49 4C 45 20 21
CAB8: 45 52 52 4F 20 0D 0D 0D 39
CAC0: 48 49 4C 4F 41 44 2E 03 A4
CAC8: 36 34 20 56 30 37 32 38 7B
CAD0: 38 32 29 29 43 29 20 31 4B
CAD8: 39 38 32 20 42 59 20 03 5B
CAE0: 4F 45 4D 5F 45 4F 53 05 0F
CAE8: 20 43 55 1B 49 4B 01 1B 6D
CAF0: 53 39 79 19 43 58 59 09 0E
CAF8: 51 5B 19 38 3D 20 35 20 A9
CB00: 4C 19 03 A9 E0 85 00 A5 1E
CB08: 00 30 FC C9 02 90 09 A2 3D
CB10: 00 8A 9D 00 06 E8 D0 FA F2
CB18: 60 78 A9 06 85 31 4C D1 75
CB20: F4 85 08 AD 00 18 4A 90 43
CB28: FA A9 08 8D 00 18 AD 00 28
CB30: 18 4A B0 FA A2 03 A9 00 8D
CB38: 06 08 2A 0A 06 08 2A 0A BC
CB40: 8D 00 18 CA 10 F0 A2 01 55
CB48: CA D0 FD EA EA A9 00 8D EE
CB50: 00 18 60 78 A5 3A 20 21 62
CB58: 03 A0 00 B9 00 06 20 21 FC
CB60: 03 C8 D0 F7 AD 01 06 85 2F
CB68: 07 AD 00 06 85 06 58 60 67
```


www.commodore.ca

THE TORONTO PET USERS GROUP INC.

TPUG

A membership in the world's largest Commodore users' group will provide you with:

- 10 issues of TPUG magazine
- Advice from experts like Jim Butterfield and Elizabeth Deal
- Access to a huge library of outstanding public domain software
- An invitation to attend the Annual TPUG conference, May 25 and 26 in Toronto, Canada, with two full days of seminars on all aspects of Commodore computing.

JOIN NOW!

I would like to join TPUG as an associate member at a cost of US$25.00.
Send your cheque, money order or credit card number to:

TPUG Inc.,
1912A Avenue Road,
Toronto, Ontario, Canada
M5M 4A1

Name _____

Address _____

_____

☐ Cheque ☐ Money Order ☐ Credit Card _____ Amount in U.S. funds $ _____

Credit Card # _____ Expiry date _____ Signature _____

Type of Computer _____ Disk Drive _____ Printer _____ Modem _____ Other _____

Reader Service No. 124

www.commodore.ca

```
CB70: 20 00 C1 A9 12 85 06 A9 43
CB78: 01 85 07 20 03 03 20 64 B0
CB80: 03 A0 07 84 09 A9 00 88 EA
CB88: 85 6F 84 70 20 A9 03 F0 30
CB90: 3D 18 A5 6F 69 20 85 6F 79
CB98: C6 09 10 F0 A5 06 D0 DB C1
CBA0: 20 0F 03 EE 02 06 4C DC F2
CBA8: 03 A0 05 A2 00 BD F0 03 A5
CBB0: C9 2A F0 11 C9 3F F0 07 A7
CBB8: D1 6F F0 03 A9 FF 60 C8 C0
CBC0: E8 E0 10 D0 E8 A0 02 B1 A8
CBC8: 6F C9 82 D0 EF 60 A0 03 49
CBD0: B1 6F 85 06 C8 B1 6F 85 EC
CBD8: 07 20 03 03 20 53 03 D0 4D
CBE0: F8 AD 00 1C 29 F7 8D 00 52
CBE8: 1C 60 00 B2 00 FA 00 BA CD
CBF0: 00 00 00 00 00 00 00 00 F0
CBF8: 00 00 00 00 00 00 00 20 19
CC00: 0E 0E
```

COMMODARES
PROGRAMMING CHALLENGES

Continued from page 110

```
4 REM SOLUTION BY BOB MARTIN
5 REM
10 POKE 88,24
20 POKE 89,121
30 POKE 90,232
40 POKE 91,7
50 POKE 95,0
60 POKE 96,4
70 SYS 41919
80 POKE53281,11:PRINT"+":POKE53281,12
90 Q=0
100 MM=30000
110 NE=1024
120 POKE NE+39-Q,PEEK(MM)
130 NE=NE+40 : MM=MM+1
140 IF NOT (NE>2023) THEN 120
150 Q=Q+1 : NE=1024 : MM=MM+15
160 IF NOT (Q>24) THEN 120
```

Bob uses a built-in ROM memory-move routine in lines 10 through 70. Locations 88 and 89 get the address of the destination and plus one, 90 and 91 get the source end plus one, and 95 and 96 get the address of the source start. The screen characters are copied row by row into a temporary buffer at address 30000 before they are put back onto the screen column by column.

Congratulations to the many other readers who sent solutions to these *Commodares*. Your letters, suggestions, questions, and solutions are always welcome. People with valid solutions who were not mentioned earlier this month include Bill Sherman (Garson, ONT), Michael Springer (Huntington Beach, CA), Jacqueline Callaway (Orange Beach, AL), Gerald Roberts (Green Cove Spgs., FL), G. A. Skaggs (Adelphi, MD), Michael Marron (Stony Brook, NY), Guillermo Gonzalez (Hialeah, FL), Troy Myers (Fort Ashby, WV), Paul M. Lalli (McAlester, OK), Wallace Leeker (Lemay, MO), Len Lindsay (Madison, WI), Terry Moss (Tuskegee Institute, AL), Emru Townsend (Pierrefords, QUE), David Hoffner (Brooklyn, NY), Paul Dawson (Waitsfield, VT), Dale Moose (Plattsburg, NY) and Derrell Harrison (Cabot, AR).

Here is one final program that might provide you with a chuckle. Type this one-liner from Mike Combs (Kansas City, MO) near the middle of the screen (around line 10).

```
1 ?"[HOME][RIGHT][UP][UP][INSERT]":POKE
218,136:RUN
```

Type a line of text on the second line of the screen. Then move to the bottom line of the screen and type RUN. The result is a variation on the *Screen Scramble* theme. It's up to you to figure out what is going on. See you next month. □

...CALLING ALL PROGRAMMERS...

Ahoy! is constantly in search of the best Commodore programs being written today. If you have a utility, game, or other type of program that fits that description, in BASIC or machine language, we'd like to see it.

Send a copy of your program on disk or tape, with documentation and a printout, to *Ahoy!*, 45 West 34th St.—Suite 407, New York, NY 10001. Include a self-addressed envelope with sufficient return postage affixed.

...COMING IN THE DECEMBER *AHOY!* (ON SALE NOVEMBER 15)...

THE IBM CONNECTION
ONE ELECTRONIC BRAIN MEETS ANOTHER!

GYPSY STARSHIP
LINK ML INTERRUPTS AND BASIC INTO A COMPLETE GAME

HIGH RESOLUTION GRAPHICS, PART II
ADD JOYSTICK ACTION TO BIT-MAPPED PROGRAMS

www.commodore.ca

Why settle for less when you can have Mo?

Mitey Mo turns your Commodore 64 into a telecommunications giant. It's the best-performing modem with upload/download.

Mitey Mo is being hailed as "the best price/performance communications package available." Its software has received the endorsement of the U.S. Commodore Users Group, which gives a money-back guarantee to members. It is truly the industry standard, and no wonder. It's the most user-friendly modem you can buy—it will take you online faster and easier than anything else.

Mitey Mo opens up a world of practical and exciting uses for your C-64. It lets you send and receive electronic mail, link up with community bulletin boards, play computer games with people in distant places, tap into library resources, and much more. All at your convenience.

Until Mitey Mo, Commodore's 1650 Automodem was the obvious choice when you went looking for a modem for your computer. Like Mitey Mo, it has "auto answer"—it receives data while unattended. And both modems are "auto dialers"—you dial right on the computer's keyboard. But that's about where the similarity ends.

Mitey Mo can dial up to 9 numbers sequentially. But suppose you dial a number and find it's busy. Mitey Mo has "auto redial"—it hangs up and redials immediately until it gets through. With the other modem you have to redial each time—and somebody with auto redialing can slip in ahead of you.

Mitey Mo is menu driven. It lists the things you can do on the screen. Select a number and you're on your way. Since Automodem isn't menu driven, you'll be hunting through the manual a lot.

With Mitey Mo, your computer's function keys are programmable—you can save yourself plenty of keystrokes. Not so with the other modem. And only Mitey Mo lets you store data to review or print it later.

Mitey Mo has just one switch, the Smart 64 software does the rest. With the other modem you'll have to remember to check three switches, otherwise you may be answering when you mean to be originating.

Mitey Mo is half the size of the other modem. The very latest technology allows miniaturization and increased reliability, as well. Mitey Mo is so reliable, we gave it a full three-year warranty. The other modem gives 90 days, then you're on your own.

Not only will you find Mitey Mo mighty useful, you'll find it mighty reasonably priced. When you buy it, you'll get $15 of CompuServe access time and 2 hours of PlayNet free, as well. See your dealer or call us directly to order your Mitey Mo.

MODEM FEATURES	MITEY MO	COMMODORE AUTOMODEM
Auto Dial/Answer	YES	YES
Auto Redial	YES	NO
Smart 64 Software	YES	NO
Function Keys Programmable	YES	NO
Upload/Download Text & X-Modem	YES	NO
VT-52/VT-100 Emulation	YES	NO
Menu Driven	YES	NO
28K Software Buffer	YES	NO
Easy-to-Use Manual	YES	NO
Bell 103 Compatible	YES	YES
Multiple Baud Rates	YES	YES
Cable Included	YES	YES
Single Switch Operation	YES	NO
Warranty	3 years	90 days

Some mighty interesting features—ours and theirs. Yours to decide.

CDI
Computer Devices Int'l
1345-A2 Doolittle Drive
San Leandro, CA 94577
(415) 633-1899

Reader Service No. 138

www.commodore.ca

BEACH-HEAD II

FEATURES "VOICE SIMULATION" AND TRUE "HEAD TO HEAD" TWO PLAYER ACTION

THE DICTATOR STRIKES BACK

By Roger and Bruce Carver

The war time drama of the award winning game Beach-Head continues with Beach-Head II, "The Dictator Strikes Back."

Beach-Head II pits allied forces against the cruel dictator, who escaped the destruction of the fortress and fled into the tropical forest with portions of his army and prisoners captured during the land battle. The player must locate and penetrate the heavily armed sanctuary, capture the dictator, and free the prisoners.

$39.95

AVAILABLE ON DISK FOR COMMODORE 64/128, APPLE II+/E/C, ATARI 48K

TRY THESE OTHER FINE ACCESS PRODUCTS

MACH 5™

NEW

CARTRIDGE FAST LOADER

- Load programs up to 500% faster.
- Full help menu at the touch of a key.
- Eliminate 1541 disk drive rattle.
- Print screen (screen dump to printer).
- Restore basic program (old).
- List disk directory without erasing program in memory.
- Pause or exit from directory listing.
- Execute short hand load commands from directory.
- Auto run *any* basic program.
- Open and close printer channels with a single key.
- Disable and re-enable features without turning off computer.
- Compatible with all popular printers.
- Send dos command (@).
- Load virtually 95% of all software, even copy protected games.
- Completely transparent. Does not disturb ram.

$34.95

Compatible with your Commodore 64/128 and 1541, Indus GT, Enhancer 2000 or Tech 16.

BEACH-HEAD™

"The graphics are remarkable—it's almost like watching a John Wayne Movie."

—Computer Games

$34.95

Commodore 64/128, Atari 48K

RAID OVER MOSCOW™

"Raid Over Moscow is action gaming at its finest—realistic graphics, superb sound effects and highly challenging tests of skill."

—Home Computing & Video News

$39.95

Commodore 64/128

ACCESS
Software Incorporated

VISIT YOUR FAVORITE RETAILER . . .

For mail orders, enclose check or money order, plus $2.00 shipping and handling, and specify machine version desired. Order by phone on Visa, MasterCharge or C.O.D. by calling 801/973-0123.

ACCESS SOFTWARE, INC. 925 East 900 So. Salt Lake City, UT 84105

Reader Service No. 286

www.commodore.ca