lar or an annuity. These functions can be used within your formulas.

SwiftCalc's documentation includes extensive explanations of how to use these formulas and functions to their maximum and numerous examples which are included in the sample spreadsheets on the disk.

Like *SwiftCalc, Data Manager* allows you to use calculations with the data that you enter: not as complex as those in *SwiftCalc*, but perfectly adequate for a program that is intended to organize data rather than manipulate it.

Data Manager's manual gives a great set of instructions on how to set up your database. Once it is set up, the features to enter and organize the information are simple and effective. Special programs to sort and search are included, making it easy to isolate parts of your database to print or copy out to another file. The report writer (which is reached through the pull-down menus) makes it easy to print reports or labels.

The initialization procedure for setting up your database is simple and uses onscreen layout. Your layout can contain multiple pages and titles be printed or omitted. By properly setting up the database initially, anyone can sit down and start entering information. A function is even included to allow you to add new fields to the database after it's been created and data has been entered. Although new fields can only be added at the end of the existing form, this is vastly superior to having to redefine the entire form and reenter all of the previous information.

Written especially for the 128, all three programs reviewed make good use of the added memory and expanded keyboard.

Timeworks, 444 Lake Cook Road, Deerfield, IL 60015 (phone: 312-948-9200). *—Cheryl Peterson*

PLAYNET

C-64

Starter Kit $39.95 (see below)

The people who run PlayNET like to refer to their system as "the network that has people talking." Those of you who have not tried the system

A PlayNET membership kit is yours free with a one-year subscription to Ahoy!

For $19.95 you'll receive 12 issues of *Ahoy!*, PlayNET membership kit ($19.95 value), and the following services available only through PlayNET:

- The *Ahoy!* Port of Call, a weekly teleconference with *Ahoy!*'s writers and editors
- Downloading of *Ahoy!* programs at a nominal fee

- Discount prices on *Ahoy!* disks and back issues
- Multiuser access to *Ahoy!* Bulletin Board

Don't have a modem? PlayNET will give you one free! Take advantage of our $19.95 *Ahoy!*/PlayNET offer, and agree to *subscribe to PlayNET for three months*, and you'll receive a free auto dial/auto answer modem right now! The famous make, touch tone or rotary capable modem carries a 90 day warranty.

☐ Send me 12 issues of *Ahoy!* Enclosed is my check or money order, payable to Ion International Inc., for $19.95 ($26.95 Canada and elsewhere). Also, sign me up to PlayNET* for one month, and send me the membership kit ($19.95 value) free of charge.

☐ In addition, I agree to subscribe to PlayNET* for three months. Send me my free auto dial/ auto answer modem now.

*As a PlayNET member I will be billed on my credit card $2.75 per on-line hour and $8.00 per month account maintenance fee. I can cancel my membership at any time by writing by PlayNET, but if I receive a modem and cancel before 90 days I will be billed $44.00.

NAME_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

TELEPHONE_____

Circle One: MC/VISA Card #_____

Expiration Date_____

Signature_____ Date_____

SEND TO: *Ahoy!* 45 West 34th St., New York, NY 10001

www.commodore.ca

HOW TO GET OVER $2000 WORTH OF NEW CAPABILITIES FOR YOUR COMMODORE 64™

FOR $599*

The Spartan™ is the Apple™ II+ emulator for your Commodore 64™ that will open up a whole new world of hardware and software for you! Imagine adding these features to your Commodore 64™ for the Spartan™ price of $599: ☐ Apple™ II+ hardware and software capabilities ☐ 64K RAM expansion ☐ four software selectable Commodore 64™ cartridge slots ☐ non-dedicated 8-bit parallel port ☐ standard audio cassette deck capabilities for your C-64™. The suggested retail value of comparable products offering only these capabilities is over $2200.00* — but the Spartan™ gives you much, much more! By building on your investment in your Commodore 64™— an excellent introductory computer — you create a whole new system with both C-64™ and Apple™ II+ capabilities. There is a whole other world out there! The huge selection of Apple™ II+ hardware and software is now yours to explore! Call toll free for the Spartan™ dealer nearest you.

*All prices quoted are in U.S. funds, freight and taxes not included. Value of components equivalent to the Spartan™ system are quoted from Apple™ II+ CPU and Apple™ II+ single disk drive 1983 list prices, and from current suggested list prices and component specifications of other peripheral manufacturers. Commodore 64™ and Commodore logo are trademarks of Commodore Electronics Ltd. and/or Commodore Business Machines, Inc. Apple™ II+ is a trademark of Apple Computer Inc. Spartan™ is a trademark of Mimic Systems Inc. and has no association with Commodore Electronics or Apple Computer Inc. The Spartan™ is manufactured by Mimic Systems Inc. under license granted by ATG Electronics Inc. of Victoria, B.C. Canada.

MIMIC

FOR INFORMATION WRITE:
MIMIC SYSTEMS INC.
1112 FORT ST., FL. 6S
VICTORIA, B.C.
CANADA V8V 4V2

TO ORDER CALL:
1-800-MODULAR
(663-8527)

Reader Service No. 109

www.commodore.ca

might dismiss that slogan as typical marketing hype. If so, you would be missing out on one of the best values around for Commodore users. PlayNET really does have people talking! All kinds of people, from all around the country.

What are they talking about? Just about any topic you can imagine, from accounting tips to zucchini recipes and anything in between. They do this through the use of a function called "People, Games & More," which gives users from the smallest towns to the largest cities a place to meet and talk with each other and even play games together. The "People, Games & More" section of PlayNET is divided into electronic "rooms" that can each hold a dozen users at one time. There are several standard public rooms that are open to all users, called reception rooms. Although they are not only public rooms, these reception rooms provide a starting point for your electronic journey across the nation. By switching from one room to another you might discover a conversation about children's books or a weekly meeting of computer bulletin board system operators from major cities across the country. And if you don't find the conversation you're looking for you can start your own room simply by going to it. You can even start a private room where only those people you've invited can join you.

Another nice feature of the PlayNET system is the ability to send online messages or electronic mail to any other person on the system. Messages are delivered to the user in seconds wherever they might be on the system. If that person is not signed on when you want to send your message, you can leave electronic mail that will be waiting for them the next time they log on. And PlayNET is more than just talk. It's also games: games that you play against a real human, not a computer. Strategy games such as Chess, Checkers, Backgammon, and Capture the Flag; word games like Hangman; and card games such as Contract Bridge. In all there are 14 different games on the PlayNET system.

PlayNET also supports a wide variety of computer bulletin boards. Here, PlayNET subscribers can leave messages and trade information with other users about such topics as current events, arts and entertainment, hobbies and sports, and many others. Classified advertisements are also available.

Another interesting area is the PlayNET Shopping Center. Here PlayNET users may purchase a number of different items such as PlayNET T-shirts and key rings and various books and magazines. This area is currently being expanded and will soon be offering a much wider range of products and services.

The Software Delivery Service of the PlayNET system is an online area where subscribers can freely trade programs that they have written or public domain software obtained from other sources. There are three different ways of using the PlayNET Software Delivery Service:

1) The Personal File Transfer area is where one subscriber uploads a file to PlayNET. The file is then held for two days. During that two-day period the file is available to any other subscriber who knows the filename and the name of the person who uploaded it. There is a small fee for downloading one of these files.

2) The Software Trading Post is where members may buy or sell software that they, or other users, have written. When a program is uploaded to this area, it is made available to other users at the price specified. If another user purchases the program, the price is deducted from that user's account and added to yours.

3) The Public Domain Software Library is where any PlayNET user may upload or download public domain software. There is no fee charged for uploading, but a downloaded program will cost you 50 cents.

Each PlayNET Master Account has the option of creating up to ten additional sub-accounts, the first five of which are free. There is a small fee for each sub-account after that. Sub-accounts allow each family member to have his own individual account number and password, as

READER SERVICE INDEX

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

www.commodore.ca

well as individual onscreen name. Sub-account charges are automatically added to the master account for billing at the end of each month, and each sub-account may have a different credit limit attached to it. This allows the holder of the master account to create separate accounts for each person in the family without fear of running over budget. Just give the kids a monthly PlayNET allowance and turn them loose. Complete and comprehensive billing information is always available online, and you can even sign up a friend while online. For individuals without children, the sub-accounts allow multiple personalities online and PlayNET makes it simple to switch accounts without the need to logoff and sign back on.

In general, I have found PlayNET users to be just about the friendliest group of people around. Dropping into a reception room is sure to bring a chorus of hellos from the occupants, even if they are complete strangers. There is an overwhelming family feeling at work on this system that makes many other networks seem cold and impersonal by comparison. PlayNET supports and encourages this feeling by sponsoring many monthly events and activities. Special Interest Groups (SIGS) abound on PlayNET, with such subjects as life-sharing and Physical disabilities appearing alongside poetry readings and comic books.

There is certainly something for everyone on this system. In fact, I have only found two aspects of the PlayNET system disagreeable. The first is that it takes so long for the software to load when going from one area of the system to another. This is not a fault of PlayNET's but rather due to the slowness of the 1541 disk drive. Although the PlayNET software is not copyprotected, most fast loader programs will not work with it. The only fast loader that will function reliably is 1541 Flash from Skyles Electric Works. PlayNET is aware of this problem, though, and have told me that by the time you are reading this review they will have incorporated their own fast loader into the software. My second complaint with PlayNET is that they are only open from 6 p.m. local time to 7 a.m. eastern time, and 24 hours on weekends.

The suggested retail price of the PlayNET package is $39.95, and connection charges are $2.75 per hour with an $8.00 monthly service charge. You may sign up by calling 1-800-PLAYNET and tell them Captain B sent you. Please feel free to drop me a line any time you're on the system.

PlayNET Inc., 200 Jordan Road, Troy, NY 12180 (phone: 518-283-8682 or 1-800-PLAYNET).

—B.W. Behling

1541 DISK DRIVE ALIGNMENT PROGRAM, VERSION 2.0

CSM Software, Inc.
Commodore 64 and 1541
Disk; $44.95

Alignment, or more precisely the lack of it, is an affliction which many 1541 disk drive users are all too familiar with. Disk drive misalignment results from a combination of factors. The early versions of the 1541 were more apt to suffer from misalignment, due to a design deficiency in a critical part of the hardware. This was further aggravated by the "head bump" error checking used by early software protection schemes. Interestingly enough, we have found that Commodore was one of the worst offenders with regard to this type of primitive copy protection.

Although the 1541's hardware has been improved and copy protection no longer needs to perform a head bump, the disk drive may still need to be aligned from time to time. Some normal disk operations, such as formatting a disk, will cause a head bump. Many of the old protection schemes are also still around. Even with the most solidly constructed 1541, long hours of use will eventually necessitate adjustment of its mechanism.

The symptoms of disk drive misalignment are easily recognized. In its most severe form, all commercial software (which was presumably manufactured on properly aligned hardware) will refuse to load. If the drive is allowed to attain this sorry condition, other problems will manifest themselves. The most aggravating of these is the inability to read disks which were previously formatted on the same disk drive. The irony of this problem is that once the offending drive is properly aligned, your most recently formatted disks will become unreadable. The only solution is to copy these disks to a second aligned disk drive before realigning the first.

Fortunately, there is no need to allow matters to reach this deplorable state. The early symptoms of misalignment can be easily recognized by the alert user. At first a slight increase in the loading times of commercial software may become evident. This will shortly be augmented by an intermittent flickering of the disk drive's error light. In the advanced stages, the drive will seem to detect numerous disk errors. The resulting proliferation of "head bumps" serves to accelerate the process to the point where commercial software will not load at all. Very often many of these symptoms are mistakenly assumed to be the fault of the program disk rather than a problem with the disk drive. In the most extreme case the stepper motor cam will slip an entire track, effectively bringing the drive back into alignment. Unfortunately, this condition is only temporary.

For most users, disk drive misalignment will mean a trip to a qualified service technician. Use of the electronic instruments required is beyond the knowledge of the average user. The *1541 Disk Drive Alignment Program* by T. N. Simstad aims to eliminate the need for any electronic expertise from the alignment procedure, at the cost of a single qualified service call. The process is reduced to the execution of a purely mechanical series of operations. Of course, the ability to load the alignment program and use the computer is also required.

What is required of the user is a fairly good mechanical aptitude. The accompanying manual does provide step by step instructions, as well as prolix descriptions of the events in question. Unfortunately, the accompanying sketches provide only the es-

www.commodore.ca

sential information as referred to by the text. Your own imagination will be required to bridge the gap from the physical presence of your disassembled drive to the primitive drawings in the manual – an exercise not made any easier by the three physical and electronic variations which have come about as the 1541 has evolved.

In addition to the manual, the package includes two disks: the alignment software and a precisely formatted alignment disk. Neither of these disks can be copied, although one set of backup copies is available for $15. The first disk is protected by some rather sophisticated copy protection schemes. Of course, copying the second disk would only negate the value of the original's precision.

The directory of the program disk may be viewed, but not LOADed and LISTed, using the DOS wedge on the 1541 test demo disk. Doing so reveals a copyright notice, some apparently humorous comments, and a tongue-in-cheek challenge to copy the disk. We suppose this goes hand in hand with T. N. Simstad's and CSM's other products, in particular *The Program Protection Manual For the C-64,* Volumes I and II. These books include detailed discussions on the various forms of hardware and software protection used by software manufacturers for the C-64. We can only conclude that the protection on this disk must be some form of final exam for these volumes. However, instructions for the grading of your results were not included.

Operation of the program is straightforward, although loading it may present a problem. The complex copy protection and the expected sorry state of your disk drive may conspire against the successful booting of the package. In the worst case a second disk drive may have to be called into service. This may require some swapping of disk drives, as the program will only boot from device eight. However, once loaded, the alignment may be performed on disk drives with any legitimate device number. You may also have to disconnect any other serial bus devices, as the copy protection scheme is

GUARANTEED SOFTWARE

NEW 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

VIZAWRITE CLASSIC for C128

This is the new word processor from Vizastar's author, Kelvin 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.** 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. 131

www.commodore.ca

sensitive to these otherwise innocuous intruders.

A number one Phillips head screwdriver, in very good condition, is also required. We have found the stepper motor hold down screws to be very tight on most 1541 disk drives. Using a worn screwdriver will chew up the head, requiring heroic measures to undo the damage.

The alignment program provides several menu selections. Speed accuracy is important to disk drive operation. The actual drive speed is displayed and continuously updated as a percentage of the proper speed. Adjustment can be easily made in real time. You will need a formatted "scratch" (i.e., no valuable data) disk for this purpose. The main menu has a format disk option which interestingly enough refused to format an unformatted disk. On the other hand, a preformatted disk formatted just fine. This is clearly a utility of futility. Just remember to format a disk before you start. The two-minute boot time of the alignment program is not to be taken lightly.

Two alignment checks are provided. The coarse adjustment sets the read/write head over track one. The head stepper motor must then be set to center the head between the alternate positions of noticeable disk drive error. This is the point where the disk error light just begins to flicker. The proper position is then determined by rocking the stepper motor between these two points.

The fine adjustment process is far more critical and time consuming. A total of nine tracks are cyclically scanned by the program. Although the readability of each track is noted, it is the timing of the entire cycle which is important. The objective is to minimize the time it takes to run a cycle check. Since each cycle takes at least 30 seconds, this iterative process can become lengthy, in particular towards the end when the setting of the stepper motor is somewhat touchy.

In addition to the head alignment, the track one stop must be properly set. Failure to do so will cause the drive to have difficulties in formatting a disk. The symptom of an improperly set stop is the inability of the drive to read track one on a disk which it had just formatted. The setting of the track one stop is purely mechanical. The program positions the head over track one and requires the stop to be adjusted within .006 to .01 inch of the stepper motor cam. The hard part is the lack of any refined adjustment mechanism on the disk drive itself. Most 1541's have a single screw holding down the track one stop. Tightening this screw invariably upsets the stop's position. As a result, the whole procedure becomes somewhat hit or miss. Fortunately, the track one stop rarely needs adjustment.

As a last resort, the manual addresses the underlying cause of disk drive misalignment. The earliest 1541's are the ones most likely to suffer from this fault. Once the drive loses alignment there is an increased tendency for misalignment to reoccur. The manual recommends what we believe to be the only truly effective cure, pinning the stepper motor pulley to its drive shaft. Unfortunately, the precise drilling of a hole through this miniature shaft requires a rather specialized set of tools and skills.

Overall the *1541 Disk Drive Alignment Program* offers a viable, purely mechanical alternative to a fully instrumented disk drive alignment. The program also offers a fast and effective means of periodically checking the disk drive's speed and alignment. However, a full alignment using the CSM method will take longer than a "traditional" calibration with proper test instruments. A description of at least one instrumental alignment should have been included as an option for the qualified user.

CSM Software, Inc., P.O. Box 563, Crown Point, IN 46307 (phone: 219-663-4335). —*Morton Kevelson*

FONT Magick

Beautiful fonts designed by a professional calligrapher especially for C-64 screen display. Load them directly, no word processor needed. Use them for games, graphics, or just for programming variety.
("Programmer" makes reading the screen *much* easier.)
PLUS a fast, easy-to-use font designer so you can create your own.
PLUS "illuminated" Gothic initials—designed as sprites for easy use.
(Have the best text adventure games in town!)

Disk 1 ..... **$16.95**
☐ Fast, easy-to-use font designer, + Gothic, Programmer fonts + "illuminated" Gothic initials

Disk 2 ..... **$16.95**
☐ 9 fonts—Celtic, Data, Italic, Cursive, Broadway, Bold, Upside-Down, Mirror, Stencil

Send check or money order to: **Magick Enterprises**, 1742 Second Avenue, Suite 134, New York N.Y. 10128.
Add $3.00 per order shipping & handling. N.Y. residents add sales tax.

• Fonts • Fonts • Fonts • Fonts • Fonts •

ITALIC/1 italic/2
Gothic initials
CURSIVE/1 Cursive/2
DATA/1 Data/2
GOTHIC/1 Gothic/2
BROADWAY/1 Broadway/2
BOLD/1 Bold/2
PROGRAMMER/1 Programmer/2
STENCIL/1 Stencil/2
CELTIC/1 Celtic/2

Reader Service No. 134

SUPERFORTH 64

Parsec Research
Commodore 64
Disk; $99.00

SUPERFORTH 64 is my fifth. My fifth review of a version of the FORTH language, that is. Writing reviews is usually like Christmas—tear off the wrapping and play with a new

www.commodore.ca

toy. I have to admit that my heart dropped when I realized that writing this review was going to seem like déjà vu.

You see, FORTH as a generic computer language has some pretty well-recognized standards. True, there are a couple of variations on the theme, but FORTHs are typically much more similar to one another than are, say, BASICs. I didn't initially see how reviewing *SUPERFORTH 64* would involve much more than checking it against the standards and then trying to find an interesting way to tell you how different FORTH is from other computer languages, in both style and power.

Parsec Research had a few surprises for me, however, and I ended up having balls of fun and getting back into FORTH more than I had been in years. (Yes, I used to and still occasionally do program in FORTH. It is much faster than BASIC and it reinforces some good programming habits.)

Programming in FORTH forces you to structure your thinking and your code. Imagine a BASIC program that is totally modular, with the modules loosely strung together. The main body of that BASIC program might look like:

```
10 GOSUB 1000
20 GOSUB 2000
30 GOSUB 3000
40 GOSUB 4000
```

Each line in the main body might, in turn, direct you to another series of modules until you finally reach small blocks of free-standing code. Those blocks are FORTH words. Words in FORTH are compiled into a collection called the dictionary. A word in the dictionary may be nothing more than a series of other words linked together.

FORTH code is organized into physical and logical screens, essentially computer screenfuls of code. Source code screens are loaded into the system and compiled before they can be used. Once compiled, a source word can be called interactively or used in the definition of a new word. *SUPERFORTH 64* includes a decompiler, a trace facility, and a non-destructive stack dump, all to ease your debugging efforts.

FORTH's stand-alone words force your programming into a linear style and make debugging oh-so-easy. But what's special about *SUPERFORTH 64*, you ask? Well, it contains all the words required by the FORTH-79 standard and a bunch of those defined by the FIG (FORTH Interest Group) FORTH standard. It is actually a superset of the MVP-FORTH system. (I told you it follows the standards—as do all versions of FORTH.)

But *SUPERFORTH 64* goes beyond the other FORTHs I've used by giving you lots more words to start with. The predefined word set includes (bear with me here!): 15 editor, 13 source screen file mode, 8 byte/bit manipulation, 26 I/O, 11 Kernal interface, 36 utility (including backup), 50 graphics, 33 Turtle graphics, 23 sound, 11 music editor, 20 string extension, 6 interrupt, 4 display, 4 high RAM access, 2 array, 9 floating point, and 6 trig words. (Total 262.)

Furthermore, *SUPERFORTH 64* gives you an extension package for floating point math, with support for either decimal or scientific notation form. And it gives you words to manipulate two- or multi-dimensional matrixes. And an algebraic expression evaluator, so you do not have to work in FORTH's standard Reverse Polish Notation (where 2 + 2 is written 2 2 +). And an RS-232 word set. And a printer/plotter word set, for the 1525 and 1520. And even a couple of Koala pad utility words.

This is like buying BASIC and receiving several disks full of powerful subroutines and utilities thrown into the bargain. *SUPERFORTH 64* does, by the way, come on four disks—actually, on both sides of each of two disks. With all the extension packages and the source code, there's a lot to learn here. Although FORTH is about as different from other computer languages as it can be, this *SUPERFORTH 64* package is just waiting for you to jump in and use its power.

If all that is not enough, you even get the chance to dabble in artificial

The Universal MacInker(s)™ are here

Re-ink any Fabric ribbon automatically for less than

5¢

Now one **Universal Cartridge MacInker (UC)** re-inks all fabric cartridges and one **Universal Spool MacInker (US)** re-inks all spools. We have **MacInker(s)** dedicated to specialized cartridges, zip pack, harmonica etc. Over 1000 printer brands supported. Use your **MacInker** to re-ink your dry, fabric cartridges (for less than 5 cents in ink) and watch the improvement in print-out quality. Our new, residue-less, lubricated, dot matrix ink yields a darker print than most new ribbons. Or get any of our basic ink colors: brown, blue, red, green, yellow, purple and use **MacInker** to **create and/or Re-ink your own colored cartridges.** We have uninked or colored cartridges for the popular printers and ribbon re-loads for any printer. Operation is extremely simple & automatic with new, twin drive electric motor that supports CW and CCW rotating cartridges. A good quality fabric ribbon of average length can be re-inked almost indefinitely. In our tests one re-inked Epson* 80 ribbon has outlived the estimated life of the print-head!! We receive consistent & similar feedback from our customers. As of August 85 we have over 40,000 MAC INKER(s) in the field, in 5 continents (220 V motors available). **MacInker (UC) is $60.00.** Cartridge drivers are **$8.50/ea.** We still have our first generation, dedicated **MacInker(s)** for most popular printers. Prices start at **$54.95** with most units below $60.00. **MacInker** has been reviewed, approved and flattered in most magazines and even in the NEW YORK TIMES and the CHICAGO SUN TIMES.

Universal Cartridge Mac Inker

Computer Friends

6415 S.W. Canyon Ct.
Portland, OR 97221
(503) 297-2321

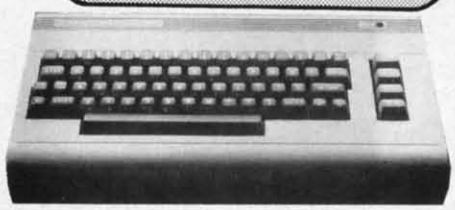
Order toll free 1-800-547-3303
or ask for free detailed brochure.
Dealers inquiries welcome.

*EPSON is a trademark of EPSON CORP.

Reader Service No. 135

www.commodore.ca

NEW
FOR COMMODORE 64 / 128
HOME INFORMATION MANAGEMENT SYSTEM
INTEGRATED HOME DATA MANAGER with WINDOWS, ON SCREEN CALCULATOR, and CALENDAR.
HOME CHECKBOOK maintains multiple accounts for quick and easy analysis.
HOME ADDRESSBOOK up to 200 records of mailing data with search and label options.
HOME INVENTORY provides valuable property information recall and analysis.
AVAILABLE ON DISK ONLY... $29.95
CHECK/MONEY ORDER + $2.00 SHIPPING.
NEW YORK RESIDENTS ADD SALES TAX.
VMC SOFTWARE
P.O. BOX 326
CAMBRIA HTS.N.Y.11411



Reader Service No. 112

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. 113

intelligence. A program called *EXPERT-2*, written in FORTH, is provided as an inference machine. *EXPERT-2* is primarily a learning tool that allows you to compile expert rule programs and to perform logical inferences on these rules.

Your *EXPERT-2* programs can have two components—a set of If-And-Then rules to manipulate truth values and a set of analytical subroutines. Playing with *EXPERT-2* can give you some good insights into human reasoning that is based on recognizing and sorting patterns to form the types of "rules" we each use daily. Eight example programs are included on the disk to show you how to use rules to do such things as predict the weather or analyze a digital circuit.

The only language you are likely to have used that is faster than FORTH is assembly language. If you are familiar with assembly language, *SUPERFORTH 64* will accommodate you. It contains an integrated assembler that will allow you to rewrite time-critical FORTH words to run in machine language. FORTH assembly code looks much like 6502 assembly code, except that FORTH's structured approach is maintained so 6502 branching commands are not included. If you have written your own assembly language routines, *SUPERFORTH 64* will allow you to use them after going through a simple conversion process.

SUPERFORTH 64 includes 500 pages of manual and tutorial materials just to cover the basics. The printed materials specifically do not cover in detail FORTH programming or artificial intelligence. Plan to buy one of the standard texts on these topics. Parsec does reference many good books, including all the standard FORTH programming and reference guides.

To help you get started, and assuming that most of us are familiar with BASIC, Parsec has included a BASIC to FORTH command translation table. It won't help you understand the structural differences between the languages, but it will make the vocabulary meaningful. And right there on page one of the manual, Parsec lists its address and phone and invites us to report problems, ask questions, and give user feedback.

Parsec Research, Drawer 1766, Fremont, CA 94538 (phone: 415-651-3160). *—Richard Herring*

BLUE CHIP D12/10 DAISY WHEEL PRINTER

Blue Chip Electronics, Inc.
Price: $249.00

The Blue Chip D12/10 is a low-cost daisy wheel printer intended for home applications. It cannot be said that it is the best performing printer ever released, but it has to be looked at from a home market point of view.

I found the D12/10 virtually identical in performance to the Brother HR-15. This is no surprise, for the D12/10 is manufactured by Brother. As a matter of fact, the rear of the printer is stamped Brother HR-10.

Print speed is a lethargic 12 characters per second. This is slow, but bearable. Additionally, there is a 2K buffer that will free your computer while the printer is busy.

The print qualify is in line with other printers of this type. It's clear and aligned well, making it quite acceptable for most home applications. Any of the standard Brother daisy wheels will work with the printer, giving the user a wide variety of fonts to choose from.

The D12/10 has the ability to perform an assortment of tricks. These include automatic underlining, strikeout, shadow print, and double strike. It can also move the carriage in 1/120" increments for precise alignment of text.

Interfacing is achieved through a standard Centronics parallel port. In order to connect it to your Commodore you must purchase the proper interface. As this is not a dot matrix printer, and thus incapable of reproducing onscreen graphics, the least expensive, non-graphic interface should suffice.

The Blue Chip D12/10 isn't going to win any awards, but for home use should be more than adequate.

Blue Chip Electronics, Inc., 2 West Alameda Drive, Tempe, AZ 85282 (phone: 602-829-7217).

—David Barron

www.commodore.ca

MICROSIM

A Flight Simulator for the C-64

By Tim Gerchmez

I was inspired to write *Microsim* when I purchased a commercial flight simulator package for the C-64. I had never before realized just how exciting it was to fly—the feelings of freedom and power are fantastic. This flight simulator is not as realistic as others available commercially. It is written in BASIC, which means that several things had to be sacrificed. *Microsim* is meant for entertainment purposes only: it is not intended to simulate any particular real aircraft. Please do not assume that because something works a particular way in *Microsim*, that's the way it works in real flight (though that may be the case). Also, please realize that *Microsim* is *not* a teaching tool—if you want to learn to fly an aircraft, be sure you take real flight lessons with a certified flight instructor.

To use *Microsim*, type it in and SAVE it. When you first RUN the program, the instruments will zero themselves out, and the cabin of the plane will pressurize. Following pressurization, you will hear two bumps signifying that the plane has landed. From here on *you* are in control.

Microsim uses keyboard input to control its instruments. First of all, let's go over these controls. The plus and minus keys control the engine RPM's (indicated by the ERPM indicator). Pressing plus speeds up the engine and minus slows it down. The minus key also acts as a brake for the aircraft when on the ground.

The greater than and less than keys (> and <) control the rudder, which steers the aircraft. You can use these keys either shifted or unshifted. (will bank the aircraft left, and) will bank it right. The HEADING indicator shows the current compass heading of the aircraft, and the BANK INDICATOR (indicated by BI> on the screen) shows which way the aircrat is banking. Imagine the BANK INDICATOR to be a view of the aircraft from the rear.

The U and D keys move the aircraft's ELEVATORS up or down, respectively. When the ELEV. indicator reads a positive number, the elevator is up from center. This tends to pull the nose of the aircraft up. When the indicator reads below zero, the aircraft will tend to pitch downward. The elevator can therefore be used to control the plane's altitude.

Pressing R or L will Raise or Lower the plane's FLAPS. The flaps work along with the elevator to help control the plane's altitude. In *Microsim,* the flaps should be down when taking off, and up when landing.

Press H to "hear" ATIS (Automatic Terminal Information Service). ATIS will display the current temperature, barometric pressure, and visibility conditions. It will also display the total number of nautical miles your aircraft has traveled since you booted the simulator (or since your last crash). This is indicated on the screen as TNMT (Total Nautical Miles Traveled).

Pressing the X key will refuel the aircraft. You have two fuel tanks available, the "green" tank and the "red" tank. Both tanks will be refueled with this command. This keypress works on the ground only (ever heard of a floating gas station?).

The number keys 1-4 control the indicators labeled 1-4 in the lower right corner of the instrument panel. These instruments will be covered shortly.

Pressing the 0 key will toggle between day and night flying. The program starts out in day flying mode. In *Microsim*, the only difference between flying in the day and flying at night is the "color of the air."

Finally, pressing the O key will cause the simulator to take an automatic demonstration flight. This command will be covered further in a while.

KEY CONTROL	QUICK REFERENCE
+ −	= Throttle controls
> <	= Rudder controls
U D	= Elevator controls
R L	= Raise/Lower flaps
H	= Hear Terminal Information
X	= Refuel aircraft
O	= Demo mode
0	= Toggle day/night flying situation
1	= Raise/lower landing gear
2	= Set fuel mixture
3	= Carb heat on/off
4	= Select fuel tank
5	= Lighting on/off
6	= Air conditioning on/off
7	= Cabin heating on/off

Now let's discuss some of the instrumentation not already covered. The TIME display shows a real time clock that is reset to zero each time you take off. Thus it indicates total time in the air. The GROUNDSPEED indicator measures the speed of your aircraft while on the ground. The FUEL and OIL TEMP. gauges are pretty much self-explanatory. The light with the P under it turns red when the cabin is pressurizing. The STALL WARNING LIGHT turns on when the plane drops to within 5 MPH of stalling in the air. If this light turns red, you should either decrease your elevator or speed up the engine. The altimeter displays the current altitude of your aircraft above ground level.

There are 4 instruments to the lower right of the instrument panel which are controlled with the number keys

www.commodore.ca

1-4. Instrument 1 retracts and releases the plane's landing gear. When the indicator is red, the landing gear is out. Be sure you retract the landing gear in the air only, to avoid an embarrassing situation. Instrument number two controls the carburetor fuel mixture—white is lean, red is the normal rich mixture for flight. Instrument number three controls carb heating, which prevents the carburetor from icing over on cold days. Red is on, white is off. Since the air temperature is always warm when using *Microsim,* you need not use this control if you don't want to. Instrument 4 indicates which fuel tank is currently in use, the "red" or the "green." Each of the number keys 1-4 toggles each indicator to one particular setting or the other.

The Collision Warning Indicator (CWI) is a radar instrument that will turn red when there is a collision danger to the aircraft. This could be almost anything—a flock of birds, another aircraft, etc. The CWI becomes active at some arbitrary altitude above 4000 feet. When it turns red, use the > and < keys to steer your aircraft out of the way. In *Microsim,* dangers exist only in one dimension (you cannot fly above or below an obstacle to avoid it—you must steer around it).

Finally, to round out the complement, there are three indicator lights to the right of the instrument panel. The top light indicates whether the aircraft's external wing/tail lights and internal lighting are on or off (red=on, white=off). The middle light indicates whether the cabin's air conditioning system is on or off (red=on, white=off). The bottom light indicates whether the cabin heating system is on or off (red=on, white=off). Press 5, 6, or 7 respectively to toggle each of these functions on or off. The heating, air conditioning, and external lighting cause fuel to be eaten up a little faster than usual, so don't use them unless it's necessary.

When you first RUN *Microsim,* choose the demo flight (Press O) to get an idea of how to fly the aircraft. Let's go through a quick test flight right now. Read the following and follow the instructions:

1. With the plane on the ground, press and hold the plus (+) key to rev up the engine. Hold this key down until your groundspeed indicator shows around 65-75.

2. Hold down the U key to raise your elevator, until the indicator shows about 25-30.

3. Press the L key to lower your flaps. The aircraft should now take off, which will be indicated by the border of the screen turning blue (or black at night).

4. Lower your elevator (using D) until it reads below 5, to keep the aircraft from ascending too fast. Then press the 1 key to retract your landing gear. As you climb skyward, keep an eye on the airspeed indicator—the engine of this plane slows down intermittently. Apply a little throttle if need be. Also watch your altitude—the plane will quit if it goes above 31000 feet. Adjust the elevators for level flight once you reach cruising altitude. If you're flying above 4000 feet, keep an eye on the CWI (collision warning indicator). If it turns red, steer the plane out of the way using the rudder controls. Be quick about it! If you're going over 100 MPH when the CWI turns red, you will have only seconds to steer out of the way before a collision. (Note: If you pass through a cloud layer, the screen border will turn white.)

5. Landing—I'll leave this up to you, to provide you with a challenge. A few hints: decrease your speed to below 80 knots before landing, or your plane will bounce severely and you will crash. Remember to put your flaps up, or the plane won't land. Also be sure to lower your landing gear before landing, or the result may well be unpleasant!

Some of *Microsim*'s functions can be optionally controlled with a joystick plugged into Port 2. To increase or decrease throttle, hold down FIRE while pushing the stick forward or backward. To raise or lower the elevators, push the stick forward or backward without pressing FIRE. To move the rudder left or right, push the joystick in the corresponding direction.

I hope you have as much fun using *Microsim* as I had writing it (and I did have a very good time). If you have any comments or questions regarding *Microsim,* write me care of *Ahoy!*. Please restrict yourself to questions about the program—I am not a pilot. Also, please, no letters from pilots telling me how unrealistic my simulation is—it is not meant to be completely realistic, just to be an enjoyable simulation. I guess you could say *Microsim* is a flight simulator SIMULATOR. □

SEE PROGRAM LISTING ON PAGE 133

RENT BUY

commodore 64

IBM Apple – Atari Software

EXCITEMENT

Hint Book 75 Adventures $19.95

Diskey (D)	$ 10.88
Seafox (D)	$ 8.98
Superbase (D)	$ 36.95
Drelbs (D)	$ 7.98
Protector II (D)	$ 7.98
Sentinel (D)	$ 7.98
Zeppelin (D)	$ 7.98
Adventure Master (D)	$ 18.95
Letter Wizard/Spell (D)	$ 28.95

+ MUCH MORE

All titles new and have manufacturers warranty. Limited quantities.
Add $3.00 shipping and handling

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. 294

www.commodore.ca

SCRATCH PAD

For the C-64

By Don Schmidt

If while programming you're not using the area of memory from 49152 to 53247 (user RAM), *Scratch Pad* may be helpful for a variety of purposes. The program allows you to create in the direct mode and safely store three 'screenfuls' of information (three 1000-byte blocks) in the above area of your Commodore 64's memory. Once stored, each of the 1000-byte blocks can be instantly reprinted to the screen at the touch of a single key.

You can design your own reference material or cheat sheets, use one or more screens as a scratch pad to jot down important notes or calculations, or maintain an ongoing 'Table of Contents' of your program's subroutines and their line numbers and keep a record of all your variable symbols and what they stand for. You'll quickly learn that you can also use *Scratch Pad* as a general file for letters, recipes, graphics designs, addresses, and much more.

Type in *Scratch Pad* exactly as listed on page 123 and save it to disk or tape before running it. Then run the program, and when the READY prompt appears type SYS 49188 and press RETURN. If you haven't made any typing errors, pressing the f1, f3, or f5 key should fill the screen with garbage. What you are seeing are the 1000-byte blocks of 'unprepared' memory.

Now clear the screen (if you do so by pressing the RUN/STOP-RESTORE combination, you'll have to SYS 49188 again to reactivate the program). Print anything you want to the screen. To avoid the SYNTAX ERROR message, press the SHIFT and RETURN keys together instead of the RETURN key alone to return the cursor to the left edge of the screen. When the screen is prepared the way you want to save it, home the cursor and press the SHIFT key and the f1, f3, or f5 key together. This should have saved your screen of information. Now clear the screen and press the same function key *without* the SHIFT key. Your screen of information should reappear. You can recall, modify, and resave your screens as often as you wish.

The second listing (see page 124) is a *Load/Save* routine written in BASIC, but using several of the Kernal's machine language routines to save and load your screens to and from disk or tape. *Load/Save* not only allows you to save the screens you have designed, but the machine language of *Scratch Pad* as well. So when you load your screens back into memory, *Scratch Pad* is automatically loaded and activated and all you have to do is press the function keys. □

SEE PROGRAM LISTINGS ON PAGE 123

Reader Service No. 115

www.commodore.ca

PROGRAMS THAT WRITE THEMSELVES

Simple and Relative Address Modification

By Mark Andrews

One of the most strange and wonderful features of Commodore 64/128 assembly language is the availability of a programming technique known as address modification. Once you know how to use address modification, you can create machine code that will actually rewrite itself on command, sometimes saving enormous amounts of money and processing time.

Here is a short subroutine that shows how the principle of address modification works:

```
ASSEMBLY LANGUAGE          MACHINE LANGUAGE

Label    Code              Address Code

ADDRESS  LDA VALUE         8040    AD A7 02
         INC ADDRESS+1     8043    EE 41 80
         BNE NEXT          8046    D0 03
         INC ADDRESS+2     8048    EE 42 80
NEXT     RTS               804B    60
```

Examine this subroutine carefully, and you'll see that when it is called, the accumulator is loaded with a number labeled VALUE. This value could be any eight-bit number. In the above example, however, the value of VALUE is the hexadecimal number $02A7. Look closely, and you can find the $02A7 in the language listing of the above subroutine. It is the number listed, low-byte first, following the hexadecimal number $AD in the first line of the listing. (The value $AD is the machine language equivalent of the instruction LDA.) So, when the subroutine listed above is executed, the first thing that will happen is that the accumulator will be loaded with the value of memory address $02A7.

Loading the accumulator with an eight-bit value is a simple enough operation. But in the next three lines of our sample subroutine something quite extraordinary happens. The algorithm that is used in these three lines is a common operation for incrementing a 16-bit number. But just what number is being incremented in this example? Once you know the answer to that question, you'll know the secret behind address modification.

Take a very close look at the second and fourth lines of the illustrated subroutine, and you'll see that the value being incremented is whatever 16-bit value happens to reside in a pair of memory addresses labeled ADDRESS+1 and ADDRESS+2. And what addresses are those? Well, when the subroutine that we're examining is assembled into machine language and loaded into memory, the machine language equivalent of the assembly language instruction LDA will be stored in a memory address labeled ADDRESS. And this address, as you can see by consulting the machine language column of the listing, is memory location $8040 (the dollar sign indicates that the address is a hexadecimal number).

Now the plot thickens. When our sample subroutine is assembled and executed for the first time, the accumulator will be loaded with the value stored in memory address $02A7. Then, in the next three lines of the subroutine, the operand of the mnemonic LDA will be incremented from $02A7 to $02A8. So the next time the subroutine is called, the accumulator will be loaded not with the value of memory address $02A7, but with the value of memory address $02A8. And the operand of the instruction LDA will continue to be incremented in this way every time the subroutine is called.

If you're familiar with indirect indexed addressing, you'll probably notice that indirect addressing and address modification can be used in a similar way. But address modification has certain advantages over indirect addressing in some applications.

Some programmers don't like to use address modification because routines that make use of it are not easily transportable from one program to another, and are often somewhat difficult to understand. Nevertheless, address modification is a very powerful technique that is used quite often in high-performance assembly language programs. Routines that use address modification are compact and fast-running, and they leave the X and Y registers of the 6510 chip free for other uses. And, although address modification routines can be used in much the same way as zero-page indexed addressing, they don't require the use of zero-page memory, which is always in short supply. So a thorough understanding of address modification techniques can be of great value to an assembly language programmer.

RELATIVE ADDRESS MODIFICATION

A more sophisticated variety of address modification, called relative address modification, is used in the assembly language program called SKETCHER that appears on page 142. SKETCHER is a completed version

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

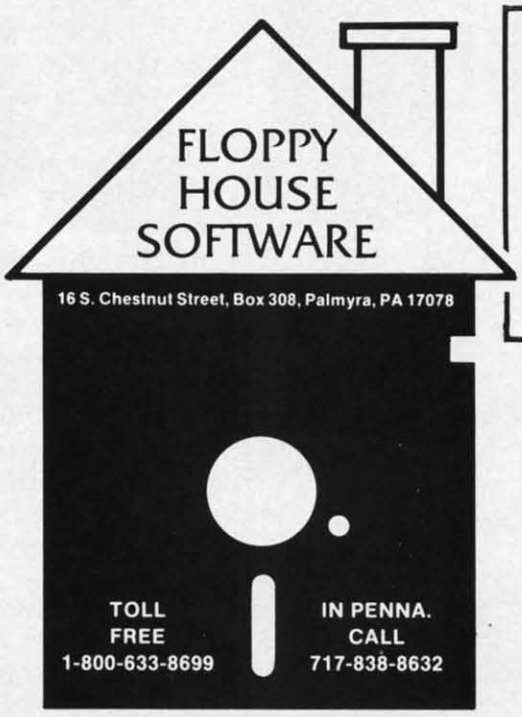
Bank Street Music Writer	34.95
Bulk Discs - Quantities of 10	9.95
Clone Machine	37.95
Compuserve Starter Kit	21.95
Copy II 64	27.95
Facsimile	17.95
Fast Load	27.95
Isepic	52.95
Mach 5	24.95
Maxell Disks - Box of 10	15.95
Ortho Complete Gardening	37.95
Pro-Tech Vinyl Covers	5.35
Simon's Basic	38.95
Tac 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. 279

www.commodore.ca

of a program that was presented, broken down into two parts, in last month's column. With the SKETCHER program and a Commodore-compatible joystick, you can draw high-resolution pictures on a computer screen. When you've finished drawing a picture, you can hit your joystick's trigger button and clear your screen.

In the SKETCHER program, relative address modification is used to make the program branch to a set of subroutines labeled UP, DOWN, LEFT, and RIGHT. These subroutines are used to detect the direction in which the joystick is being held, and to move a cursor in a corresponding direction on the screen. As you may know, this is one way in which an ON...GOTO routine could be used in a BASIC program.

The address modification routine in SKETCHER makes use of a data table that appears in Lines 414 through 424. This table is labeled RELADS (which stands for "relative address"). But the values of the bytes in the RELADS table are not defined as specific numbers. Instead, each value in the table is defined as the result of a subtraction operation—specifically, as the difference between the address of a given value in the table and the address of a line labeled MODR1 in the SKETCHER program. The line labeled MODR1, as you can see by studying the SKETCHER program, is the first line in a series of joystick-reading routines. So, by using the addresses of MODR1 as a base, the address of each joystick-reading routine in the program can be easily calculated.

Look carefully at the RELADS table, and you will see that each value in the table is equal to the address of one specific joystick-movement routine, minus an offset value that corresponds to the address of Line 294 of the SKETCHER program—the line labeled MODR1. And that is how the address of MODR1 is used to calculate the addresses of the joystick-reading routines in the program.

The segment of the SKETCHER program that uses address modification extends from Line 289 through Line 297. In Line 290, the direction switch of a joystick has just been read, and the value thus obtained has been stored in the 6510 chip's X register. If the joystick's trigger button is currently being pressed, the screen is cleared and the joystick is read again. If the trigger button has not been pressed, the accumulator is loaded with an eight-bit value that points to a specific address: namely, the address of one of the joystick-movement routines in Lines 298 through 350 of the SKETCHER program. An offset that points to the address of the desired routine is then calculated and stored in an address labeled MODREL+1.

The address of MODREL+1 can be found in Line 293 of the SKETCHER program:

```
293 MODREL BNE *
```

In assembly language programs that are written using the *Merlin 64* assembler (as this one was), an asterisk used in the above fashion is always interpreted as the current value of the assembler's program counter. So, when the above line is assembled into machine language, memory addresses MODREL+1 and MODREL+2 will hold nothing but a 16-bit value pointing to the address of MODREL+1. However, as soon as SKETCHER is executed, the contents of MODREL+1 and MODREL+2 will be changed. MODREL+2 will retain its original value, but MODREL+1 (the low byte of the value stored in MODREL+1 and MODREL+2) will be changed to whatever value is currently stored in the accumulator. This value, as we have seen, will now be the value of a specific byte in the data table labeled RELADS. And each byte in that table, as mentioned previously, is an eight-bit pointer which the SKETCHER program uses as an offset to calculate the address of a specific joystick-movement routine.

Address modification is quite an advanced concept, even for an experienced assembly language programmer. So if all of this seems a little foggy at first, please don't despair. Just run the SKETCHER program, observe what it does, and take another look at the program to see how it does it. Once you understand what the program does, learning how it does it should be much less of a problem. □

SEE PROGRAM LISTING ON PAGE 142

NEXT MONTH: Customizing the Commodore 64/128 Character Set—How to create your own text characters, and how to incorporate text characters into high-resolution graphics programs.

Program Like a Pro!
In the C-128 Mode

New!!

C-128 Update!

1. **The Commodore 128 Mode** Learn about memory organization and operating system of the C-128 and storage on the 1571 drive.
2. **Algorithms:** Learn the formulas for computer programming tasks.
3. **Assembly Language** Updated for machine language programming in the C-128 mode as well as C-64 mode. Monitor instructions.

$14.95 Each at your favorite book or computer store or order direct. Mail or call. Master Card/Visa orders call (619) 484-3884 or (619) 578-4588 or Modem orders (619) 486-4694 Fridays-Sundays.

Dealer Inquiries invited.

Mail to:

MasterCard **microcomscribe** VISA

8982 Stimson Ct., San Diego, CA 92129

Reader Service No. 101

www.commodore.ca

CADET'S COLUMN

Buying, Interfacing, and Operating a Printer

By Cheryl Peterson

Ah! New Year's Resolution time, folks. Are we all resolved to learn to use our computers more effectively? I'm certainly going to try to pick up a few new tricks this year. My educational endeavors will probably focus on the C-128 and Amiga computers, but I'll still be fiddling around with my 64. Since Commodore users catch on fast, I've got to hustle to stay at least one step ahead of you.

This month I'll focus on printers; how to choose one, and what to do with it after you've got it. Those of you who already have a printer, stick around. We've got something for you, too. We'll take a look at printer control codes, ASCII lookup tables, and DIP switches. Although it may sound a little complicated, interfacing a printer to a Commodore computer isn't so bad. And once you get the hang of it, you can do some really amazing things.

CHOOSING THE RIGHT PRINTER

Because you could end up spending $500 or more by the time you're really happy with your printer, it's a purchase you should consider carefully. As with anything you buy for your computer, you need to make sure your prospective buy works with the software and other peripherals you already have. If the printer you're considering does everything but make peanut butter and doesn't work with your word processor or graphics package, keep looking. If you decide to get that printer anyway, you'll have to buy new software.

So before you start shopping, make a list of the programs you have and the printers that work with them. If you want to use *Print Shop*, for instance, there is a large list of printers and interfaces that will work, but Broderbund recommends non-Commodore printers because the printouts look better. Some programs (*Fontmaster* is one) don't work with Commodore printers. Some graphics packages work better with Epson printers than with Commodore. The best way to be sure is to buy from a store where you can check the printouts to be sure you're satisfied.

Also keep in mind what you need to do with the printer. There are many different types of printers, but I'll divide them into two categories, printers that do graphics and printers that create typewriter quality (TQ) output. TQ printers create a page that is hard to differentiate from one created with a typewriter. The nicer ones turn out print comparable to an IBM Selectric. The cheaper ones look cheap, but perfectly legible. These machines have a few drawbacks. They are either slow at printing or very expensive. And you get no graphics!

The graphics-capable printers come in varieties from dot matrix with ribbons to thermot dot matrix to laser printers that rival typesetting machines. I seriously doubt if any of you want to put a $2000-$4000 printer on your Commodore, and as far as I know the software to drive laser printers isn't yet available for us, so we'll stick to the standard dot matrix type. Some of these support a letter quality printing mode that is not as crisp as the TQ printers, but for most uses the difference isn't worth mentioning. Generally speaking, dot matrix printers are faster than TQ printers, even in their letter quality mode.

POINTS TO CONSIDER WHEN BUYING A PRINTER

1. What software do you use? Want to use?
2. Do you need graphics or typewriter quality? Do you need letter quality printing?
3. Do you want color?
4. Do you need speed? Quiet?
5. How much money do you want to spend?
6. Commodore or third party? Does it require an interface?
7. Cost of ribbons, replacement parts like printheads or alternate printwheels.
8. Tractor feed or friction feed or both? Does tractor feed cost extra?
9. Serial or parallel communication? Both?

Most printers offer friction feed or tractor feed to get the paper in front of the printhead. I personally prefer tractor feed, because I print multipage documents and I hate to keep putting in new sheets of paper. Each page must be inserted and aligned individually with a friction feed printer. This is a necessary consideration only if the software you use doesn't have a "pause at end of page" feature. Otherwise, it's just a matter of convenience. Almost all printer manufacturers offer tractor feed

www.commodore.ca

as an option, though sometimes it costs extra.

Not all printers work with the Commodore. In fact, most need some kind of interface because the Commodore computers use PETASCII instead of the standard ASCII (American Standard Code for Information Interchange) used by other computer and printer manufacturers. Unless you buy a Commodore printer (or one with a built-in Commodore interface), you'll have to pay from $50-$125 for an interface cable to run from your computer to the printer (see below).

Also consider whether or not you plan to move up to a different computer later on (or already own another computer). Most computers use an RS-232C connector or a Centronics parallel connector to interact with the outside world. Commodore's serial connectors are nonstandard and you may have a hard time getting a Commodore-compatible printer to work with another computer. If you face this problem, I'd recommend looking for a dual interface printer or a Centronics printer and a third party serial to parallel interface. There are Commodore-to-RS-232 interfaces being marketed, but they are less plentiful and offer a more limited variety.

Before I get into the nitty gritty of interfacing printers with the Commodore, I want to recommend a couple of articles that have appeared in *Ahoy!* in the past: Tom Benford's article on choosing a printer in the December '84 issue and Morton Kevelson's three-part series on printer interfacing beginning in that same issue and continuing in February and March '85. *Ahoy!* has run many articles on getting the most from graphics programs and the most popular printers.

We haven't done anything on getting to the fancy print styles that can be used in text printing: things like near letter quality, boldface, doublestrike, alternate pitch and line spacing, super- and subscript, expanded and compressed characters, italic printing, and international character sets. Most of these are supported by the more popular third party dot matrix printers. TQ printers support many of these, but expanded, compressed, and italic print aren't usually possible. Commodore printers may also have some of these features.

Rather than be too specific and limit our discussion to only some printers, I'm going to explain the general principles involved in communication between printers and computers. You should be able to apply this information to whatever model you use.

INTERFACING NICETIES

ASCII is a way of representing all the commonly used typing characters with numbers. These numbers are what is sent along the cable between a computer and printer. The computer converts the numbers back into letters and prints them. Special characters are used to represent things like tabs, carriage returns, linefeeds, and other printhead positioning commands. These characters are all part of the standard established years ago. You can find a modified version of the ASCII code list in the *Programmer's Reference Guide* or your *Commodore 64 User's Guide.* Commodore chose to redesignate some of the code in order to facilitate graphics usage.

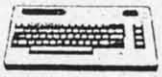
In addition to these characters, each printer company has chosen certain character strings to recognize as commands that cause the printer to switch printing modes. Getting an Epson printer to switch from 10 characters per inch (CPI) to 12 requires sending an escape (ESC) character followed by an M. Frequently printer manufacturers use the ESC to designate that the characters that follow are a command. Have you heard computer users talking about sending escape or control codes to their printers? Perhaps your word processor's documentation mentions sending "special" codes to access alternate print styles? This is what they mean.

In some printer manuals, these characters may be represented by their CHR$ codes. The reason for this is that many printer manuals expect the user to be sending these codes using a BASIC program. For instance, a PRINT CHR$(27)"M" could be used to send the 12 CPI code to the printer. Of course, with the Commodore you'd have to open a channel to the printer first. Somewhere in almost every printer manual, there is a table that shows the codes needed to get that particular printer to turn on the fancy footwork. Sometimes that's all you need.

With many application programs (word processors, especially) it's not that simple. If the program offers imbedded print codes that will automatically turn on fancy

ELECTRONIC ONE*

commodore



ELECTRONIC ONE

THE LOWEST PRICES THE BEST SERVICE

CALL (614)864-9994

P.O. Box 13428 • Columbus, Oh. 43213

COMMODORE HARDWARE	
C128	259.99
C64	139.99
1541 DISK DRIVE	169.99
1571 DISK DRIVE	249.99
1702 MONITOR	179.99
1902 MONITOR	249.99
PRINTERS	
STAR SG 10	228.00
PANASONIC 1091	228.00
EPSON LX80	228.00
COMMODORE 803	99.99
COMMODORE 802	199.99
CONNECTION INTERFACE	54.99
CARDCO G. WIZ	44.99
MONITORS	
TENIKA MJ22 (RGB)	269.99
TENIKA MJ10	169.99
XTRON 14" COLOR/GREEN	139.99
INDUS GT DISK DRIVE	219.99

SPECIAL
NASHUA DISKS GENERIC S/S D/D **8.88**

SPECIAL
TOTAL TELECOMMUNICATION
MODEM-UPLOAD-DOWNLOAD
AUTO ANS/DIAL
24.77

THE LOWEST PRICES

COMMODORE SOFTWARE	
KENNEDY APPROACH	19.99
F-15	21.99
KARATEKA	18.99
PRINT SHOP	24.99
GRAPHIC LIBRARY	14.99
FLIGHT SIMULATOR	32.99
COMPUTER BASEBALL	21.99
FAST LOAD	23.99
MUSIC CONSTRUCTION	18.99
ARCHON II	24.99
GHOSTBUSTERS	19.99
ON FIELD FOOTBALL	19.99
SKY FOX	24.99
ROAD RACE	19.99
HACKER	19.99
WINTER GAMES	24.99
COMPUSERVE STARTER KIT	14.99
MACH 5	21.99

HOW TO ORDER: CASHIER CHECK, MONEY ORDER, MASTERCARD* or VISA* (Add 4% for charge cards) . . . NO PERSONAL CHECKS . . . NO C.O.D.'s . . . SHIPPED U.P.S. PRICES SUBJECT TO CHANGE.
SHIPPING: Prompt one day shipping on in-stock merchandise. Ohio residents add 5.5% sales tax. Add $3.00 on all orders under $100.00 . . . Add $5.00 on all orders over $100.00.
INTERNATIONAL: Actual freight charge on all orders outside the continental United States including A.P.O.
CALL OR WRITE FOR FREE CATALOG

CALL ELECTRONIC ONE (614) 864-9994

Reader Service No. 299

www.commodore.ca

Aquarian Software Introduces . . .
Powerful Programs at Affordable Prices

Mike J. Henry's

FAST BOOT!

- Works with **Both** 1541 and MSD Drives!
- Multiple loading options for increased compatibility.
- Disk-Based & Non-Protected

Only $14.95

Graphic Screen Exporter 64

- A Universal Graphic Converter
- Converts Anything to Anything — Including:
  Koala Pad Flexidraw
  Doodle Print Shop
  And Many More!
- Complete On-Screen Editor

NEW!

The Most Versatile Graphics Utility Ever Released for the Commodore 64!

Only $29.95

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

Now Only $17.95

The Cataloger

The Ultimate Disk Cataloging System for the 64!

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.

Only $24.95

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

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

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

Disk-Lock V2.1

At Last! A professional copy protection system is available to YOU. Now you can protect your programs from most copy programs!

- Disk-Lock is Designed to Prevent Full Disk Copies.
- Stops Over 90% of All Copiers
- Offers Multiple Protection Schemes
- Place Protection on Any Track From 1 to 40!
- Fully Menu Driven

Only $39.95

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 10 Now Available — Featuring an Advanced Sound Synthesizer

Volumes 1 through 9 Are Also Available

Dealers Welcome! — Call (503) 654-2641 For Details.

The Complete Thriller Collection

All six thrilling adventures now together on one disk! Plus — as a bonus — **Three Hours To Live!** Each is a full-length machine language text adventures that will take you weeks to complete. They feature full-sentence input (not just two words!) with color and sound.

Includes:

- Night of the Walking Dead
- Perils of Darkest Africa
- Revenge of The Moon Goddess
- Son of Ali Baba
- Frankenstein's Legacy
- Shipwrecked!
- Plus Three Hours to Live

Only $24.95
For All SEVEN!

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

To Order, Call: (503) 654-2641
VISA and MasterCard Accepted

VISA MasterCard

Dealer Inquiries Welcome!

Add $3.00 S & H Per Order
Add $5.00 for COD
Canadian Orders: Add $10.00 S & H-U.S. Funds Only, Please.
Allow 4 to 5 weeks for delivery.

Reader Service No. 113

www.commodore.ca

features for you, you may not need to use the CHR$ codes. But usually, these codes are optimized for certain printers at the expense of using them on others. For instance, one word processor I have claims to let the user switch from 10 to 12 CPI just by imbedding a special character (created by holding down the C= key and another). This may work with other setups, but for my Epson RX-80 and Cardco +G interface it did not. While it did recalculate the line length so that it put more than 80 characters on each line it didn't switch the pitch.

Instead, another special character that sent the CHR$ value of the next character to the printer had to be used to imbed an ESC in the text, and the ESC had to be followed by an M. Using this roundabout method, it was possible to access most of the Epson's neatest features. It was not, however, convenient to do so. This is one reason that I recommend checking out how the software you have will work with the printer you are interested in buying. Since most printers use a similar scheme of operation, you'll either have to learn the codes yourself or buy software optimized for your printer.

If you buy a printer that has a letter quality mode, you can be sure that your word processor won't have a built-in command to turn on letter quality. You'll have to figure out how to send the proper codes. I have such a printer and it took me two days to get the letter quality mode on it working. The Mannesmann Tally 160L needs an ESC followed by a [, a 1 and a y to turn on word processing mode, then an ESC[4y to turn on 10 CPI letter quality mode. An ESC[5y is used to turn on 12 CPI letter quality.

Though it sounds easy, reading the printer manual for the MT160L did nothing but confuse me. The authors describe the process in terms of hexadecimal values for the ASCII codes and refer the user to a cryptic table on the inside of the back cover. That's why it took two days to figure it out! Fortunately, the software I use has a printer file contained on the disk. Once the correct codes are entered into the file, it will remember them and use them each time it prints a file. I have two different printer files —one for fast printing, the other for letter quality.

Normally the Commodore's reassigned ASCII values must be translated into codes that the printer can understand in order for it to print the graphics. Intelligent interfaces do this translation, unless a special code is sent that switches them off. When working in text mode it is sometimes better to turn the graphics off altogether. Sometimes this can be done by using an internal switch.

Usually a group of DIP (Dual Inline Position) switches have to be set before the interface is used. These switches allow the interface to be used with a variety of printers, even though each printer is different. These switches may control whether a linefeed is automatically generated with a carriage return, whether the interface can be turned off by the software in the Commodore (transparent mode), and whether the printer is device four or five.

Many printers also have such switches inside. In order for interface, printer, and Commodore to work together, all the switches must be set correctly. For instance, if the printer's auto-linefeed is turned on, the auto-linefeed in the interface is turned on, and the software you are using does an auto-linefeed with each carriage return, you'll get a triple-spaced document. Which is fine, if you want a triple-spaced document. Otherwise, you'll need to turn some of those linefeeds off.

Of course, by buying a Commodore printer or one with a built-in interface, you avoid many of these problems. Again, if you are really struggling with interfacing difficulties, I'd suggest contacting your local user group. You may find someone there who's gone through the same trials.

MAYBE I CAN HELP

Although it is difficult to give advice long distance, I can be found on Viewtron. Leave me a message in the For Starters SIG and I'll try to help you figure out what's wrong. If you've heard of a new printer and aren't sure about how well it works, you might leave a message asking if anyone else has experience with that brand. I'll be happy to give help in any areas you may be having trouble with, so drop me a line. My user ID is 266399CCP. I can also be found hanging around in Viewtron's CB section under the handle Cherp! Hope to see you there. □

Next month in *Cadet's Column*: We'll learn a few lessons about structured programming by sprucing up some sloppy BASIC. Also: how computerphobic are you?

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. 286

www.commodore.ca

ALARM CLOCK

For the C-64

By Tony St. Clair

Alarm Clock is an interrupt-driven utility which uses the C-64 time-of-day (TOD) clock to display the current time and alarm at some predetermined time. When run, the program asks the user to enter the current time. Either military or standard format can be used. If the standard format is used, the program will prompt for "am or pm" after entering the hour.

After entering the current time, the alarm time is entered in the same manner. When the current time reaches the time set for the alarm, the border will flash red and an audible alarm is produced. Two options are included in the program: 1) The audible alarm can be defeated, and 2) The present time can be continuously displayed in the upper right corner; or, to avoid any interference while entering BASIC statements on the first line, it can be disabled. In either case the alarm function will still be active.

The program works as follows. First, the top-of-BASIC pointer is lowered by 512 bytes to make room for the machine language (ML) portion of the program which is POKEd into memory in the read-data loop. In this manner the alarm clock can be active while other BASIC programs are in memory. Also, none of the free RAM at location 49152 is used, so that other BASIC utilities that may be resident will not be disturbed. The ML routine changes the IRQ vector so that 60 times a second the TOD clock is read, compared to the alarm time, and, optionally, written to the screen. If it is found that the current time matches the alarm time the alarm sequence is initiated. Every half-second the border will begin to alternate between red and the original border color. During this time the border color cannot be changed using the normal POKE command. (The IRQ will change it right back.) If so indicated, a tone will be heard with each flash of the border.

Since this program uses the TOD clock and not the jiffy clock, there are some advantages. Tape saves and loads will not interfere with the TOD clock (the screen display will be temporarily halted but will be updated at the completion of the save/load). Even a cold reset (SYS 64738) will not alter the TOD registers. (The IRQ vector will, however, be returned to normal.) Also, the TOD clock is automatically kept in the hour:minute:second format that is easy to understand, as opposed to the obscure jiffy system.

The following memory locations, i.e. constants in the Data statements, can be changed to provide different effects during program operation:

Enable/Disable routine, SYS	40449
Color of flash, 0-15	40708
Rate of flash, 0-255	40699,40736
Alarm hour, Binary Coded Decimal	40737
Alarm minute, Binary Coded Decimal	40738
Time display flag, 0=No	40739
Audio level, 0-15	40680
Note frequency (Lo-byte), 0-255	40625
Note frequency (Hi-byte), 0-255	40630
Waveform, 17, 33, or 129	40645

Some of these locations may be changed at any time; others may only be altered while the alarm sequence is not activated. □

SEE PROGRAM LISTING ON PAGE 143

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. 298

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 #25-1: CHARACTER SPIRAL

This problem was submitted by Necah Buyukdura of Ankara, Turkey. Consider 25 adjacent columns and the 25 rows of the screen forming a 625-character square. Write a program which fills the blank square with 625 characters beginning in the center of the square and progressing in a counter-clockwise direction, like a winding spiral. The program should then unwind the spiral by filling the square with different characters or spaces, and the whole process is repeated. Can you come up with the fastest BASIC solution?

PROBLEM #25-2: DANDY DELETION

This problem was submitted by Jim Speers (Niles, MI) and can be approached similarly to his REM Remover discussed this month. The user specifies a low and a high line number. Write a subprogram beginning at line 60000 which deletes all program lines within those limits.

PROBLEM #25-3: COLOR CRITERIA

Solve this useful one proposed by Steven M. Steckler (Columbia, MD). When this program is run, the user types "RED, WHITE, BLUE", for example, to select the border, background, and text colors on the monitor.

PROBLEM #25-4: CENTRAL LOCATOR

My word processor has a text centering function which is activated before the line of text to be centered is typed. The cursor is at the midpoint of the line initially. Here's what it looks like when the word "Test" is typed (the cursor is indicated by "□"):

□
T□
Te□
Tes□
Test□

The first letter appears at the cursor's initial position and the cursor moves one space to the right. The second letter appears where the first letter was, the first letter is pushed to the left, and the cursor stays where it was. This process is repeated. The odd numbered characters replace the cursor and the cursor moves one space to the right. The even numbered characters shove all the text on the line one space to the left and the cursor doesn't move. Can you program this function?

This month we will look at readers' solutions to the September 1985 *Commodares. Problem #21-1: Geometry Fun,* proposed by Phil MacLean (Columbus, OH), brought numerous responses. The problem involved determining whether three specified points formed a straight line or not. If the three points are colinear, the program tells which of the three points is between the other two.

The solution involved not only some analytic geometry but some sorting as well. Most readers used the straightforward procedure of calculating the slopes of the lines between points one and two and between points two and three. If the slopes are equal (within accuracy limits of the computer), the three points are collinear. One complication is the fact that a vertical line has an undefined slope since the horizontal coordinates of the three points are equal.

The program listed below takes a more unusual approach.

```
1 REM
2 REM COMMODARE #21-1 : GEOMETRY FUN
3 REM SOLUTION BY JOYCE AND B.A. ZIDOVEC
4 A$="COLLINEAR POINTS!"
5 P$="POINT #":P1$=" LIES BETWEEN POINTS
 "
10 PRINT CHR$(147):FORI=1TO3:PRINT"PAIR
```


www.commodore.ca

"IT DON'T GET NO BETTER THAN THIS"

EPYX
Summer Games.....23.95
Summer Games II....23.95
Winter Games.....23.95
Pitstop II.....23.95
Eidolon.....23.95
Rescue on Fractalus..23.95
Fast Load.....23.95
Koronis Rift.....23.95
Ball Blazer.....23.95
Greatest Football Game23.95

INFOCOM
Wishbringer.....21.95
Enchanter.....21.95
Zork 1-2-3.....23.95 ea.
Suspect.....21.95
Witness.....21.95
Sorcerer.....23.95
Cuthroats.....21.95
Infidel.....23.95
ALL INVISICLUES.....6.00

ACCESS
Beachead II.....21.95
Raid over Moscow....23.95

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

RIBBONS
SG10-15.....2.35
Oki 82,92,83,93.....2.35
Oki 182, 192.....9.75
1526.....6.50
Legend 880, 1080, 808..6.50
Epson FX, RX, MX, LX 80 4.50
Epson FX, RX, MX, 100..6.50
Epson 85.....6.50
Epson 185.....6.50
Brother 15,25,35.....4.99
Juki 6000,6100.....2.50
1525.....6.75
Banana.....6.75
MPS 801.....6.75
Axiom Elite 5 LCD.....7.99
MIN 6 Ribbons per order of any type.

BRODERBUND
Print Shop.....27.95
P/S Library 1,2,&3..16.95 ea.
Whistlers Bros.....18.95
Karateka.....18.95
Music Shop.....27.95
Bank St. Writer.....29.97
Stealth.....18.95

ACTIVISION
Alcazar.....18.95
Cross Ctry Road Race.18.95
Rock n Bolt.....18.95
Master of the Lamps..18.95
Web Dimension.....18.95
Star Rank Boxing.....18.95
On Track Racing.....18.95
Countdown to Shutdown.....18.95

BATTERIES INCLUDED
Home Pack.....29.95
Consultant.....35.95
Paper Clip 64/128.....35.95
P/C w/Spell 64/128....46.95

MICRO LEAGUE
Microleague Baseball.29.95
Managers Disk.....29.95
Team Disks.....14.95

MICROPROSE
Crusade in Europe....24.95
F-15 Strike Eagle.....21.95
Kennedy Approach...21.95
Solo Flight.....21.95
Silent Service.....21.95

DISKS
(Guaranteed Error Free)
Generic No Label
Pack of 25 w/sleeves.....18.75
Maxell MD 1D.....12.95
Drive Head Cleaning Kit.....11.95

ACCESSORIES and DISKS
Disk Notchers.....6.50
Disk Mailers.....0.49 ea(min.10)
Drive and Printer Cables (specify length).....1.20
per foot (6 ft. min.)

STRATEGIC SIM
Carrier Force.....35.95
Computer Baseball...23.95
Computer Quarterback23.95
Field of Fire.....23.95
Kampgreuf.....35.95
Mech Brigade.....35.95
Operation Mkt. Garden29.95
Germany 1985.....35.95

TIMEWORKS
Data Manager II.....29.95
Data Mgr. II 128.....49.95
Electronic Checkbook.16.95
Money Manager.....16.95
Accts. Payable.....35.95
Accts. Receivable....35.95
General Ledger.....35.95
Inventory.....35.95
Evelyn Wood Reading.29.95
SwiftCalc 128.....40.95

MISC.
Spy vs Spy.....18.95
Spy vs Spy II.....18.95
Tapper.....27.95
Doodle.....23.95
States and Traits.....27.95
Typing Tutor III.....23.95
Body Transparent....27.95
I am the 64.....18.95
Compuserve Starter...22.95
Superbase 64 V.2.....46.95
Superbase 128.....56.95
Practicalc 64.....18.95
Practifile 64.....18.95
Tax Advantage '85....41.95
Sargon III.....29.95
Koala Printer.....16.95
Super Script.....46.95
Super Bowl Sunday...19.95
Kung FU.....18.95
Karate Champ.....18.95
C-64 Koala Pads.....49.95
Flight Sim. II.....29.95
Fl. Sim. Scenery.....call
Intl. Hockey.....16.95
Halley Project.....23.95
Ultima III.....35.95
Ultima IV.....35.95

AMERICAN INTERNATIONAL COMPUTER PRODUCTS
P.O. Box 1758, STATEN ISLAND, NY 10314
PHONE ORDERS 718-351-1864

Ordering Info: In stock items shipped within 24 hrs. for prepaid orders (cert. chk. - visa - m/c). C.O.D. orders accepted please add $4.00 must be cash or cert. chk. only. NYS residents add 8¼% sales tax. Visa & Master Card add 4%. All orders add 3.50 shipping and handling. International Orders add $5.00 + S & H. School inquiries invited. Personal checks allow 2 wks. for clearance. Availability and pricing subject to change without notice. Orders over $125.00 receive free shipping in continental U.S. Make all checks payable to: American International Computer Products.

TRY OUR GREAT EXPRESS SERVICE

****SPECIAL INTRODUCTORY OFFER****

FREE 1541 Disk Drive Cleaning Program With Every Order

Reader Service No. 106

www.commodore.ca

```
(X,Y) #"I:INPUTX(I),Y(I):NEXT
15 S(1)=SQR((X(1)-X(2))^2+(Y(1)-Y(2))^2)
16 S(2)=SQR((X(2)-X(3))^2+(Y(2)-Y(3))^2)
17 S(3)=SQR((X(3)-X(1))^2+(Y(3)-Y(1))^2)
20 T1=X(1)*Y(2)+X(2)*Y(3)+X(3)*Y(1)
21 T2=X(1)*Y(3)+X(2)*Y(1)+X(3)*Y(2)
25 IF T1-T2 THEN PRINT "NOT "A$:GOTO 70
30 PRINT A$
40 IFS(3)>S(1)ANDS(3)>S(2)THENPRINTP$"2"
P1$"1 & 3":GOTO 70
50 IFS(2)>S(1)ANDS(2)>S(3)THENPRINTP$"1"
P1$"2 & 3":GOTO 70
60 IFS(1)>S(2)ANDS(1)>S(3)THENPRINTP$"3"
P1$"1 & 2":GOTO 70
65 PRINT"TWO OR MORE POINTS ARE IDENTICA
L"
70 WAIT 198,1:GOTO 10
```

This program from Joyce and B. A. Zidovac (Kitchener, ONT) uses a method which Wylie A. Smith (Annapolis, MD) refers to as "The Surveyor's Method" based on Green's Theorem. The quantity (T1-T2) in line 25 is zero if the three points are collinear. Otherwise it is non-zero.

Lines 15 through 17 of their program calculate the lengths of the line segments between each pair of points. For three collinear points, the longest line segment is between the two outside points. The point which is between the other two shares the two shorter line segments. Lines 40 through 60 decide which point is in the middle. Line 70 waits until a key is pressed at which time execution resumes at line 10. The key is actually used by the program, so don't press the <RETURN> key to continue. Your next keystroke should be the next X coordinate to be entered.

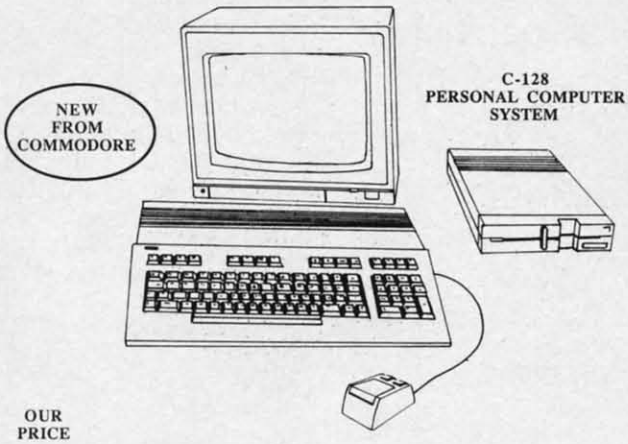
Wylie A. Smith mentioned that this method is applicable to any number of points. He also said that the value of (T1-T2) can be used to find the area of the polygon defined by non-collinear points. The area is simply the absolute value of (T1-T2) divided by two. You could easily modify the program to display the area. If you are interested in the general method of determinants for finding the area and even the direction of travel around an N-sided polygon, send me a stamped envelope with your request.

Problem #21-2: Logical Fun submitted by Michael Marron (Stony Brook, NY) was fun for quite a few reader. The solution from Frank T. Smith (Wilmington, DE) is representative of the majority of the solutions received.

```
1 REM
2 REM COMMODARE #21-2 : LOGICAL FUN
3 REM SOLUTION BY FRANK T. SMITH
4 REM
10 FOR S=1 TO 2
20 IF S=1 THEN PRINT"STATEMENT #1 IS TRU
E"
30 IF S=2 THEN PRINT"STATEMENT #1 IS FAL
SE"
40 FOR A=65 TO 95 STEP 10
50 FOR B=65 TO 95 STEP 10
60 FOR C=65 TO 95 STEP 10
70 FOR D=65 TO 95 STEP 10
120 IF A=95 AND C<>65 THEN 1000
130 IF C=75 AND A<>65 THEN 1000
140 IF A<=D THEN 1000
150 IF B<>95 AND A<>75 THEN 1000
160 IF C=85 AND D=65 THEN 1000
170 IF C<>95 AND D<>85 THEN 1000
180 IF B<>85 AND D=85 THEN 1000
190 T=0
200 IF A=B OR A=C OR A=D OR B=C OR B=D O
R C=D THEN T=1
210 IF S=1 AND T=1 THEN 1000
220 IF S=2 AND T=0 THEN 1000
230 PRINT "AL:"A;" BETTY:"B;" CONNIE:"C;
" DAN:"D
1000 NEXTD: NEXTC: NEXTB: NEXTA: NEXTS
```

Frank's program combines the two parts of the problem. The first statement that all grades must be different is true for part one and false for part two. Several readers correctly pointed out that there are many solutions unless we assume that the only valid grades are 65, 75, 85, and 96. Lines 40 through 70 generate all possible combinations of those four grades. Lines 120 through

IT'S HERE



OUR
PRICE

C-128 COMPUTER $269.95*
1571 DISK DRIVE $239.95*
1902 RGB MONITOR $269.95*

*ADD SHIPPING CHARGE $5.00 PER ITEM,
COD AND UPS 2ND DAY DELIVERY AVAILABLE
CALIF. RESIDENTS ADD 6% SALES TAX
VISA, MASTERCARD ADD 5%

MICROVATIONS INTERNATIONAL
5333 MISSION CENTER ROAD
SUITE 345
SAN DIEGO, CA 92108.

TO ORDER
CALL
(619) 291-2722

DEALERS WELCOMED

WE CARRY SOFTWARE AND HARDWARE FOR ALL COMPUTERS

COMMODORE IS THE REGISTERED TRADEMARK OF COMMODORE BUSINESS MACHINE

Reader Service No. 287

www.commodore.ca

180 correspond to conditions 2 through 8 of the original problem. Line 200 determines whether any of the grades are the same or not.

Whenever the conditions in any statement (lines 120 - 180) are met, the program branches to line 1000 since the current grades do not meet the requirement in the problem. For example, statement 2 said that if Al (A) gets a 95, then Connie (C) will get a 65. Line 120 of the program sees if the current values of A and C meet that stipulation. If A equals 95 but C does not equal 65, then that particular combination of grades is not valid. The program branches to line 1000 to get the next set of grades.

Only when all conditions are met does the program reach line 230 which prints out the names and their corresponding grades. There is one correct solution for all grades being different:

Al = 75 Betty = 85 Connie = 95 Dan = 65

If we assume that the first statement is false, hence "We will all get different marks" is a false statement, there are five solutions:

Al	Betty	Connie	Dan
75	65	95	65
75	75	95	65
75	95	95	65
85	95	95	65
85	95	95	75

Most readers sent solutions that included the single solution listed above as part of this second set of solutions.

Several readers used a slightly different way of approaching this problem. As an example, consider rule 7: "If Connie doesn't get a 95, then Dan will get an 85." Rather than stating

```
IF C<>95 AND D<>85 THEN ... (skip
invalid values)
```

some people preferred

```
IF C<>95 AND NOT(D=85) THEN ... (skip
invalid values)
```

In other words, if the first condition is valid but the second condition is not, the statement as a whole is invalid. Obviously both statements listed above give identical results. Use the form that is more natural for you.

Extra credit goes to Frank T. Smith, Wallace Leeker (Lemay, MO), Chris Roseman (Silver Spring, MD), David Hoffner (Brooklyn, NY), and Jim Speers (Niles, MI). These readers stated that either conditions 2, 3, 4, and 6 or conditions 2, 3, 6, and 8 can be omitted and still give the same solutions. Wallace Leeker and the proposer Michael Marron were the only two to list both sets of extraneous conditions. The other readers listed only one. There were no exotic methods revealed for solving this bonus problem. Apparently everyone simply replaced

PROTECTION REVEALED

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**

ALL C-64

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**

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.

Disk **$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.

Disk **$24.95**

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 **$19.95**

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!

Disk **$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**

Your order will be shipped within 48 hours.

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

MasterCard VISA

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.

Dealer Inquiries Welcomed

CREATED BY USERS FOR USERS

Reader Service No. 288

www.commodore.ca

groups of statements with REM statements until the proper combination of unneeded statements was found.

The biggest response this month was for *Problem #21-3: Decimal Columns* proposed by E. Harvey Hammett (Houston, TX). The program below was adapted from the solutions of Matt Shapiro and Ken Karow (unfortunately their addresses were on their envelopes, not on their listings; the envelopes were discarded).

```
1 REM
2 REM COMMODARE #22-3 : DECIMAL COLUMNS
3 REM SUGGESTED BY KEN KAROW
4 REM        AND BY MATT SHAPIRO
5 REM
6 SP$="                                        "
10 DC=8 :REM  DECIMAL COLUMN (1 TO 30)
20 FOR I=1 TO 5:READ N:GOSUB 1000
30 NEXT I : END
100 DATA 12.5,-134.56,-.0026,23,1.234
999 REM +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
1000 F$=STR$(N):FOR L=1 TO LEN(F$)
1010 C$=MID$(F$,L,1):IF C$<>"." AND C$<>
"E" THEN NEXT L
1020 PRINT LEFT$(SP$,DC-L)+F$:RETURN
2000 REM -=-CHANGE N TO F$ IN 20:DELETE
F$=STR$(N) IN LINE 1000 TO PRINT
2010 REM   SCI. NOTATION AS DECIMAL
```


Connect with the leading UNCENSORED national computer communications network.

Adults can experience "live" interactive chatting, electronic mail and many other full-service videotex features.

For a FREE brochure call (212) 972-4719 or write:

CVC ONLINE, dept. SL2 801 Second Ave., N.Y., N.Y. 10017

Reader Service No. 297

Line 10 allows the programmer to specify the column on the screen or the printer at which the decimal points should be aligned. The three-line subroutine at lines 1000 to 1020 converts the numerical value in N into a string value. The decimal point or the exponential "E" is located in line 1010. The proper number of spaces and then the number are printed in line 1020. This version of the program displays numbers which are between −0.01 and 0.01 in their scientific notation format, e.g. 2E−3 instead of 0.002. If you prefer to keep the number in the same form as it is entered, change the N in line 20 to F$ and delete the F$=STR$(N) in line 1000. Many other readers sent very similar programs and procedures.

To send the output to a printer instead of the screen, simply add line 7:

```
7 OPEN 4,4
```

and change the PRINT in line 1020 to PRINT#4.

The shortest BASIC solution to *Problem #21-4: REM Remover* was written by Don French (Minneapolis, MN). The problem proposed by Jim Speers (Niles, MI) was to append a routine onto a main program which would remove all program lines which begin with either a REM statement of a semicolon. Don's solution is listed below.

```
1 REM
2 REM COMMODARE #21-4 : REM REMOVER
3 REM SOLUTION BY DON FRENCH
4 REM
55000 J=43:POKE631,19:POKE632,13:POKE633
,13:PRINT"[CLEAR][DOWN]GOTO55000"
55003 IFK=58ORK=143THENPRINT"[HOME]";STR
$(PEEK(J+2)+PEEK(J+3)*256):POKE198,3:END
55005 J=PEEK(J)+PEEK(J+1)*256:K=PEEK(J+4
):IFJ>0THEN55003
```

Most of the solutions to this problem were similar to Don's approach. He uses the "dynamic keyboard" technique. Characters are POKEd into the keyboard buffer during execution of the program. When the program ends, those keystrokes are executed by the computer just as if they had been typed directly. (Refer to this month's *Rupert Report* for another example using this technique.) Typically the keystrokes cause the program to run again, after some onscreen editing features have been utilized.

Specifically, line 55000 puts the <HOME> and two <RETURN> characters (characters 19 and 13) into the keyboard buffer. It also clears the screen, moves the cursor down to line 2 on the screen, and prints "GOTO 55000". Line 55003 looks at the first character in the program line to see if it is a semicolon (character 58) or a REM statement token (character 143). If so, the cursor is HOMEd and the line number of that line is determined and printed on the screen. The value three is put into the keyboard buffer counter, and the program ends.

The computer executes the three keystrokes it finds in the keyboard buffer. It moves the cursor to the top cor-

www.commodore.ca

ner of the screen and behaves as if the <RETURN> key is pressed. Consequently the program line corresponding to the line number on the screen is deleted just as if you had typed a line number, then pressed <RETURN> to delete that line.

Actually K equals zero on the first time through this subprogram so line 55003 is bypassed. Line 55005 calculates the memory location of the first program line and stores it in J. The first two bytes starting at location J are the pointer to the start of the next program line. The bytes at location J+2 and J+3 are the line number. The first character in that line is stored in location J+4. K is now given the value of the first character in the line.

If the end of the program has been reached, the next-line address in J will equal zero. In that case, the program ends. If J does not equal zero, the program branches back to line 55003 where the value of K is tested.

This program starts back at the beginning of the main program every time one line is deleted. Some readers sent solutions which kept track of the last line deleted. Theoretically they should be faster than the program above. On the other hand, Don's program is fast enough to be quite impressive.

Jim Speers uses a similar program during debugging. Since a line beginning with a colon still functions properly, Jim adds trace statements to print out variables or to halt the execution to his program but precedes them with a semicolon. Then when debugging is finished, he types RUN55000 to clean up the program. Normally you should load this utility first, then type your program to be debugged. Alternately you may load the main program, then type or merge the REM Remover utility.

Congratulations and thanks to the many others with solutions to these *Commodares*. People with valid programs this month who were not mentioned earlier include the following:

Richard Pohland (Pittsford, NY)
Don Ackerman (Grass Lake, MI)
Paul Mather (Warminster, ONT)
David Butcher (Morgantown, WV)
Ross Parlette (Sunnyvale, CA)
Brian Wilcox (New Britain, CT)
David Wright (New Britain, CT)
Kenneth Hill (Kansas City, MO)
Steven Steckler (Columbia, MD)
Larry Anderson (San Andreas, CA)
Ron Barnhouse (Zanesville, OH)
Eric Biberhofer (Dundas, ONT)
Gerald Pothier (Yarmouth City, NS)
A. D. MacDonald (Millgrove, ONT)
Chuck Slotter (Philadelphia, PA)
James E. Killman (Memphis, TN)
Dennis Robertson (N. Pt. Richey, FL)
David Rasnake (New Port Richey, FL)
Ed Polyberne (Bricktown, NJ)
James Borden (Carlisle, PA)
John R. Prager (Bay City, MI)
Mark Bearden (Steele, AL)
Ron Weiner (Levittown, PA)
Bill Binder (Northville, MI)
Russell Prater (Parker, FL)
Chris Barth (Clinton, NJ)
Doug Olney (Coventry, RI)
Fred Theilig (Riverside, RI)
Rick Tyhurst (Ridgecrest, CA)
Jeffrey Mantei (Anderson, IN)
Linda C. Garcia (Fontana, CA)
Maurice Tift (Albany, GA)
Jim Johnston (Haskell, TX)
Paul DeLuca (Bradford, MA)
Thomson Fung (San Diego, CA)
Brian Wilcox (Benton, KY)
Steven G. Eason (Benton, KY)
Todd Hauser (Bicknell, IN)

And last but certainly not least, John Immarino (Hackensack, NJ).

One final tidbit sent by Jim Speers. What BASIC keyword will function as intended even when misspelled? Keep those solutions coming! □

IMAGEWARE
presents
EROS
An Erotic Adventure Game

You are about to embark on an erotic adventure to Eros, an uncharted island in the South Pacific Ocean. Your mission, if you choose to accept it, is to save yourself and the people of Eros, from the impending eruption of the formidable Mount Zayton. Along the way, you will encounter adventure, obstacles, and erotic sexual experiences.

Eros is an interactive fiction game. You control the story, but the story controls your mind. Your experiences while on Eros, will stimulate your imagination and create vivid images in your own mind's eye.

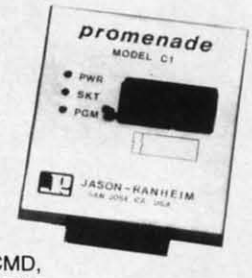
To Order: Send $24.95 plus $3.00 postage & handling to:
(Calif. residents add 6% sales tax.)

FOR C-64
IMAGEWARE
362 Los Alamos Rd.
Santa Rosa, CA 95405
FOR C-64

FOR ADULTS ONLY
Reader Service No. 117

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 eraseabl 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
Reader Service No. 116

www.commodore.ca

TIPS AHOY!

Compiled by Michael Davila

C-64 WITH C-128 KEYS

The Commodore 128 has a handy numeric keypad. Unfortunately, it does not function in 64 mode. With *Keypad Enable,* the numeric keypad and the four cursor movement keys are enabled. This utility works by intercepting the key code from the numeric keypad and then translating it to the same key code for the main set of numeric and cursor keys.

```
100 REM* C-128 (64 MODE) KEYPAD ENABLE *
110 REM        *  BY PAUL DELEO *
120 B=828
130 READ A$:A=VAL(A$):POKE B,A:B=B+1:C=C
+A:IF B=974 THEN B=B+1:GOTO 130
140 IF B<>989 THEN 130
150 IFC=18512 THEN SYS976:PRINT"[CLEAR]N
UMERIC KEYPAD BY PAUL DELEO <<ENABLED>>"
:NEW
160 PRINT "[CLEAR]ERROR IN DATA STATEMEN
TS! PLEASE CHECK!":END
500 DATA 169,3,72,169,75,72,8,72,165,197
,72,72,76,49,234,120,160,,165,203,201,64
510 DATA 208,88,169,255,141,,220,140,47,
208,173,1,220,201,255,240,73,134,197,169
520 DATA 254,72,162,8,141,47,208,173,1,2
20,205,1,220,208,248,74,176,9,72,185,183
530 DATA 3,240,2,133,203,104,200,202,208
,240,104,56,42,192,23,144,219,165,203
540 DATA 201,64,240,26,162,129,160,,144,
8,41,127,133,203,162,194,160,1,169,235
550 DATA 140,141,2,134,245,133,246,32,22
4,234,169,255,141,47,208,32,66,235,76
560 DATA 129,234,,27,16,,59,11,24,56,,40
,43,,1,19,32,8,,35,44,135,7,130,2,,120
570 DATA 169,60,141,20,3,169,3,141,21,3,
88,96
```

The keypad and cursor keys can be deactivated by pressing the RUN/STOP-RESTORE keys. To reactivate, type SYS 976.

—Paul Deleo
Troy, MI

DOUBLE SPACING

Here's a short utility for the VIC 20 and C-64 that I've found very useful when modifying a program. What this utility does is provide you with a blank line between lines of text, thereby making a listing easier to read and/or modify. It can be disabled with the RUN/STOP-RESTORE combination. A SYS to the starting address, which can be relocated, will reactivate Double Spacing. In brief, this utility wedges into the CHROUT routine of the Kernal, which will output a character to the current output channel (usually the screen), and every time a carriage return is to be output, the machine language routine prints an additional carriage control character.

```
10 REM*PROGRAM-ID.       DOUBLE SPACING.
   REM*AUTHOR.           SHAWN K. SMITH
20 INPUT "PLACE AT[RIGHT][RIGHT]747[5"[L
EFT]"]";P
30 FORD=P TOP + 2 0:READY:POKED,Y:NEXT
50 DATA 162,000,160,001,142,038,003
60 DATA 140,039,003,096,201,013,208,003
70 DATA 032,001,000,076,001,000:X=P+11
90 POKEP+3,INT(X/256):POKEP+1,X-(PEEK(P+
3)*256):L=PEEK(806):H=PEEK(807)
100 POKEP+16,(L):POKEP+17,(H):POKEP+19,(
L):POKEP+20,(H):SYSP:LIST
120 **** RUN/STOP-RESTORE DISABLES ****
     ***SYS (PLACE-AT) REACTIVATES****
```

NOTE: It also works with a printer!

—Shawn K. Smith
Bronx, NY

BASIC TITLE SCREENS

Here's a tip for those of you who want to add a magnificent touch of style to your programs! It's a REM statement that can be appended anywhere in a BASIC program. It can be used to do nearly anything that a PRINT statement can. This is activated by the listing of the line that contains it.

```
1 REM ""[DEL][RVSON][s M][whatever else]
```

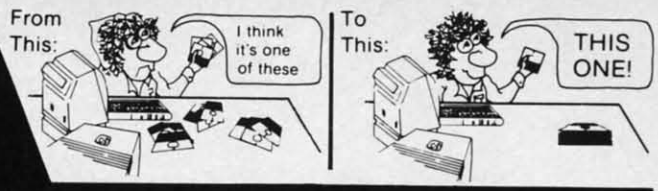
Where "WHATEVER ELSE" is in the line, you can put nearly anything that you wish. For example, to have the screen cleared when that line is listed, just place a reverse heart (shifted S) after the shifted M and press RETURN. Another example is

```
1 REM ""[DEL][RVSON][s M][s S]E[RVSOFF]T
HIS PROGRAM WAS WRITTEN BY JOHN DOE
```

This line will clear the screen, change the cursor color to white, and print the message without a line number. The contents of the REM have to appear as they do in a PRINT statement, just as SHIFT CLR/HOME appears as a reverse heart and CTRL 1 (WHT) appears as a reverse E in PRINT statements. As you can see, this routine can be used to make different sections of programs different colors, print a complete title screen when a program is listed, and endless other things to amaze

www.commodore.ca

C-64 and C-128* OWNERS:



Now YOU too can get the "Software Edge!"

Do you know the full contents of all your Disks in your Software Library?

Stop playing "Hide-and-Seek" with your Disk software! Let your Commodore 64™ "memory" do all the work for you. Easy Manual and Program Menu-Screens make everyone a "pro" in 1 day!

Here's a small "PEEK" at SOME of the FULLY-INTEGRATED POWER of PRODUCTIVITY at your fingertips. . .

MASTERDISK©:

- a SORTED Master Catalog of ALL your Disks (even those un-listable "protected" Directories: Names, ID's, File info, etc.)
- Search + List any File/Disk/ID in seconds, automatically, in any of SEVEN WAYS.
- Transfer ML-Programs or Files from Disk to Disk (CATEGORIZE your software)
- Stop disaster before it strikes: change duplicate ID's fast, without total re-copy.
- Restore accidentally-scratched files in a flash, with VERIFICATION that they're OK.
- Works with 1 or 2 Single Disk Drives (1541, MSD, etc.) with or without Printer (Commodore or ASCII).
- Handy-Reference Card for HEX/ASCII/ BINARY-TO-DECIMAL VALUE TRANSLATION + FORMULAS, included FREE.
- All this + MUCH MORE, for just $34.95. (U.S.A.), $45.95 (Canada)

NEW!! - MASTERDUAL©, "Son of MASTERDISK©" . . . same power, but with Dual Drive Speed!

- Same Catalog Features as MASTERDISK©
- Same Disk repair functions, too
- For all C-64 & C-128* systems with Dual Disk Drives (4040, MSD, etc.)
- Utilizes all Dual Drive functions for maximum speed and ease of use
- FREE Handy Reference Card for HEX/ASCII/ Binary to Decimal Translation + Formulas
- Experience FULLY-INTEGRATED POWER for just $29.95 (U.S.A.), $39.95 (Canada)

***MASTERDISK© and MASTERDUAL© are also compatible with the NEW C-128 (in C-64 Mode)!**

Don't MESS with anything LESS! ***OUR CUSTOMERS WRITE:***

"Been meaning to write and let you know how much I've enjoyed it . . . So, let me say that MASTERDISK is EXCELLENT." — Middletown, NY

". . . just the thing I need." — APO, NY (Germany)

"Hope to get my [MASTERDISK] yesterday." — Maple Grove, MN

"I am impressed with your product and would like to be able to endorse it for TPUG." — Ontario, Canada

". . . nothing else even comes close!" — Burlington, IA

We ship within 24 hours! Send Money Order or Check to:

INTEGRATED-SOFTWARE SYSTEMS
P.O. Box 1801, Ames, IA 50010
VISA/M.C./C.O.D., CALL **(515) 233-2992**
9 am - 9 pm (Central Time) Mon.-Fri.
Add $3 Shipping/plus $2 Foreign, C.O.D.
IA Residents add 4% Sales Tax.

IN CANADA: Cricket Distribution Co.,
P.O. Box 953, Cobourg, Ontario, Canada K9A 4W4
VISA/M.C./C.O.D., CALL **(416) 372-3692**
9 am - 6 pm (Eastern Time) Mon.-Fri.
Add $3 Shipping/plus $2 C.O.D.
Ontario Residents add 7% Sales Tax

Whether you have tens or hundreds of disks, now YOU can be the MASTER of your TOTAL Software Library!

Reader Service No. 258

www.commodore.ca

KIWISOFT PROGRAMS

Mainframe-quality graphics for the Commodore 64

"1985 SOFTWARE OF THE YEAR"!
New Zealand Microcomputer Software Awards

- CADPIC — Paint it, print it - 16 colors 160 x 200 screen for hobbysists and professionals ..... ~~$79.95~~ **Special Sale $39.95**
- More pictures with PICTUREDISKS Cars and planes ..... **$24.95 ea.**
- See our COMPUTER ART PRINTS - 20x24" - "Room", "Storm", "Holy Land" (3 for $25) ..... **$8.95 ea.**

Works with Epson Jx-80 and Okimate 10 Color Printers

SEND FOR OUR FREE BROCHURE — Dealer Inquiries Welcome

Add $2 p.h., Pay in advance by check, m.o., VISA or MC. In MS add 6% sales tax.

Programs on diskette from

KIWISOFT™ PROGRAMS

Ace Software, P.O. Box 9332, Columbus, MS 39705
Phone: (601) 328-9491

Reader Service No. 300

New!

Now You Can Add
ARTIFICIAL INTELLIGENCE
To Your Programs Using a Powerful Combination

SUPERFORTH 64 +AI

"A professional system thats **easy** for anyone to use"

Easy Programming with English Language Words

By Elliot Schneider & Jack Park

Here's Your Chance to Profit by being on the Forefront, Write 5th Generation Software

AI Module
- Create Intelligent Programs
- Build Expert Systems
- Write Stand Alone License Free Programs
- Construct Rule Bases
- Do Knowledge Engineering
- Use Inference Engines

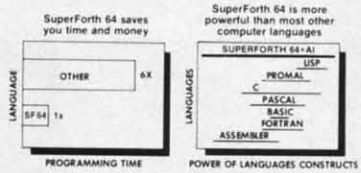
Write Intelligent Programs For:
- Home Use
- Robotics
- Medical Diagnosis
- Education
- Intelligent CAI
- Scientific Analysis
- Data Acquisition
- Data Analysis
- Business
- Real Time Process Control
- Fast Games
- Graphics
- Financial Decisions

Extended Math Module
- Fast ML Floating Point & Integer Math
- Double Precision 2E±38 with Auto. Sci Not.
- n*e* Logx Loge Sin Cos Tan SQR 1/X...
- Matrix and Multidimensional Lattice Math
- Algebraic Expression Evaluator

SUPERFORTH 64
- Interactive Interpreter
- Forth Virtual Memory
- Full Cursor Screen Editor
- Full String Handling
- Trace & Decompiler
- Conditional Macro Assembler
- 1/6 the Programming Time
- Interactive Compiler
- Romable Code Generator
- 40K User Memory
- All Commodore File Types
- Conversational User Defined Commands
- 10 to 600x Faster than Basic

Graphics I/O Module
- Hires Plotting
- Split Screen
- Printer/Plotter Ctrl
- Sprite Editor
- Turtle Graphics
- Koala Pad Graphics Integrator
- Hires Circle, Line, Arc
- Music Editor
- Sound Control

Easy Control of all I/O...
- RS232 Functions
- Access all C-64 Peripherals
- Interrupt Routines



Great Documentation
- Easy to Read 350 pg. Manual with Tutorials
- Source Screen Provided
- Meets all MVP Forth-79 Industrial Standards
- Personal User Support

A Total Integrated Package for the Commodore 64

SF 64	$59
AI Module	$39
Extended Math Module	$39
Graphics I/O Module	$29
Package Price Only	**$99**

SPECIAL OFFER
All for only **$99.00**
(Disk & Manual)
• $3.00 Ship U.S.A. • $2.00 C.O.D. U.S.A.
• $15.00 EUROPE

See Your Local Computer Store or CALL TOLL FREE
(800) 633-6335
IN CALIF. CALL
(415) 651-3160
PARSEC RESEARCH
Box 1766 Fremont, CA 94538
Dealer Inquiries Invited

VISA AMERICAN EXPRESS MasterCard

Commodore 64 TM of Commodore *Parsec Research PARSEC RESEARCH 41805 ALBRAE ST., FREMONT, CA 94538

Reader Service No. 118

your less brilliant programming buddies. It can really add a lot of uniqueness to any program!

—Kevin Brown
Anderson, IN

NO CRASH

I always buy *Ahoy!*, and I have always enjoyed the programs and other features in the magazine. But when it comes to using *Bug Repellent*, I feel very vulnerable. Many times I have mistyped SYS 49152. Most of the time the computer will lock up. This can become quite a nuisance, especially if I did not save the program that I was working on.

I abolished this problem with *No Crash*. What the program does is set an interrupt wedge which prints SYS 49152 in the upper right hand corner of the screen. To use the wedge, LOAD and RUN the program. The BASIC portion of the program will be erased. Next load the *Bug Repellent* as you normally would. To put the wedge to work, clear the screen and hit RETURN. *Bug Repellent* will then work as it normally does. Remember to clear the screen before you use the wedge by hitting the RUN/STOP-RESTORE combination.

```
5 PRINT"[CLEAR]"
10 FORX=828TO901:READA:POKEX,A:NEXT:SYS8
28:NEW
100 DATA120,169,73,141,20,3,169,3,141,21
,3,88
110 DATA96,169,58,141,31,4,169,19,141,32
,4,169
120 DATA25,141,33,4,169,19,141,34,4,169,
52,141
130 DATA35,4,169,57,141,36,4,169,49,141,
37,4
140 DATA169,53,141,38,4,169,50,141,39,4,
76,49
150 DATA234,120,169,49,141,20,3,169,234,
141,21,3
160 DATA88,96,0,0,0,0,0,0,0,0,0,0
```

—David Roscoe
Passaic, NJ

MOVE OVER MEMORY!

In many game programs it is necessary to move a large block of code from ROM to the hidden RAM below. This is usually done by the execution of a BASIC statement similar to the one that follows:

```
100 FOR J=40960 TO 49151: POKE J,PEEK(J)
: NEXT J
```

However, using this method takes over 30 seconds. The people at Commodore have included in the heart of BASIC ROM a mass move routine that you can use to do the same thing in less than two seconds. Let's say that

www.commodore.ca

we wish to move the BASIC ROM to BASIC RAM. First we must POKE location 95 and 96 with the start address of BASIC ROM, which is 40960. Then we must POKE location 90 and 91 with the ending address or BASIC ROM plus 1, which is 49152. Then we must POKE location 88 and 89 with the ending address of the RAM area, which in this case is also 49152. Finally we must SYS to location 41919 to make the move. Once this is accomplished, POKE the memory to reveal the hidden copy of RAM and you are finished. Below is the full routine in the form of a BASIC program.

```
100 POKE 95,0: POKE 96,160 :REM ROM STAR
T ADDRESS
105 POKE 90,0: POKE 91,192 :REM ROM END
ADDRESS +1
110 POKE 88,0: POKE 89,192 :REM RAM END
ADDRESS +1
115 SYS 41919
```

—S.D. Betesh
Kingston, ONT

CURSED CURSOR

Positioning text on the screen can be very complicated, especially when using the cursor keys or the POKE statement. I wrote the following program to allow easy cursor positioning on the text screen. The variable A indicates where the machine language program will be stored. I used the cassette buffer (832), but the routine can be stored at other places in memory. To activate the routine, use the command SYS A,X,Y where A is the location of the routine in memory, X is the column (0-39), and Y is the row (0-24). The next character printed on the screen will be placed at the X and Y coordinates specified by the command.

```
10 A=832:FORI=ATOA+28:READJ:POKEI,J:NEXT
20 DATA 32,253,174,32,138,173,32,247,183
30 DATA 152,72,32,253,174,32,138,173,32
40 DATA 247,183,152,170,104,168,24,32
50 DATA 240,255,96
```

—Mike Hoyt
Richardson, TX

So you think you know it all? So do most of the programmers who submit to *Tips Ahoy!* Unfortunately, very few of them actually have what it takes to be published in these pages. We're serious when we say that we want only the *best* programming tips that the Commodore community has to offer--and we'd much rather run a shorter installment of *Tips Ahoy!* than print second rate material. If you're willing to risk the postage, be assured that financial remuneration is more than competitive by industry standards. Send your best to *Tips Ahoy!*, c/o Ion International Inc., 45 W. 34th St.—Suite 407, New York, NY 10001.

BridgePro®

BridgePro is the first program I've seen that provides a challenge for the average-to-excellent bridge player. . . The documentation is excellent and allows a new bridge player to learn the basics.
—Harvey Bernstein, Antic Magazine, Feb. 1985

After having tried three other bridge programs, I find that BridgePro is indeed a pro game. . . It is designed for both the beginner and the advanced player. . . I didn't find anything that could be improved upon.
—Helen Garret, Apple-Dayton Journal, March 1985

If you like to play bridge and don't have three other players ever-eager to play, this software is a must. For bridge freaks it's good enough to justify buying a computer . . . Whether you are a "master" or a beginner, this is great software.
—Christian Basler, NY Commodore Users Group Review, Sept. 1984

BridgePro is designed to let you learn, improve, or just enjoy the card game of bridge. The program provides complete bidding, play and scoring for 1 or 2 players. Features include random hands, bidding help, demonstration mode, hand replay/quit, best hand, auto finish, duplicate mode, and fast machine language speed.

$35 DISK
CA Residents add $2.28 Sales Tax
Add $2 for COD/UPS 2nd Day Air
VISA/MasterCard Welcomed

Money Back Guarantee
24-Hour Toll Free Orders
1-800-443-0100 ext. 158

Computer Management Corp.
2424 Exbourne Ct.
Walnut Creek, CA 94596
Dealer Inquiries Invited

Commodore 64 • Atari
Apple II • IBM PC/PC jr
Reader Service No. 102

DIGITAL VISION

COMPUTEREYES™

VIDEO IMAGES ON YOUR COMPUTER!

Finally—an inexpensive way to capture real-world images on your computer's graphics display! COMPUTEREYES™ is an innovative slow-scan device that connects between any standard video source (video tape recorder, video camera, videodisk, etc.) and your computer. Under simple software control, a b/w image is acquired in less than six seconds. Unique multi-scan modes also provide realistic grey-scale images. Hundreds of applications!

Package includes interface module, complete easy-to-use software support on disk, owner's manual, and one year warranty all for $129.95 plus $4.00 S&H (USA).

Also available as a complete package including:
- COMPUTEREYES™
- Quality b/w video camera
- Connecting cable

for only $399.95 plus $9.00 S&H.

Demo disk available for $10.00 postpaid (refundable).

ONLY $129.95

Available for:
- Apple II series
- Commodore 64/128
- Atari 800/800XL/65XE/130XE

See your dealer or order direct. Mass. residents add 5% sales tax. Mastercard, Visa accepted. To order, or for more information, write or call:

DIGITAL VISION, INC.
14 Oak Street — Suite 2
Needham, MA 02192
(617) 444-9040, 449-7160

Reader Service No. 103

www.commodore.ca

INFRARAID

An Error-Trapping Wedge for the C-64

By Timothy VanDeventer

A fact of life: bugs are unavoidable. They can creep into the darkest recesses of programs and, when everything seems to be running the smoothest, come out and make garbage of your labors. Assuming you are not immune to programming errors, *Infraraid* can help you find, and trap, pesky bugs.

TRAPPING *INSECTUS INTHEPROGRAMMUS*

Recently, while working on a BASIC calculator simulation program, I came across an interesting (read frustrating) problem with the way BASIC 2.0 handles errors. Or should I say, the way BASIC *doesn't* handle errors. My problem arose when I tried to do arithmetic operations on numbers that exceeded the maximally allowed limit that is stored by BASIC, which is approximately 1.7E+38. Any numbers larger than this limit would result in an ?OVERFLOW ERROR. An example is 1E+20 multiplied by 1E+25, which should result in 1E+45; instead, my program would give the dreaded ?OVERFLOW ERROR and crash.

One way around this, I thought, would be to convert my BASIC program into machine language (and lose the weeks of programming already spent). Another idea was to somehow modify BASIC to either handle larger numbers or, alternatively, trap the error within the program. My final solution was a synthesis of all three ideas: using machine language, modifying BASIC somewhat, *and* trapping the error within the program.

(Some of you may see an easy solution to this problem: namely, adding the exponents and comparing the sum with 38, the upper limit. However, for a calculator simulation program to be user friendly, it must allow for any size and type of number. Therefore the program would have to normalize the mantissa of a large number before adding the exponents. This would be a slow process in BASIC, and henceforth I came up with a better solution.)

IDENTIFYING *INSECTUS INTHEPROGRAMMUS*

It is essential to understand how errors are generated to be able to properly handle them. There are two potential sources for bugs. The type are within the program itself and are created by the programmer. These errors can be broken into two groups: syntax (or format) errors and logic errors. Syntax errors occur when the programmer breaks the rules of BASIC text syntax. What happens is that the BASIC interpreter comes across a character it cannot handle. Perhaps it was looking for a number and came across an ASCII letter. Another possibility is a spelling mistake in a command word. Syntax errors always generate a BASIC error message and, therefore, are usually easy to find. (I did say usually, didn't I?)

Logic errors sometimes, but not always, generate a BASIC error message and therefore are harder to track down. The most common logic error is when the program jumps to another section and, say, into the middle of a FOR-NEXT loop. The NEXT is encountered without a pending FOR and the program crashes. The SYNTAX of the program is correct, but the LOGIC is at fault —even though the error message given, ?NEXT WITHOUT FOR ERROR, might indicate to you a program syntax error. A logic error that doesn't crash the program, such as jumping to a wrong line, just won't do what you want it to, period.

The second type of bugs are not created by the programmer, but must nevertheless be handled properly to assure a smooth-running program. These can be divided into externally generated errors and system limitations. Externally generated errors can come from various sources, such as a file being used by the program, or directly from the user. The best way to handle inputs from any source is always to expect the unexpected. Make sure all data received is what the program requires before using it, and reject any garbage that might come across. In BASIC, I always use GET rather than INPUT, and always store data in a string variable and convert to a numeric variable as needed.

The problem I encountered in my calculator program is a sample of a system limitation. As you might guess, system limitations are the hardest potential source of problems to account for. In my case, I couldn't do arithmetic operations on large numbers simply because BASIC wouldn't allow me to. But yet I had to allow for any number the user could possibly enter. That didn't mean I *had* to do the calculation, merely that I had to allow for the user to attempt to do the calculation without crashing my program, if this makes sense to you. (It did to me!) Another limitation on any computer system is the amount of RAM the programmer has available. On the C-64, large array tables can quickly use up available memory. In the case of a hardware system limitation, you can either upgrade or find an alternative software solution.

Now that we know all the potential sources of bugs that can infest, I will show you my solution for finding and trapping these critters.

TABLE 1
Code numbers for BASIC errors

ER%=	:BASIC error message	ER%=	:BASIC error message
1	TOO MANY FILES	16	OUT OF MEMORY
2	FILE OPEN	17	UNDEF'D STATEMENT
3	FILE NOT OPEN	18	BAD SUBSCRIPT
4	FILE NOT FOUND	19	REDIM'D ARRAY
5	DEVICE NOT PRESENT	20	DIVISION BY ZERO
6	NOT INPUT FILE	21	ILLEGAL DIRECT
7	NOT OUTPUT FILE	22	TYPE MISMATCH
8	MISSING FILENAME	23	STRING TOO LONG
9	ILLEGAL DEVICE NUMBER	24	FILE DATA
10	NEXT WITHOUT FOR	25	FORMULA TOO COMPLEX
11	SYNTAX	26	CAN'T CONTINUE
12	RETURN WITHOUT GOSUB	27	UNDEF'D FUNCTION
13	OUT OF DATA	28	VERIFY
14	ILLEGAL QUANTITY	29	LOAD
15	OVERFLOW	30	BREAK

THE WEAPON AGAINST *INSECTUS INTHEPROGRAMMUS*

Infraraid is technically a BASIC error-trapping wedge. It is a 495-byte machine language program stored starting at 50176. Note that the Commodore DOS wedge is stored starting at 51200, and it and *Infraraid* can coexist. Also note that 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

TABLE 2
Values for Different TR%
(Trap Variable) Configurations

High byte: 16 bit TR%: 0110 0000
Low byte: 0000 1111

Zeroes indicate unused bits. Values in these bits can be zero of one as they are not checked in this version of *Infraraid*.

SPECIFIC FUNCTIONS AND VALUES

Binary expanded	Decimal value	Function (to trap)
0000 0000 0000 0001	1	OVERFLOW ERROR
0000 0000 0000 0010	2	DIVISION BY ZERO ERROR
0000 0000 0000 0100	4	FILE NOT FOUND ERROR
0000 0000 0000 1000	8	DEVICE NOT PRESENT ERROR
0010 0000 0000 0000	32*256	STOP key interrupt
0100 0000 0000 0000	64*256	STORE/RESTORE key sequence

Note that any or all relevant bits may be set or clear at any time. Also, TR% may be changed anywhere in a BASIC program.

It must also be understood that TR% is only referenced by *Infraraid* when an error is generated. Therefore, if you wish to turn off the STOP key or the STOP/RESTORE key sequence at the beginning of a BASIC program, you must generate an error immediately after you set TR%. This next example will do this properly:

```
10 IF A=0 THEN A=1:LOAD "INFRARAID",8,1
15 SYS 50176
20 TR%=64*256+8+4+2+1:REM set STOP/REST
ORE and other trap bits
30 A=1/0:REM Generate error so INFRARAI
D can disable STOP/RESTORE
40 LN=your line number:REM put the line
number of your error routine here
50 EL%=LN+(LN>32767)*65536:REM This wil
l properly set EL% for any valid LN
```

cause the error routine vector at 768- 769 has to point to *Infraraid*, it is not relocatable.

To save *Infraraid* to tape or disk, type in the BASIC loader program and run it. I suggest entering the loader exactly as listed to allow my error checking routine to work properly. REM's may be deleted. As the loader runs it will print the line number of the current data line across the screen seven times, which corresponds to seven data numbers per line. If the loader comes across bad data (less than 0 or greater than 255) it will halt execution and tell you which line to examine. In this case, simply count the number of times the line number was printed and the next data item in that line is the problem. As an example, if the loader stops and prints the message BAD DATA IN LINE 310 and 310 was printed four times across the screen, the bad data is the fifth number in that line. The loader also calculates a checksum of all data (all the numbers are added together) and will tell you of a checksum error. In this case, first make sure the number in line 60 is 65731. PRINT CS will tell you the computed checksum. Then you will have to check all data individually. If BASIC crashes the loader with an ?OUT OF DATA ERROR, you missed some data somewhere. After the loader is all debugged and runs through it will ask you if you want the wedge saved to tape or disk.

Now that all prelim's are aside and you have a working version saved, let's find out what this software can do.

EXTERMINATING *INSECTUS INTHEPROGRAMMUS*

Infraraid is useful both in program development and as an error-trapping extension to BASIC 2.0. After loading, initiate *Infraraid* with SYS 50176. This causes several things to happen. First *Infraraid* sets the IERROR vector at 768-769 to point to itself and stores the original vector. Next a few variables must be created. ER% is used to store the code number BASIC uses to identify the error (see Table 1). TR% is a program-defined variable that specifies which errors to trap. It is also used to turn on or off the RUN/STOP key or the RUN/STOP RESTORE sequence (see Table 2). EL% is a third, program-defined variable used to specify which line in the BASIC program to jump to when a trapped error, referenced from TR%, occurs (see Table 3). Note that all three variables, whether used by a BASIC program or not, are created by *Infraraid*.

Probably the best way to learn how to use the Wedge is to do some examples. Load and activate *Infraraid* if not already done and, enter PRINT ER% in direct mode. If there is no current error condition you will get 128 as the result. (Although not technically an error, the READY prompt is vectored through the BASIC error routine and has the value of 128. This is the easiest way to verify that the Wedge is activated when in direct mode.) Cursor up to the READY prompt and hit RETURN to generate an ?OUT OF DATA ERROR, then enter PRINT ER%. ER% should be 13. I suggest experimenting in direct mode, generating various errors.

Infraraid alone, without setting any parameters in the variables, is a handy debugging tool. In direct mode, except for setting ER%, *Infraraid* is transparent, but in program mode when an error occurs it will clear the screen and display the line where execution stopped. Control is then passed to the BASIC error handler which prints the normal error message. In most cases, the last character read by BASIC will be displayed in white to highlight it from the rest of the line, which is displayed in the normal blue on blue.

To understand how this works we must know a little bit about how BASIC reads text. The CHARGET routine from 115 to 138 in page zero RAM is a short machine language routine that does the actual reading of program text and the vector TXTPTR at 122-123 is the address of the next text character. The important addresses to *Infraraid* are TXTPTR and another location, CURLIN, at 57-58, which is the current BASIC text line number. *Infraraid* uses CURLIN to find the text line and then prints the line character by character. If an address of a character matches that of TXTPTR, *Infraraid* changes the color of that character when printed. This is where the error occurred and BASIC stopped execution. Note that keywords are stored as one character, and if TXTPTR happens to point to it, the whole keyboard will be printed in white when expanded to ASCII characters.

Again, the best way to see this is to do some examples, such as

```
10 PRINT 10↑100
```

when run, *Infraraid* will clear the screen and print

```
10 PRINT 10↑100
?OVERFLOW ERROR IN 10
```

Continued on page 145

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

FREE OFFER FOR COMMODORE OWNERS.

www.commodore.ca

Owners of Commodore 64 or 128 computers are invited to go on-line with Viewtron® – and step into the future.

What does the future hold for you as a Viewtron subscriber?

Simply accept this *free* offer, and see for yourself. Put this unique, advanced generation, on-line service to the test – as Viewtron's guest.

Put Viewtron to the Test – Free.

Now see how easy it is to give your Commodore capabilities it's never had before.

Simply load the diskette Viewtron will send you. Then sit back and explore a huge database of useful time-saving, money-saving information packaged for you by America's leading communications company.

Suddenly, your computer is a powerful information center. For the first time, at almost any time, you have push-button access to:

Instant financial information – including one-button access to stock quotes.

On-line banking services that you can even use to pay your bills!

Instant airline schedules – complete with fare updates.

A complete encyclopedia that's constantly updated electronically.

Electronic shopping at discount prices on everything from computer software to new cars.

Instant sports reports – including late scores, standings, and statistics.

Instant news, national and international or even your own custom-designed news report.

Electronic messaging so you can talk on-line to fellow Viewtron subscribers for less than it costs to talk by phone.

Even electronic auctions where prices keep dropping instead of rising.

Special services exclusively for Commodore owners who'd like to "Ask the Experts" for answers to Commodore questions, order bargains on Commodore software and hardware. Plus, Viewtron is the only complete on-line service that *runs in color* on Commodore machines.

Viewtron makes all these services and more "yours to command" whenever you wish. And commanding them is so easy.

You issue these commands in plain English. In fact, part of Viewtron's sophistication is simply to make it easy for you to get exactly the information *you* want!

Of course, this speed and ease of use saves you time. And that, in turn, saves you money. But that's only part of the Viewtron "savings story."

Pay only for what you use.

For all its sophistication, Viewtron is the *least expensive* on-line service you can 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 the three major communications networks at no extra charge. (A local call for most people.)

Step into the future – free!

Taking your first step into the future with your Commodore is easy and affordable, too. All you need is the computer itself, a disk drive, a modem, and the Viewtron Starter Kit.

Order your Starter Kit now for just $9.95, and you'll get your first hour of service *free.*

If you haven't already invested in a modem, that's no problem. Viewtron can save you up to 50 percent on a modem! Mail the coupon today. Or call anytime toll-free. Then take a long step into the future.

To get both your Viewtron Starter Kit and 1 Free Hour of On-Line Service

CALL TOLL-FREE Or mail this coupon.

Call anytime 1 (800) 543-5500. Dept. 9016. Or mail coupon:
Viewdata Corporation of America, Inc., Dept. 9016. 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 Volks 6420 by Anchor Automation with FREE Viewtron Starter Kit. Only $49.95.

SATISFACTION GUARANTEED or your money back. All modem offers include cables (where necessary). 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 2-4 weeks for delivery. Shipping included. Florida, Illinois and New York residents will have the appropriate sales tax added to their order. **Offers expire February 28, 1986.**

Viewtron®

To order, mail the coupon or call anytime, 1(800) 543-5500. Ask for Dept. 9016.

Satisfaction Guaranteed. Viewtron® is a service of Viewdata Corporation of America, Inc., which is a wholly-owned subsidiary of Knight-Ridder Newspapers, Inc. *Viewtron is not available in Me., Ma., N.H., Vt. Prices may vary in some cities and outside the U.S.

Reader Service No. 114

www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** This page and page 113 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Read these pages **before** entering programs.

BUG REPELLENT

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

VIC 20 VERSION

By Michael Kleinert and David Barron

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

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

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

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

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

To pause the line codes listing, press SHIFT.

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

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

C-64 VERSION

By Michael Kleinert and David Barron

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

To pause the listing depress and hold the SHIFT key.

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

```
·5000 FORX=49152TO49488:READY:POKEX,Y:NEXT:END GJ
·5001 DATA32,161,192,165,43,133,251,165,44,133 DL
·5002 DATA252,160,0,132,254,32,37,193,234,177  DB
·5003 DATA251,208,3,76,138,192,230,251,208,2    OF
·5004 DATA230,252,76,43,192,76,73,78,69,32      KN
·5005 DATA35,32,0,169,35,160,192,32,30,171      CA
·5006 DATA160,0,177,251,170,230,251,208,2,230   CE
·5007 DATA252,177,251,32,205,189,169,58,32,210  JE
·5008 DATA255,169,0,133,253,230,254,32,37,193   CL
·5009 DATA234,165,253,160,0,76,13,193,133,253   NB
·5010 DATA177,251,208,237,165,253,41,240,74,74  MB
·5011 DATA74,74,24,105,65,32,210,255,165,253    EP
·5012 DATA41,15,24,105,65,32,210,255,169,13     GH
·5013 DATA32,220,192,230,63,208,2,230,64,230    AN
·5014 DATA251,208,2,230,252,76,11,192,169,153   NG
·5015 DATA160,192,32,30,171,166,63,165,64,76    BF
·5016 DATA231,192,96,76,73,78,69,83,58,32       EP
·5017 DATA0,169,247,160,192,32,30,171,169,3     PJ
·5018 DATA133,254,32,228,255,201,83,240,6,201   FK
·5019 DATA80,208,245,230,254,32,210,255,169,4   FL
·5020 DATA166,254,160,255,32,186,255,169,0,133  CL
·5021 DATA63,133,64,133,2,32,189,255,32,192     GC
·5022 DATA255,166,254,32,201,255,76,73,193,96   NN
·5023 DATA32,210,255,173,141,2,41,1,208,249     NH
·5024 DATA96,32,205,189,169,13,32,210,255,32    IM
·5025 DATA204,255,169,4,76,195,255,147,83,67    KC
·5026 DATA82,69,69,78,32,79,82,32,80,82         DC
·5027 DATA73,78,84,69,82,32,63,32,0,76          ML
·5028 DATA44,193,234,177,251,201,32,240,6,138   GN
·5029 DATA113,251,69,254,170,138,76,88,192,0    JI
·5030 DATA0,0,0,230,251,208,2,230,252,96        NA
·5031 DATA170,177,251,201,34,208,6,165,2,73     DN
·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)=ASCH(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
$:RETURN                                                              IM
·2000 REM GET FOUR DIGIT HEX                                          PC
·2010 PRINT:PRINTB$;:INPUT$                                           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 113 and 116 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

TALKING CLOCK
FROM PAGE 38

```
·1 PRINT"[CLEAR]"                                                    HH
·2 PRINT  SPC(7)"[9"[DOWN]"]C64TIME"                                 KN
·3 PRINT SPC(9)"[4"[DOWN]"]BY"                                       OB
·4 PRINT SPC(2)"[DOWN]ISAAC MICHALOWSKI"                             LI
·5 PRINT SPC(6)"[DOWN][DOWN]12/09/83"                                JC
·6 PRINT "[DOWN][DOWN] MOD. FOR THE C64/1
 28"                                                                 PC
·7 PRINT SPC(9)"[DOWN][DOWN]BY"                                      LO
·8 PRINT SPC(2)"[DOWN]MORTON KEVELSON"                               JF
·9 PRINT SPC(6)"[DOWN][DOWN] 5/17/85"                                KO
·10 FORX=1TO800:NEXTX                                                BP
·14 PRINT"[CLEAR]"                                                   HH
·15 DIMA$(6)                                                         HG
·20 PRINT "[4"[DOWN]"][7"[RIGHT]"][GREEN]
 [RVSON]TIME SET[RVSOFF]"                                            FJ
·25 PRINT"[DOWN][DOWN][4"[RIGHT]"]24 HOUR
  FORMAT"                                                            PN
·30 PRINT"[RIGHT][RIGHT][6"[DOWN]"][RIGHT
 ]ENTER TIME IN THE"                                                 CL
·35 PRINT"[3"[RIGHT]"]FOLLOWING MANNER"                              LM
·40 PRINT"[DOWN][DOWN][6"[RIGHT]"][YELLOW
 ]HH=HOURS"                                                          JE
·44 PRINT"[GREEN][6"[RIGHT]"]MM=MIUTES"                              HL
·45 PRINT"[6"[RIGHT]"][WHITE]SS=SECONDS"                             IE
·50 PRINT"[RIGHT][RIGHT][DOWN][DOWN][YELL
 OW]HH[GREEN]MM[WHITE]SS"                                            LF
·55 INPUTA$                                                          PH
·56 IFMID$(A$,7,7)=>"0"THEN3050                                      PE
·60 IF  A$>"235959"GOTO3000                                          JO
·65 TI$=A$                                                           NK
·70 PRINT SPC(8)"RUNNING"                                            CB
·120 SP(1)= 31 :SP(2)= 24                                            JN
·130 DRT=37136:DDR=37138:DFL=37149:DCB=37
 148: REM FOR VIC-20                                                 PD
·135 REM: DRT=56577:DDR=56579:DFL=56578:D
 CB=56576:REM FOR C-64/128                                           KG
·140 POKE DDR,127                                                    DA
·150 A=PEEK(DCB)AND 15:REM FOR VIC-20                                CB
·155 REM: A=PEEK(DFL)OR 4:REM FOR C-64/12
 8                                                                   LA
·160 POKE(DCB),160 OR A:REM FOR VIC-20                               KL
·165 REM: POKE(DFL),A:REM FOR C-64/128                               GC
·170 REM: GOSUB 3110:REM FOR C-64/128                                AN
·200 REM**STRIP TIME**                                               BO
·210 A$=LEFT$(TI$,2)                                                 DK
·220 A=VAL(A$)                                                       HB
·230 REM**STRIP MINUTES**                                            NM
·240 B$=MID$(TI$,3,2)                                                KJ
·250 B=VAL(B$)                                                       HH
·260 Z=0                                                             EH
·261 T=0                                                             DB
·265 IF A<21 THEN Z=1:GOTO330                                        DL
·266 T=0                                                             DB
·270 IF A=21 THEN A1=1:T=A:GOTO320                                   KD
·280 IF A=22 THEN A1=2:T=A:GOTO320                                   PP
·290 IF A=23 THEN A1=3:T=A:GOTO320                                   AP
·320 A=20                                                            FE
·330 REM**CHECK MINUTES**                                            MK
·340 IF B=0  GOTO 1000                                               PH
·350 IF B<10 THEN B1=B:B=0:GOTO1040                                  CM
·360 IF B=50 THEN B=23:GOTO1080                                      JF
·370 IF B>50 THEN B1=B-50:B=23:GOTO1040                              EJ
·380 IF B=40 THEN B=22:GOTO1080                                      JJ
·390 IF B>40 THEN B1=B-40:B=22:GOTO1040                              FG
·400 IF B=30 THEN B=21:GOTO1080                                      HB
·410 IF B>30 THEN B1=B-30:B=21:GOTO1040                              DD
·420 IF B>20 THEN B1=B-20:B=20:GOTO1040                              EA
·430 GOTO 1080                                                       FK
·1000 REM**TOP OF HOUR**                                             BD
·1010 IF T<21ANDA<21THEN SP(3)=A:SP(4)=29
 :SP(5)=33:N=3:GOTO2000                                              NM
·1020 SP(3)=A:SP(4)=A1:SP(5)=29:SP(6)=33:
 N=4:GOTO2000                                                        EI
·1030 GOTO 2000                                                      EP
·1040 REM**SOUND FORMAT MINUTES**                                    IJ
·1050 IF Z=1 THEN SP(3)=A:SP(4)=27:SP(5)=
 B:SP(6)=B1:SP(7)=28:N=5:GOTO2000                                    CL
·1060 SP(3)=A:SP(4)=A1:SP(5)=27:SP(6)=B
 :SP(7)=B1:SP(8)=28:N=6                                              BF
·1070 GOTO 2000                                                      EP
·1080 REM**SOUND FORMAT MINUTES10,20,30,4
 0,50 ONLY**                                                         GF
·1090 IF Z=1 THEN SP(3)=A:SP(4)=27:SP(5)=
 B:SP(6)=28:N=4:GOTO 2000                                            MH
·1100 SP(3)=A:SP(4)=A1:SP(5)=27:SP(6)=B:S
 P(7)=28:N=5:GOTO 2000                                               KC
·2000 REM**SPEAK!!**                                                 PC
·2010 FOR X=1 TO N+2                                                 IH
·2020 IF PEEK(DRT)>127 GOTO 2020                                     BN
·2030 POKE DRT,SP(X)                                                 BF
·2035 GOSUB3100                                                      FF
·2040 NEXT X                                                         NK
·2050 REM**TOP OF MINUTE**                                           KG
·2060 C$=RIGHT$(TI$,2)                                               DN
·2070 C=VAL(C$)                                                      HN
·2080 IF C=00 GOTO 200                                               AL
·2090 GOTO 2060                                                      FB
·3000 REM**ERROR MESSAGE**                                           PG
·3010 PRINT"[CLEAR]TIME SET IS GREATER[3"
  "]THAN 235959"                                                     JA
·3030 FORE=1TO3500:NEXTE                                             CN
·3040 PRINT"[CLEAR]":GOTO30                                          KB
·3050 PRINT"[CLEAR]ENTER ONLY 6 DIGITS":G
 OTO3020                                                             NH
·3100 REM POKE DCB,PEEK(DCB)AND251:REM FO
 R C-64/128                                                          AB
·3110 REM POKE DCB,PEEK(DCB)OR4:REM FOR C
 -64/128                                                             JJ
·3120 REM RETURN:REM FOR C-64/128                                    HG
```


www.commodore.ca

AHOY! BABBLER

FROM PAGE 38

VIC 20 DEMO

```
·10 REM **** AHOY! SPEAKS ****                    PN
·20 REM ****VIC-20 VERSION****                    DG
·30 REM BY **MORTON KEVELSON**                    KJ
·130 DRT=37136:DDR=37138:DFL=37149:DCB=37
 148:REM VIC-20 USER PORT                         LM
·140 POKE DDR,127:REM SET DDR FOR OUTPUT          ID
·150 A=PEEK(DCB)AND15                             OF
·160 POKE(DCB),160 OR A:REM SET BIT 2 FOR
  INPUT                                           OH
·165 GOSUB 3110                                   FP
·2000 REM**SPEAK!!**                              PC
·2010 FOR I=1 TO 17                               MC
·2015 READ X                                      OF
·2020 IF PEEK(DRT)>127 GOTO 2020                  NM
·2030 POKE DRT,X                                  BH
·2035 GOSUB3100                                   FF
·2040 NEXT I:END                                  IM
·3100 POKE DCB,PEEK(DCB)AND251:REM SET CO
 NTROL LINE LOW, UTTER ALLOPHONE                  NE
·3110 POKE DCB,PEEK(DCB)OR4:REM SET CONTR
 OL LINE HIGH, READY FOR NEXT                     JM
·3120 RETURN                                      IM
·3999 REM ALLOPHONE DATA                          GM
·4000 DATA 23,27,5,3                              FD
·4010 DATA 8,24,16,24,33,58,3                     AK
·4020 DATA 25,31,43,52,55,3                       MK
```

C-64/128 DEMO

```
·10 REM **** AHOY! SPEAKS ****                    DP
·20 REM ***C-64/128 VERSION***                    MJ
·30 REM **BY MORTON KEVELSON**                    MM
·130 DRT=56577:DDR=56579:DFL=56578:DCB=56
 576:REM C-64 USER PORT                           FA
·140 POKE DDR,127:REM SET DRT FOR OUTPUT          JA
·150 A=PEEK(DFL)OR 4                              ME
·160 POKE(DFL),A:REM SET BIT 2 FOR INPUT          PA
·165 GOSUB 3110                                   FE
·2000 REM**SPEAK!!**                              PC
·2010 FOR I=1 TO 17                               MK
·2015 READ X                                      OD
·2020 IF PEEK(DRT)>127 GOTO 2020                  BN
·2030 POKE DRT,X                                  BB
·2035 GOSUB3100                                   FF
·2040 NEXT I:END                                  IK
·3100 POKE DCB,PEEK(DCB)AND251:REM SET CO
 NTROL LINE LOW, UTTER ALLOPHONE                  DF
·3110 POKE DCB,PEEK(DCB)OR4:REM SET CONTR
 OL LINE HIGH, READY FOR NEXT                     HH
·3120 RETURN                                      IM
·3999 REM ALLOPHONE DATA                          EN
·4000 DATA 23,27,5,3                              EH
·4010 DATA 8,24,16,24,33,58,3                     PN
·4020 DATA 25,31,43,52,55,3                       LP
```

THE MAGICAL LINK

FROM PAGE 20

RS-232 RECEIVER

```
·30000 REM -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-      KC
·30001 REM    - RS-232 RECEIVER -                 KF
·30002 REM     RUPERT REPORT #25                  OM
·30003 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=      NC
·30004 REM RECEIVE AN ASCII PROGRAM FROM          FM
·30005 REM THE RS-232 PORT INTO MEMORY            LM
·30006 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=      NC
·30030 PRINT CHR$(147);                           JL
·30040 OPEN 2,2,0,CHR$(8)+CHR$(0)                 NG
·30050 REM - GET UP TO 80 CHARACTERS -            IN
·30060 GOSUB 30120                                JB
·30070 PRINT L$                                   AI
·30080 PRINT "RUN 3[4"0"]"                        PB
·30090 POKE 631,19 : POKE 632,13 : POKE 6
 33,13 : POKE 198,3 :REM FILL KBD BUFFER          JH
·30100 CLOSE 2 : END                              JB
·30110 REM == GET UP TO 80 CHARACTERS ==          MG
·30120 GET#2,C$ : IF C$="" THEN 30120             GN
·30130 IF C$=CHR$(13) THEN 30160                  OM
·30140 IF C$=CHR$(26) THEN PRINT#2 : CLOS
 E 2 : GOTO 30180                                 GG
·30150 L$=L$+C$ : GOTO 30120                      HN
·30160 L$=LEFT$(L$,79)                            EG
·30170 RETURN                                     IM
·30180 REM  DELETE LINES OF THIS PROGRAM          EK
·30190 FOR N=30000 TO 30200 STEP 10 :PRIN
 T N : NEXT                                       PC
·30200 PRINT"PRESS <HOME> AND 21 <RETURNS
 > TO DELETE THESE LINES";                        KA
```

ASCII TRANSMITTER

```
·1 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-       DD
·2 REM      - ASCII TRANSMITTER -                 OA
·3 REM        RUPERT REPORT #25                   OM
·4 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-       DD
·5 REM TRANSMIT ASCII FORM OF PROGRAM             IC
·6 REM OVER THE RS-232 CHANNEL                    PA
·7 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-       DD
·8 OPEN 2,2,0,CHR$(8) : CMD 2 : LIST              LO
·9 PRINT#2,CHR$(26):PRINT#2:CLOSE 2:END           FJ
·10 REM  -ADD THIS PROGRAM TO THE PROGRAM
  TO BE TRANSMITTED AND TYPE 'RUN 8'              ID
·11 REM  -WHEN THE LISTING IS DONE,               PG
·12 REM  -TYPE 'RUN 9' TO CLOSE THE FILE          DE
```

SEQUENTIAL TRANSMITTER

```
·10 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=       NC
·20 REM  - SEQUENTIAL TRANSMITTER -               NM
·30 REM     RUPERT REPORT #25                     OM
·40 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=       NC
·50 REM  TRANSMIT A SEQUENTIAL FILE               GC
·60 REM  TO THE RS-232 PORT                       LI
```


www.commodore.ca

```
•70 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=        NC
•80 PRINT CHR$(147)                         FG
•90 OPEN 2,2,0,CHR$(8)+CHR$(0)              NG
•100 PRINT"ENTER THE -FILENAME- OF THE      IP
•110 PRINT"SEQUENTIAL FILE TO BE SENT"      HK
•120 INPUT F$                               BF
•130 OPEN 8,8,8,F$+",SEQ,R"                 BO
•140 GET#8, C$ : IF C$="" THEN 140          HG
•150 SS=ST                                  AJ
•160 PRINT#2,C$;                            KC
•170 IF ASC(C$)<32 OR ASC(C$)>127 THEN C$
="*"                                        BK
•180 PRINT C$;                              GA
•190 IF SS=0 THEN 140 :REM NOT EOF          MK
•200 PRINT#2,CHR$(26) : PRINT#2 : CLOSE 2
 : CLOSE 8                                  NG
```

SEQUENTIAL RECEIVER

```
•10 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=        NC
•20 REM  - SEQUENTIAL RECEIVER -            BL
•30 REM      RUPERT REPORT #25              OM
•40 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=        NC
•50 REM  RECEIVE A SEQUENTIAL FILE          MH
•60 REM  FROM THE RS-232 PORT               DK
•70 REM =-=-=-=-=-=-=-=-=-=-=-=-=-=        NC
•80 PRINT CHR$(147)                         FG
•90 OPEN 2,2,0,CHR$(8)+CHR$(0)              NG
•100 GET#2,J$ :IF (ST AND 8)=0 THEN  100
 : REM CLEAR THE RECEIVE BUFFER             HP
•110 PRINT"SEND FILE TO  (1) PRINTER, (2)
 DISK, OR (3) SCREEN" : INPUT A$            LD
•120 N=VAL(A$)                              JO
•130 ON N GOTO 150,160,180                  JE
•140 GOTO 110 : REM  TRY AGAIN              HN
•150 OPEN 1,4,4 : GOTO 190 :REM PRINTER     NB
•160 INPUT"SAVE WITH WHAT FILENAME";F$      LF
•170 OPEN 1,8,8,F$+",S,W" : GOTO 190        MP
•180 OPEN 1,3    :REM SCREEN                LH
•190 REM - RECEIVE DATA -                   DA
•200 GET#2,C$ : IF C$="" THEN 200           OK
•210 PRINT#1, C$;                           KD
•220 IF C$<>CHR$(26) THEN 200               PD
•230 PRINT#1 : CLOSE 1                      NE
•240 CLOSE 2 : END                          JB
```

THE MARTIAN MONSTERS FROM PAGE 72

```
•2 REM THE MARTIAN MONSTERS                 OG
•3 REM BY J.C.HILTY                         LE
•5 POKE 52,48:POKE 56,48                    IC
•10 S=54272:FORL=STOS+24:POKEL,0:NEXT       JM
•40 V=53248                                 AD
•200 PRINT"[CLEAR]":POKE 53280,0:POKE 532
81,0                                        OD
•220 PRINT"[HOME][CYAN][3"[DOWN]"]T H E[3
" "]M A R T I A N[3" "]M O N S T E R S"     OG
•230 PRINT                                  JJ
•240 PRINT TAB(14)"[RED][RVSON] [RVSOFF][
6" "][RVSON] "                              OK
•250 PRINT TAB(15)"[RVSON] [RVSOFF][4" "]
[RVSON] "                                   KO
•260 PRINT TAB(9)"[BLUE][RVSON] [RVSOFF]
[RVSON] [RVSOFF][4" "][RED][RVSON] [RVSO
FF]  [RVSON] "                              JA
•270 PRINT TAB(9)"[BLUE][RVSON][3" "][RVS
OFF][3" "][GREEN][RVSON][6" "][RVSOFF][5
" "][BLUE][RVSON]  "                        LN
•280 PRINT TAB(10)"[BLUE][RVSON] [RVSOFF]
[3" "][GREEN][RVSON][8" "][RVSOFF][4" "]
[BLUE][RVSON] "                             NO
•290 PRINT TAB(10)"[BLUE][RVSON][3" "][GR
EEN][4" "][RVSOFF] [RVSON] [RVSOFF] [RVS
ON][3" "][BLUE][4" "]"                      NG
•300 PRINT TAB(13)"[GREEN][RVSON][10" "][
RVSOFF][3" "][BLUE][RVSON]  "               KM
•310 PRINT TAB(14)"[GREEN][RVSON][8" "]"    FA
•320 PRINT TAB(15)"[GREEN][RVSON][6" "]"    BK
•330 PRINT TAB(16)"[BLUE][RVSON] [RVSOFF]
[3" "][RVSON] "                             IO
•340 PRINT TAB(16)"[RVSON] [RVSOFF][3" "]
[RVSON] "                                   GO
•350 PRINT TAB(15)"[RVSON][3" "][RVSOFF]
[RVSON][3" "]"                              KK
•360 PRINT                                  JJ
•370 PRINT TAB(2)"[CYAN][3"."]SPACESHIPS
ARE OUR FAVORITE FOOD[3"."]"                CJ
•380 PRINT                                  JJ
•390 PRINT TAB(4)"[3"."]AND WE ARE GETTIN
G HUNGRY[3"."]"                             PL
•405 POKE S+24,15:POKE S+1,15:POKE S+5,16
:POKE S+6,240:POKE S+4,21                   HG
•420 FOR T=1 TO 3                           KB
•425 R=10                                   GK
•430 POKE S+15,R                            HB
•450 R=R+1:IF R=200 THEN 470                MH
•460 GOTO 430                               CH
•470 NEXT T                                 NG
•480 FOR I=0 TO 24:POKE S+I,0:NEXT          DI
•520 PRINT"[CLEAR]":PRINT"[4"[DOWN]"]"TAB
(4)"PLEASE WAIT FOR ONE MOMENT[3"."]"       HL
•521 PRINT:PRINT"..WHILE WE WORK UP AN AP
PETITE FOR YOU.."                           JJ
•525 FOR X=50880 TO 51116:READ A:POKE X,A
:NEXT                                       ND
•530 FOR X=49152 TO 49528:READ A:POKE X,A
:NEXT                                       LP
•535 FOR X=12288 TO 12671:READ A:POKE X,A
:NEXT                                       JI
•540 POKE 49522,1:POKE 49523,0:POKE 49524
,30:POKE 49525,0:POKE 49526,23              JA
```


www.commodore.ca

```
·554 PRINT"[CLEAR]"                                          HH
·555 FOR P=0 TO 18                                           MB
·560 T$="":FOR I=0 TO 30:T=32                                CO
·570 IF INT(RND(2)*7)<1 THEN T=46:IF INT(                    OG
 RND(2)*5)<1 THEN T=42                                       JJ
·580 T$=T$+CHR$(T):NEXT I                                    DD
·590 T$=T$+CHR$(170)                                         LN
·600 PRINT T$                                                OK
·610 NEXT P                                                  AN
·620 PRINT"[RED][RVSON]  [c *][RVSOFF][9"                    BA
 "][RVSON][sEP] [c *][RVSOFF] [RVSON][sE                     KO
 P] [c *][RVSOFF][11" "][RVSON][sEP]"                        NC
·630 PRINT"[RVSON][4" "][c *][RVSOFF]  [R                    JA
 VSON][sEP] [c *][sEP][11" "][c *][RVSOFF                    NN
 ] [RVSON][sEP] [c *][RVSOFF] [RVSON][sEP                    LN
 ]  "                                                        GG
·640 PRINT"[RVSON][31" "]"                                   HG
·650 PRINT"[RVSON][31" "]"                                   HG
·652 PRINT"[HOME]":PRINT TAB(34)"[GREEN]T                    NO
 HE"                                                         AG
·653 PRINT TAB(32)"MARTIAN"                                  BC
·654 PRINT TAB(32)"MONSTERS"                                 IK
·655 PRINT"[4"[DOWN]"]":PRINTTAB(33)"SCOR                    NG
 E"                                                          KH
·656 PRINT:PRINT:PRINT TAB(33)"SHIPS"                        FH
·660 PRINT:PRINT:PRINT TAB(33)"MISSED":PR                    BK
 INT TAB(33)"SHOTS"                                          BE
·662 SC=0:SH=5:MS=0                                          NO
·665 PRINT"[HOME]":PRINT"[13"[DOWN]"]"TAB                    IO
 (34)SH                                                      OL
·666 PRINT"[HOME]":PRINT"[10"[DOWN]"]"TAB                    GO
 (34)SC                                                      EK
·667 PRINT"[HOME]":PRINT"[17"[DOWN]"]"TAB                    KK
 (34)MS                                                      CG
·669 REM PREPARE SPRITES                                     JJ
·672 POKE 2040,193:POKE 2041,194                             KJ
·674 FOR T=2042 TO 2046                                      KP
·675 POKE T,192                                              PL
·676 NEXT T                                                  NK
·680 POKE V+39,1:POKE V+40,10:POKE V+41,5                    LL
 :POKE V+42,4                                                HG
·685 POKE V+28,1:POKE V+37,6:POKE V+38,2                     NG
·690 POKE V+43,7:POKE V+44,8:POKE V+45,6                     KB
·700 X0=140:Y0=160                                           GK
·710 POKE V+0,X0:POKE V+1,Y0                                 DD
·720 POKE V+4,35:POKE V+5,60                                 HB
·730 POKE V+6,85:POKE V+7,60                                 AC
·740 POKE V+8,135:POKE V+9,60                                MH
·750 POKE V+10,185:POKE V+11,60                              LL
·760 POKE V+12,235:POKE V+13,60                              CH
·765 POKE V+21,125                                           OJ
·770 POKE 50437,0:POKE 50438,255                             NG
·780 POKE 50439,0:POKE 50440,2                               CL
·790 POKE 50441,0:POKE 50442,254                             DI
·800 POKE 50443,0:POKE 50444,1                               BG
·810 POKE 50445,0:POKE 50446,253                             OB
·815 POKE 50435,0:POKE 50436,255                             HL
·820 POKE 50432,124                                          LE
·830 SYS 51104                                               DP
·840 REM MAIN LOOP                                           JL
·850 POKE V+0,X0:POKE V+1,Y0                                 JJ
·860 X0=X0+5:IF X0>250 THEN X0=30                            HF
·870 JY=PEEK(56321)AND15                                     LO
·880 IF JY=13THENY0=Y0+4:IF Y0>205 THEN Y                    ND
 0=205                                                       JA
·890 IF JY=14 THEN Y0=Y0-4:IF Y0<70 THEN                     FM
 Y0=70                                                       NN
·900 FB=-((PEEK(56321)AND16)=0):IF FB=1 T                    CC
 HEN 2000                                                    PB
·910 W=PEEK(V+30)                                            LP
·920 IF W=5 THEN Q=2042:GOTO 3000                            HN
·930 IF W=9 THEN Q=2043:GOTO 3000                            LB
·940 IF W=17 THEN Q=2044:GOTO 3000                           OD
·950 IF W=33 THEN Q=2045:GOTO 3000                           DI
·960 IF W=65 THEN Q=2046:GOTO 3000                           AA
·965 SYS 49152                                               LC
·970 GOTO 850                                                PA
·2000 REM FIRE LASER                                         LE
·2005 POKE 50432,0                                           ED
·2010 POKE V+2,X0:POKE V+3,Y0-12                             JA
·2015 POKE V+21,127                                          DB
·2020 POKE 50432,2                                           PF
·2030 IF PEEK(V+3)<55 THEN 2500                              GI
·2035 W1=PEEK(V+30)                                          KF
·2040 IF W1=6 THEN Q=2042:GOTO 2600                          CN
·2050 IF W1=10 THEN Q=2043:GOTO 2600                         DB
·2060 IF W1=18 THEN Q=2044:GOTO 2600                         ME
·2070 IF W1=34 THEN Q=2045:GOTO 2600                         NK
·2080 IF W1=66 THEN Q=2046:GOTO 2600                         ED
·2090 GOTO 2030                                              IM
·2499 REM MISSED SHOT                                        DD
·2500 MS=MS+1                                                KB
·2505 PRINT"[HOME]":PRINT"[17"[DOWN]"]"TA                    FE
 B(34)MS                                                     JJ
·2510 POKE V+21,125                                          MA
·2530 POKE V+23,124:POKE V+29,124                            CG
·2531 P=195                                                  LO
·2532 FOR T=1 TO 8                                           JL
·2533 FOR Z=2042 TO 2046                                     KH
·2534 POKE Z,P                                               KE
·2535 NEXT Z                                                 NA
·2536 P=P+1:IF P=197 THEN P=195                              HA
·2537 POKE S+5,9:POKE S+6,0:POKE S+24,15                     NM
·2538 POKE S+1,80:POKE S,15:POKE S+4,33                      OB
·2539 FOR E=0 TO 50:NEXT E                                   II
·2540 POKE S+4,32                                            OF
·2541 NEXT T                                                 LE
·2542 W1=PEEK(V+30)                                          FM
·2543 FOR Z=2042 TO 2046:POKE Z,192:NEXT                     NG
·2544 POKE V+23,0:POKE V+29,0                                ME
·2545 POKE 50432,124                                         JC
·2548 IF MS=5 THEN 7000                                      CH
·2550 GOTO 850                                               GO
·2599 REM LASER HITS MARTIAN                                 JA
·2600 SC=SC+100                                              DB
·2602 PRINT"[HOME]":PRINT"[10"[DOWN]"]"TA                    GI
```


www.commodore.ca

```
B(33)SC                                   EL
·2605 POKE V+21,125                       LO
·2606 POKE Q,197                          LB
·2608 GOSUB 3500                          FB
·2630 FOR I=0 TO 24:POKE S+I,0:NEXT       DI
·2631 POKE S+24,15:POKE S+1,15:POKE S+5,1
6                                         KG
·2632 POKE S+6,240:POKE S+4,21            DK
·2633 R=80                                HF
·2634 POKE S+15,R                         HB
·2635 R=R+1:IF R=160 THEN 2640            PK
·2636 GOTO 2634                           FG
·2640 FOR I=0 TO 24:POKE S+I,0:NEXT       DI
·2642 POKE Q,192                          KK
·2650 W1=PEEK(V+30)                       ME
·2655 POKE 50432,124                      GO
·2660 GOTO 850                            DB
·3000 REM MARTIAN EATS SPACESHIP          EA
·3005 POKE 50432,0                        AG
·3010 IF Q=2042 THEN POKE V+29,4:POKE V+2
3,4                                       OG
·3020 IF Q=2043 THEN POKE V+29,8:POKE V+2
3,8                                       EF
·3030 IF Q=2044 THEN POKE V+29,16:POKE V+
23,16                                     AB
·3040 IF Q=2045 THEN POKE V+29,32:POKE V+
23,32                                     CC
·3050 IF Q=2046 THEN POKE V+29,64:POKE V+
23,64                                     HP
·3060 POKE 2040,197                       DK
·3070 GOSUB 3500                          FB
·3080 POKE V+21,124                       LP
·3090 P=195                               KH
·3100 FOR I=0 TO 24:POKE S+I,0:NEXT       DI
·3110 POKE S+5,9:POKE S+6,0:POKE S+24,15  II
·3120 FOR T=1 TO 8                        KE
·3130 POKE Q,P                            GH
·3140 P=P+1:IF P=197 THEN P=195           OB
·3150 POKE S+1,80:POKE S,15:POKE S+4,33   OF
·3160 FOR E=0 TO 50:NEXT E                LE
·3170 POKE S+4,32                         FM
·3180 NEXT T                              NG
·3190 SH=SH-1                             ML
·3195 PRINT"[HOME]":PRINT"[13"[DOWN]"]"TA
B(34)SH                                   OL
·3200 POKE 2040,193                       DG
·3210 POKE Q,192                          KK
·3212 POKE V+29,0:POKE V+23,0             CH
·3215 X0=160:Y0=140                       OB
·3220 POKE V+0,X0:POKE V+1,Y0             CL
·3230 POKE V+21,125                       LO
·3240 POKE V+8,135:POKE V+9,60            DP
·3260 POKE 50432,124                      GO
·3265 W=PEEK(V+30)                        HN
·3268 IF SH=0 THEN 7000                   IE
·3270 GOTO 965                            DC
·3499 REM EXPLOSION SOUND                 OM
·3500 FOR I=0 TO 24:POKE S+I,0:NEXT       DI
·3510 POKE S+24,15:POKE S+12,160:POKE S+1
3,252                                     FC
·3520 POKE S+8,80:POKE S+7,40:POKE S+11,1
29                                        GN
·3530 FOR T=0 TO 100:NEXT                 LC
·3540 POKE S+11,128                       OB
·3550 RETURN                              IM
·4000 REM ML SPRITE ROUTINE               BE
·4010 DATA 169,255,45,0,198,240,16,169,0,
141,0,198,162,21,189,0                    MN
·4020 DATA 197,157,0,198,202,208,247,162,
1,169,1,141,80,197,173,80                 KK
·4030 DATA 197,45,0,197,240,3,76,243,198,
232,232,14,80,197,208,238                 OJ
·4040 DATA 76,49,234,169,0,29,0,197,208,3
,76,97,199,169,128,61                     CC
·4050 DATA 0,197,240,48,254,0,198,208,40,
222,255,207,76,144,199,80                 LE
·4060 DATA 197,45,16,208,208,12,173,16,20
8,13,80,197,141,16,208,76                 NB
·4070 DATA 43,199,173,16,208,77,80,197,14
1,16,208,189,0,197,157,0                  OO
·4080 DATA 198,76,97,199,222,0,198,208,40
,254,255,207,208,29,173,80                AB
·4090 DATA 197,45,16,208,208,12,173,16,20
8,13,80,197,141,16,208,76                 NB
·4100 DATA 91,199,173,16,208,77,80,197,14
1,16,208,189,0,197,157,0                  CD
·4110 DATA 198,169,0,232,29,0,197,208,3,7
6,140,199,169,128,61,0                    PO
·4120 DATA 197,240,11,254,0,198,208,20,22
2,255,207,76,134,199,222,0                KC
·4130 DATA 198,208,9,254,255,207,189,0,19
7,157,0,198,202,76,233,198                FP
·4140 DATA 169,255,221,255,207,240,3,76,4
3,199,173,80,197,76,17,199                BD
·4150 DATA 120,169,192,141,20,3,169,198,1
41,21,3,88,96                             JL
·5000 REM SCROLL DATA                     GJ
·5010 DATA 174,114,193,224,3,144,3,76,117
,192,188,114,193,140,121,193,174          GI
·5020 DATA 118,193,232,202,32,30,193,172,
121,193,173,119,193,201,2,208,10          EL
·5030 DATA 169,32,72,173,33,208,72,76,50,
192,177,90,72,177,92,72,204               CN
·5040 DATA 116,193,240,20,200,177,90,72,1
77,92,136,145,92,104,145,90,200           CI
·5050 DATA 204,116,193,208,238,240,18,136
,177,90,72,177,92,200,145,92,104          AK
·5060 DATA 145,90,136,204,115,193,208,238
,173,119,193,201,0,208,5,104,104          KD
·5070 DATA 76,111,192,104,145,92,104,145,
90,236,117,193,208,160,96,172,116         OJ
·5080 DATA 193,200,189,114,193,170,32,30,
193,173,120,193,201,2,208,19,136          OP
·5090 DATA 169,32,153,122,193,173,33,208,
153,162,193,204,115,193,208,239,240       KB
·5100 DATA16,136,177,90,153,122,193,177,9
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 113 and 116 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
FC
GN  2,153,162,193,204,115,193,208,240                          CE
LC ·5110 DATA 236,117,193,240,37,202,32,30,1
OB  93,172,116,193,200,136,177,90,72                           MG
IM ·5120 DATA 177,92,32,48,193,145,92,104,14
BE  5,90,32,56,193,204,115,193,208                             HJ
   ·5130 DATA 234,236,117,193,208,221,240,46
MN  ,202,206,118,193,232,32,30,193,172                         GF
   ·5140 DATA 116,193,200,136,32,48,193,177,
KK  90,72,177,92,32,56,193,145,92                              IC
   ·5150 DATA 104,145,90,204,115,193,208,234
OJ  ,236,118,193,208,221,238,118,193,232                       AL
   ·5160 DATA 32,30,193,173,120,193,201,0,24
CC  0,20,172,115,193,136,200,185,162                           DK
   ·5170 DATA 193,145,92,185,122,193,145,90,
LE  204,116,193,208,240,96,189,89,193                          EP
   ·5180 DATA 133,91,24,105,212,133,93,189,6
NB  4,193,133,90,133,92,96,72,152                              IA
   ·5190 DATA 24,105,40,168,104,96,72,152,56
OO  ,233,40,168,104,96,0,40,80                                 LL
   ·5200 DATA 120,160,200,240,24,64,104,144,
AB  184,224,8,48,88,128,168,208,248                            LK
   ·5210 DATA 32,72,112,152,192,4,4,4,4,4,4,
NB  4,5,5,5,5,5                                                JG
   ·5220 DATA 5,6,6,6,6,6,6,6,7,7,7,7,7,3,0,
CD  4,0                                                        LE
   ·5230 DATA 4,1,1                                            HP
   ·5232 REM MONSTER DATA                                      OM
   ·5235 DATA 6,0,96,3,0,192,1,129,128,0,195
KC  ,0,0,126,0,0,255,0                                         OD
   ·5240 DATA 1,255,128,3,255,192,227,255,19
FP  9,63,221                                                   HF
   ·5244 DATA 252,227,255,199,3,255,192,1,25
BD  5,128                                                      HI
   ·5248 DATA 0,255,0,0,66,0,0,66,0,0,231,0,
JL  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                              GI
GJ ·5250 REM ROCKET MULTICOLOR DATA                            KD
   ·5260 DATA 0,32,0,0,32,0,0,32,0,0,168,0,0
GI  ,168,0,1,169,0,5,169,64                                    HI
   ·5270 DATA 21,169,80,85,169,84,0,168,0,0,
EL  168,0,0,168,0,0,168,0,3,255,0                              FB
   ·5280 DATA 15,255,192,63,255,240,15,255,1
CN  92,0,48,0,0,0,0,0,0,0,0,0,0,0                              AP
   ·5290 REM LASER DATA                                        CE
CI ·5300 DATA 0,56,0,0,56,0,0,56,0,0,56,0,0,
    56,0,0,56,0,0,56,0,0,56,0                                  PG
AK ·5310 DATA 0,56,0,0,56,0,0,56,0,0,56,0,0,
    56,0                                                       MJ
KD ·5320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0                                      NH
OJ ·5330 REM GOBBLE DATA                                       GP
   ·5340 DATA 3,0,0,1,128,0,0,192,0,0,60,0,0
OP  ,126,0,0,255,0,1,247,224                                   OC
   ·5350 DATA 1,255,224,1,252,0,1,248,0,0,24
KB  8,0,0,120,0,0,60,0                                         IC
   ·5360 DATA 0,39,192,0,35,192,0,32,0,0,112
    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0                               EP
   ·5370 REM MORE GOBBLE DATA                                  JC
   ·5380 DATA 3,0,0,1,128,0,0,192,0,0,60,0,0
    ,126,0,0,255,0,1,247,128                                   PJ
   ·5390 DATA 1,255,192,1,255,192,1,255,192,
    0,255,192,0,127,128,0,62,0                                 FF
   ·5400 DATA 0,32,0,0,32,0,0,32,0,0,112,0,0
    ,0,0,0,0,0,0,0,0,0,0,0,0                                   AL
   ·5410 REM EXPLOSION                                         FG
   ·5420 DATA 0,0,0,76,1,0,0,128,4,32,4,32,0
    ,48,0,8,0,0,0,32,48,80,0,130                               JG
   ·5430 DATA 3,12,0,0,0,4,32,0,128,0,36,0,0
    ,0,192,201,0,4,0,0,0,19,32,192                             DO
   ·5440 DATA 0,4,0,32,0,1,4,192,0,0,4,8,0,0
    ,0,0                                                       NC
   ·6999 REM GAME OVER-PLAY AGAIN OPTION                       DB
   ·7000 PRINT"[CLEAR]"                                        HH
   ·7010 POKE V+21,124                                         LP
   ·7020 PRINT"[8"[DOWN]"]"TAB(6)"YUMMY[3"."
    ]"                                                         PJ
   ·7030 PRINT TAB(14)"YUMMY[3"."]"                            FO
   ·7040 PRINT TAB(22)"YUMMY[3"."]"                            DL
   ·7050 PRINT:PRINT TAB(4)"WE GOT SPACESHIP
    S IN OUR TUMMY[4"!"]"                                      EL
   ·7060 FOR I=0 TO 24:POKE S+I,0:NEXT                         DI
   ·7070 POKE S+24,15:POKE S+1,15:POKE S+5,1
    6                                                          KG
   ·7080 POKE S+6,240:POKE S+4,21                              DK
   ·7090 FOR T=1 TO 2                                          JO
   ·7100 R=10                                                  GK
   ·7110 POKE S+15,R                                           HB
   ·7120 R=R+1                                                 DB
   ·7130 IF R=200 THEN 7150                                    KI
   ·7140 GOTO 7110                                             FE
   ·7150 NEXT T                                                NG
   ·7160 FOR I=0 TO 24:POKE S+I,0:NEXT                         DI
   ·7170 PRINT"[CLEAR]"                                        HH
   ·7190 PRINT "[10"[DOWN]"]"TAB(9)"S C O R
    E[5" "]"SC                                                 MN
   ·7200 PRINT:PRINT                                           GJ
   ·7210 PRINT TAB(6)"PLAY AGAIN?[6" "]Y OR
    N"                                                         DN
   ·7215 GET JUNK$:IF JUNK$<>"" THEN 7215                      EC
   ·7220 GET A$:IF A$="" THEN 7220                             NN
   ·7230 IF A$="Y" THEN 7250                                   NH
   ·7240 END                                                   IC
   ·7250 POKE V+21,0:POKE 50432,0                              HO
   ·7260 PRINT"[CLEAR]"                                        HH
   ·7265 W=PEEK(V+30)                                          HN
   ·7270 RUN                                                   II
```

SCRATCH PAD
FROM PAGE 91

MAIN PROGRAM

```
·10 FORAD=49152TO49418:READD:POKEAD,D:NEX
```


www.commodore.ca

To enter STREAMER FONT... you must use our *Flankspeed* machine language entry program. Read the introduction to *Flankspeed* on page 116 of this magazine.

```
 TAD                                                    MA
·20 DATA160,0,177,251,145,253                           BP
·30 DATA169,14,145,30,200,240                           KH
·40 DATA13,192,232,208,241,166                          NF
·50 DATA254,236,127,192,208,234                         EL
·60 DATA76,49,234,230,252,230                           MF
·70 DATA254,230,31,76,2,192                             FD
·80 DATA120,169,49,141,20,3                             PP
·90 DATA169,192,141,21,3,88                             CM
·100 DATA96,32,159,255,166,198                          GL
·110 DATA208,3,76,49,234,202                            GC
·120 DATA189,119,2,201,133,208                          KI
·130 DATA28,169,0,133,251,133                           EB
·140 DATA30,133,253,169,194,133                         AN
·150 DATA252,169,216,133,31,169                         BP
·160 DATA4,133,254,169,7,141                            HB
·170 DATA127,192,76,0,192,201                           MP
·180 DATA137,208,29,169,0,133                           DO
·190 DATA251,133,253,133,30,169                         BF
·200 DATA4,133,252,169,194,133                          NK
·210 DATA254,169,197,141,127,192                        JG
·220 DATA169,216,133,31,76,0                            IN
·230 DATA192,7,201,134,208,30                           PI
·240 DATA169,0,133,30,133,253                           IC
·250 DATA169,4,133,254,169,232                          MG
·260 DATA133,251,169,197,133,252                        DI
·270 DATA169,7,141,127,192,169                          LA
·280 DATA216,133,31,76,0,192                            AL
·290 DATA201,138,208,30,169,0                           LD
·300 DATA133,251,133,30,169,4                           HI
·310 DATA133,252,169,232,133,253                        MK
·320 DATA169,197,133,254,169,216                        IK
·330 DATA133,31,169,200,141,127                         AJ
·340 DATA192,76,0,192,201,135                           KE
·350 DATA208,30,169,208,133,251                         MA
·360 DATA169,201,133,252,169,0                          KA
·370 DATA133,253,133,30,169,216                         MK
·380 DATA133,31,169,4,133,254                           LK
·390 DATA169,7,141,127,192,76                           IL
·400 DATA0,192,201,139,208,30                           KO
·410 DATA169,0,133,251,133,30                           DO
·420 DATA169,4,133,252,169,216                          LG
·430 DATA133,31,169,208,133,253                         AK
·440 DATA169,201,133,254,169,204                        KB
·450 DATA141,127,192,76,0,192                           GG
·460 DATA76,49,234                                      FG
```

LOAD/SAVE

```
·10 PRINT"[CLEAR][4"[DOWN]"]"                           LH
·20 INPUT"LOAD OR SAVE (L/S)";A$                        IC
·30 IFA$="L"GOTO100                                     FF
·40 IFA$="S"GOTO200                                     GH
·50 GOTO10                                              PH
·100 INPUT"TAPE OR DISC (T/D)";B$:IFB$<>"
 T"ANDB$<>"D"THEN100                                    LD
·110 INPUT"FILENAME";C$:IFB$="D"THENB=8:C
 $="0:"+C$:GOTO140                                      DG
·120 B=1                                                CO
·140 GOSUB300:POKE780,0:SYS65493:SYS49188               PL
·150 PRINT"[DOWN]RUN/STOP-RESTORE DE-ACTI
 VATES"                                                 MH
·160 PRINT"TO ACTIVATE: SYS49188":END                   OC
·200 PRINT"[3"[RIGHT]"]SAVE F-1 SCREEN[9"
  "][3"="]> 1                                           KN
·210 PRINT"[3"[RIGHT]"]SAVE F-1 AND F-2 S
 CREEN [3"="]> 2                                        DF
·220 PRINT"[3"[RIGHT]"]SAVE ALL THREE SCR
 EENS  [3"="]> 3                                        IB
·230 INPUT"CHOOSE BY NUMBER";A%:IFA%<1ORA
 %>3THEN230                                             NJ
·250 A=49664+1000*A%                                    JM
·260 INPUT"TAPE OR DISC (T/D)";B$:IFB$<>"
 T"ANDB$<>"D"THEN260                                    LN
·265 IFB$="T"THENB=1:GOTO280                            FH
·270 B=8                                                DH
·280 INPUT"FILENAME";C$:IFB=8THENC$="0:"+
 C$:OPEN15,8,15,"S"+C$:CLOSE15                          AH
·295 GOSUB300                                           CD
·297 POKE780,251:POKE251,0:POKE252,192                  EA
·298 POKE782,A/256:POKE781,A-PEEK(782)*25
 6:SYS65496:END                                         LL
·300 C=256*PEEK(54)+PEEK(53)-LEN(C$)                    EK
·310 POKE780,LEN(C$):POKE782,C/256:POKE78
 1,C-256*PEEK(782):SYS65469                             EO
·320 POKE780,1:POKE781,B:POKE782,1:SYS654
 66:RETURN                                              CK
```

STREAMER FONT
FROM PAGE 28

Beginning address in hex: 0801
Ending address in hex: 1840
SYS to start: 2061

MAIN PROGRAM

```
0801: 0B 08 0A 00 9E 32 30 36 55
0809: 31 00 00 00 AD AA 16 D0 79
0811: 08 20 91 0D A9 01 8D AA BA
0819: 16 A9 80 8D 8A 02 8D 91 92
0821: 02 A9 D8 85 FD A9 00 8D 60
0829: 15 D0 8D 7B 18 8D 76 18 4C
0831: A9 0B 8D 20 D0 8D 21 D0 E3
0839: A0 F0 A9 CF 99 FF 03 99 7A
0841: EF 04 99 DF 05 99 CF 06 23
0849: A9 00 99 FF D7 99 EF D8 C6
0851: 99 DF D9 99 CF DA 88 D0 42
0859: E1 A9 B9 85 FC A9 17 85 67
0861: FD 20 41 12 20 4A 12 A9 F8
0869: 0C 8D E7 DB A9 A0 8D E7 86
0871: 07 A0 00 B9 1D 18 99 80 22
0879: 3E C8 C0 20 D0 F5 A9 00 D1
0881: 99 80 3E C8 C0 40 D0 F8 6D
0889: A9 FA 8D FB 07 A9 08 8D FD
```


www.commodore.ca

```
0891: 17 D0 8D 1D D0 8D 27 D0 7A
0899: A9 07 8D 2A D0 A9 18 8D 22
08A1: 06 D0 A9 DC 8D 05 D0 8D EF
08A9: 07 D0 A9 19 8D 04 D0 A9 50
08B1: DD 8D 05 D0 A0 00 8C 29 49
08B9: D0 B9 E6 17 99 C0 3E C8 A3
08C1: C0 1D D0 F5 A9 00 99 C0 6A
08C9: 3E C8 C0 40 D0 F8 A9 18 5D
08D1: 8D 00 D0 A9 32 8D 01 D0 6B
08D9: A9 02 8D 98 16 A9 FB 8D F4
08E1: F8 07 A0 00 B9 03 18 99 F0
08E9: 00 3F C8 C0 1A D0 F5 A9 3D
08F1: 04 8D 99 16 A9 00 8D 10 7A
08F9: D0 99 00 3F C8 C0 40 D0 3E
0901: F8 A9 15 8D 02 D0 A9 F1 B4
0909: 8D 03 D0 A9 06 8D 28 D0 A0
0911: A9 FC 8D F9 07 20 25 12 9D
0919: CE 9B 16 D0 F8 A9 05 8D 9F
0921: 97 16 A9 03 8D 15 D0 20 0F
0929: 57 12 4C 0D 0B AD 00 DC 81
0931: 49 FF 8D 74 18 29 10 C9 97
0939: 10 F0 2C A9 00 8D 7A 18 30
0941: 20 E0 0A AD 74 18 29 03 B2
0949: C9 01 F0 24 C9 02 F0 23 09
0951: AD 74 18 29 0C C9 04 F0 7F
0959: 14 C9 08 F0 0D AD 74 18 77
0961: 29 03 D0 C9 4C 77 09 4C 41
0969: 9A 0C 4C F0 0B 4C 48 0C F8
0971: 4C 98 0B 4C BA 0B 20 E4 78
0979: FF C9 9D F0 74 C9 1D F0 1E
0981: 6A C9 85 90 04 C9 8D 90 B7
0989: 6E C9 5F F0 6D C9 13 F0 4D
0991: 75 C9 30 90 04 C9 3A 90 2A
0999: 70 C9 41 90 04 C9 5B 90 5F
09A1: 6B C9 C4 F0 6A C9 D2 F0 84
09A9: 54 C9 CD F0 53 C9 2D F0 C1
09B1: 4F C9 C6 F0 4E C9 5C F0 E7
09B9: 4A C9 CC F0 55 C9 D3 F0 6F
09C1: 54 C9 D6 F0 53 C9 21 F0 D6
09C9: 52 C9 22 F0 51 C9 23 F0 28
09D1: 50 C9 24 F0 4F C9 25 F0 30
09D9: 4E C9 D8 F0 4D C9 93 F0 57
09E1: 4C C9 14 F0 4B 20 07 15 84
09E9: 4C 2E 09 20 1E 13 4C 2E 39
09F1: 09 20 EC 12 4C 2E 09 4C E9
09F9: 3F 0A 4C 34 0A 4C 7C 16 AC
0A01: 4C 16 11 4C 51 11 4C B4 24
0A09: 12 4C 50 0B 4C F9 0A 4C 5F
0A11: 8D 0B 4C 60 0E 4C DF 0D 9D
0A19: 4C DD 15 4C 73 0A 4C 88 F6
0A21: 0A 4C 9F 0A 4C B4 0A 4C 78
0A29: CB 0A 4C 32 0B 4C 73 12 5A
0A31: 4C D3 12 AD 7B 18 49 01 EE
0A39: 8D 7B 18 4C 2E 09 C9 85 2D
0A41: F0 1B C9 89 F0 1A C9 86 FB
0A49: F0 19 C9 8A F0 18 C9 87 02
0A51: F0 17 C9 8B F0 16 C9 88 08
0A59: F0 15 4C 63 13 4C 71 13 F2
0A61: 4C 98 13 4C BF 13 4C F2 B7
0A69: 13 4C 23 0F 4C 8D 11 4C 32
0A71: 50 13 A9 01 8D AB 16 8D 5C
0A79: AC 16 A9 32 8D A8 16 A9 0E
0A81: 38 8D A9 16 4C 2E 09 A9 34
0A89: 02 8D AB 16 A9 01 8D AC BF
0A91: 16 A9 31 8D A8 16 A9 36 AE
0A99: 8D A9 16 4C 2E 09 A9 02 16
0AA1: 8D AB 16 8D AC 16 A9 31 1C
0AA9: 8D A8 16 A9 36 8D A9 16 23
0AB1: 4C 2E 09 A9 03 8D AB 16 31
0AB9: A9 02 8D AC 16 A9 30 8D 1D
0AC1: A8 16 A9 34 8D A9 16 4C F7
0AC9: 2E 09 A9 03 8D AB 16 8D 8A
0AD1: AC 16 A9 30 8D A8 16 A9 64
0AD9: 34 8D A9 16 4C 2E 09 AE 8D
0AE1: 98 16 AC 27 D0 8C 98 16 70
0AE9: 8E 27 D0 AE 99 16 AC 28 A3
0AF1: D0 8C 99 16 8E 28 D0 60 E6
0AF9: 38 E9 40 8D 76 18 38 E9 9A
0B01: 01 0A 0A 0A 69 15 8D 75 A1
0B09: 18 4C 17 0B A9 01 8D 76 3E
0B11: 18 A9 15 8D 75 18 A9 00 AC
0B19: 8D 7F 18 20 EC 12 A9 32 39
0B21: 8D 01 D0 A9 00 8D 10 D0 98
0B29: AD 75 18 8D 02 D0 4C 2E 3F
0B31: 09 20 44 E5 A9 00 8D 15 D0
0B39: D0 8D 8A 02 8D 91 02 A9 EE
0B41: 0E 8D 86 02 8D 20 D0 A9 8D
0B49: 06 8D 21 D0 4C 74 A4 C9 FD
0B51: 30 F0 2C 38 E9 12 8D 76 D6
0B59: 18 38 E9 1F 0A 0A 0A 69 3A
0B61: 05 8D 75 18 A9 00 8D 7F 38
0B69: 18 20 EC 12 A9 32 8D 01 0B
0B71: D0 A9 02 8D 10 D0 AD 75 7F
0B79: 18 8D 02 D0 4C 2E 09 A9 1F
0B81: 1E 8D 76 18 A9 FD 8D 75 66
0B89: 18 4C 17 0B AD 15 D0 49 EC
0B91: 0C 8D 15 D0 4C 2E 09 A9 3E
0B99: 08 8D 9B 16 20 25 12 CE 07
0BA1: 01 D0 CE 9B 16 AD 9B 16 53
0BA9: D0 F2 AD 01 D0 C9 2A D0 B1
0BB1: 05 A9 EA 8D 01 D0 4C 51 48
0BB9: 09 AD 01 D0 C9 EA F0 27 0F
0BC1: A9 08 8D 9B 16 20 25 12 0A
0BC9: EE 01 D0 CE 9B 16 AD 9B 54
0BD1: 16 D0 F2 AD 01 D0 C9 F2 E7
0BD9: D0 05 A9 32 8D 01 D0 A9 94
0BE1: 00 8D 1B D0 4C 51 09 A9 AB
0BE9: 01 8D 1B D0 4C C1 0B A9 27
0BF1: 08 8D 9B 16 20 25 12 EE 7F
0BF9: 00 D0 AD 00 D0 F0 13 C9 17
0C01: B4 F0 39 CE 9B 16 D0 EC 1E
0C09: AD 00 D0 C9 58 F0 0E 4C F4
0C11: 2E 09 AD 10 D0 49 01 8D AE
0C19: 10 D0 4C 04 0C AD 10 D0 E4
0C21: 29 01 F0 15 AD 10 D0 49 29
0C29: 01 8D 10 D0 A9 B8 8D 00 88
```


www.commodore.ca

```
0C31: D0 A9 01 8D 82 18 20 1E 13
0C39: 13 4C 2E 09 A9 01 8D 82 8A
0C41: 18 20 1E 13 4C 04 0C A9 B0
0C49: 08 8D 9B 16 20 25 12 CE B6
0C51: 00 D0 AD 00 D0 C9 FF F0 5B
0C59: 13 C9 B4 F0 31 CE 9B 16 8D
0C61: D0 EA AD 00 D0 C9 10 F0 66
0C69: 0E 4C 2E 09 AD 10 D0 49 D2
0C71: 01 8D 10 D0 4C 5E 0C AD 45
0C79: 10 D0 29 01 D0 0D A9 B0 BC
0C81: 8D 00 D0 A9 01 8D 82 18 B2
0C89: 20 EC 12 4C 2E 09 A9 01 D6
0C91: 8D 82 18 20 EC 12 4C 5E 83
0C99: 0C 20 3B 0D AD 7A 18 D0 1F
0CA1: 0C A9 01 8D 7A 18 A1 FC 17
0CA9: 49 04 8D 9A 16 AD 9A 16 93
0CB1: 81 FC AD 7B 18 D0 06 20 68
0CB9: 07 15 4C 41 09 AD 00 D0 EA
0CC1: F0 4E AD 10 D0 29 01 D0 8A
0CC9: 4C AD 00 D0 C9 64 90 12 65
0CD1: C9 B4 90 1C AD 7C 18 69 A8
0CD9: 1C 38 E5 FC 0A A8 C8 4C D8
0CE1: 33 0D AD 7C 18 69 09 38 0F
0CE9: E5 FC 0A A8 C8 4C 33 0D D4
0CF1: 38 A5 FC E9 09 38 ED 7C 62
0CF9: 18 0A 8D 78 18 38 A5 FC 15
0D01: ED 78 18 85 FC A5 FD E9 8F
0D09: 00 85 FD A0 01 4C 33 0D BA
0D11: A0 01 4C 33 0D 38 A5 FC 1A
0D19: E9 1D 38 ED 7C 18 0A 8D 72
0D21: 78 18 38 A5 FC ED 78 18 0B
0D29: 85 FC A5 FD E9 00 85 FD BC
0D31: A0 01 AD 9A 16 91 FC 4C 0C
0D39: B8 0C 38 AD 01 D0 E9 31 D0
0D41: 4A 4A 4A 8D 71 18 0A 0A 4B
0D49: 6D 71 18 A2 00 8E 71 18 FA
0D51: 0A 0A 2E 71 18 0A 2E 71 C6
0D59: 18 85 FC 8D 7C 18 AD 71 35
0D61: 18 69 D8 85 FD AD 10 D0 CD
0D69: 8D 73 18 38 AD 00 D0 E9 23
0D71: 17 8D 72 18 AD 73 18 E9 C3
0D79: 00 4A 6E 72 18 AD 72 18 F4
0D81: 4A 4A 8D 7E 18 65 FC 85 22
0D89: FC A5 FD 69 00 85 FD 60 77
0D91: A9 00 85 FE A2 28 AD E4 1D
0D99: 17 85 FF A0 00 A9 00 91 12
0DA1: FE C8 C0 40 D0 F7 18 A5 F0
0DA9: FE 69 40 85 FE A5 FF 69 E5
0DB1: 00 85 FF CA D0 E5 60 20 39
0DB9: C3 FF 20 CC FF 20 41 12 DD
0DC1: A9 17 85 FD A9 3C 85 FC 6E
0DC9: 20 4A 12 20 E4 FF F0 FB 38
0DD1: 20 57 12 AD 15 D0 49 03 3B
0DD9: 8D 15 D0 4C 2E 09 20 2F 20
0DE1: 12 AD 15 D0 49 03 8D 15 76
0DE9: D0 20 41 12 A9 17 85 FD 72
0DF1: A9 2A 85 FC 20 4A 12 20 E4
0DF9: DC 0E AD 3E 18 F0 4F AD D6

0E01: E5 17 8D FA 07 A9 01 8D C5
0E09: 3D 18 20 C4 0E A2 01 20 15
0E11: C9 FF A2 28 A9 00 85 FE D3
0E19: AD E4 17 85 FF A0 00 B1 9A
0E21: FE 20 D2 FF C8 C0 40 D0 AD
0E29: F6 18 A5 FE 69 40 85 FE 0B
0E31: A5 FF 69 00 85 FF EE FA AF
0E39: 07 CA D0 E1 A9 01 20 C3 4C
0E41: FF 20 CC FF A5 90 C9 40 6E
0E49: F0 04 C9 00 D0 0E 20 57 5E
0E51: 12 AD 15 D0 49 03 8D 15 E5
0E59: D0 4C 0D 0B 4C B8 0D 20 C0
0E61: 2F 12 AD 15 D0 49 03 8D 10
0E69: 15 D0 20 41 12 A9 17 85 09
0E71: FD A9 33 85 FC 20 4A 12 4B
0E79: 20 DC 0E AD 3E 18 F0 CE 48
0E81: 38 AD E5 17 E9 01 8D FA D7
0E89: 07 A9 00 8D 3D 18 20 C4 02
0E91: 0E A2 01 20 C6 FF A2 28 F4
0E99: A9 00 85 FE AD E4 17 85 F6
0EA1: FF A0 00 20 E4 FF 91 FE D7
0EA9: C8 C0 40 D0 F6 18 A5 FE F7
0EB1: 69 40 85 FE A5 FF 69 00 EE
0EB9: 85 FF EE FA 07 CA D0 E1 AD
0EC1: 4C 3D 0E AD 3E 18 A0 18 16
0EC9: A2 3F 20 BD FF A9 01 A2 D6
0ED1: 08 AC 3D 18 20 BA FF 20 D6
0ED9: C0 FF 60 A9 00 8D 3E 18 88
0EE1: 20 E4 FF F0 FB C9 0D F0 9B
0EE9: 1C C9 14 F0 19 C9 20 90 68
0EF1: EF C9 5B B0 EB AC 3E 18 A6
0EF9: 99 3F 18 20 D2 FF EE 3E 0B
0F01: 18 C0 0F D0 DB 60 AD 3E E1
0F09: 18 C9 00 F0 D3 A9 9D 20 17
0F11: D2 FF A9 20 20 D2 FF A9 4A
0F19: 9D 20 D2 FF CE 3E 18 4C 1B
0F21: E1 0E AD 15 D0 49 03 8D 7E
0F29: 15 D0 AD 76 18 8D 73 18 64
0F31: 20 2F 12 20 41 12 A9 17 C6
0F39: 85 FD A9 84 85 FC 20 4A D7
0F41: 12 20 4C 10 AD 4F 18 F0 D5
0F49: 34 A9 04 A2 04 A0 00 20 92
0F51: BA FF 20 C0 FF A2 04 20 B3
0F59: C9 FF A5 90 C9 80 F0 20 B4
0F61: A0 00 8C 79 18 AC 79 18 5E
0F69: B9 50 18 C9 28 F0 14 20 A2
0F71: 9C 0F EE 79 18 AD 79 18 DC
0F79: CD 4F 18 D0 E8 4C 30 10 F4
0F81: 4C E7 10 AD AC 16 8D 77 3B
0F89: 18 A2 0E A9 0D 20 D2 FF FB
0F91: CA D0 F8 CE 77 18 D0 F1 47
0F99: 4C 73 0F 8D 76 18 20 D8 7D
0FA1: 14 A0 00 AD AC 16 C9 03 93
0FA9: F0 0C C9 02 F0 0E 4C C3 81
0FB1: 0F C0 3C D0 EE 60 20 C9 C7
0FB9: 0F 88 88 88 20 C9 0F 88 E3
0FC1: 88 88 20 C9 0F 4C B2 0F D9
0FC9: A9 10 20 D2 FF AD A8 16 E2
```


www.commodore.ca

```
0FD1: 20 D2 FF AD A9 16 20 D2 25
0FD9: FF 20 F3 0F 20 F3 0F 20 40
0FE1: F3 0F A9 08 20 D2 FF A9 33
0FE9: 0D 20 D2 FF A9 0F 20 D2 95
0FF1: FF 60 A2 08 AD AB 16 8D F9
0FF9: 72 18 B1 FE 3D 9C 16 F0 16
1001: 1B AD AD 16 20 D2 FF AD 2E
1009: A5 16 20 D2 FF CE 72 18 11
1011: D0 F5 CA D0 DF 20 E1 FF 55
1019: F0 15 C8 60 A9 92 20 D2 77
1021: FF AD A7 16 20 D2 FF CE 4E
1029: 72 18 D0 F5 4C 13 10 A9 93
1031: 04 20 C3 FF 20 CC FF 20 26
1039: 57 12 AD 73 18 8D 76 18 F7
1041: AD 15 D0 49 03 8D 15 D0 94
1049: 4C 2E 09 A9 00 8D 4F 18 6B
1051: 8D 8A 02 20 E4 FF C9 0D 47
1059: F0 6A C9 14 F0 6C C9 20 D9
1061: F0 23 C9 21 F0 24 C9 24 63
1069: F0 25 C9 2A F0 26 C9 3F 93
1071: F0 27 C9 41 90 04 C9 5B 4E
1079: 90 24 C9 30 90 D5 C9 3A 92
1081: 90 25 4C 54 10 A2 28 4C FE
1089: B1 10 A2 1A 4C B1 10 A2 B8
1091: 1C 4C B1 10 A2 27 4C B1 83
1099: 10 A2 1B 4C B1 10 48 38 F5
10A1: E9 41 AA 68 4C B1 10 48 36
10A9: 38 E9 13 AA 68 4C B1 10 FF
10B1: AC 4F 18 20 D2 FF 8A 99 DC
10B9: 50 18 EE 4F 18 AD 4F 18 8D
10C1: C9 21 D0 8F A9 80 8D 8A 4F
10C9: 02 60 AD 4F 18 C9 00 F0 FB
10D1: 82 A9 9D 20 D2 FF A9 20 58
10D9: 20 D2 FF A9 9D 20 D2 FF 07
10E1: CE 4F 18 4C 54 10 A9 04 76
10E9: 20 C3 FF 20 CC FF 20 2F 0A
10F1: 12 20 41 12 A9 5C 85 FC FF
10F9: A9 17 85 FD 20 4A 12 20 DA
1101: E4 FF F0 FB AD 73 18 8D 99
1109: 76 18 A9 03 8D 15 D0 20 D7
1111: 57 12 4C 2E 09 20 27 16 5B
1119: A2 18 A0 00 8C 78 18 38 C9
1121: A9 13 ED 78 18 8D 71 18 73
1129: B1 FC 8D 77 18 AC 71 18 2B
1131: B1 FC AC 78 18 91 FC AC 58
1139: 71 18 AD 77 18 91 FC AC 3B
1141: 78 18 C8 C0 0A D0 D5 20 2C
1149: 3E 16 CA D0 CD 4C 2E 09 8A
1151: 20 27 16 18 AD 7F 18 69 75
1159: 98 85 FE A9 DB 85 FF A2 24
1161: 0C A0 00 B1 FE 8D 77 18 DB
1169: B1 FC 91 FE AD 77 18 91 77
1171: FC C8 C0 14 D0 ED 20 3E 29
1179: 16 38 A5 FE E9 28 85 FE 03
1181: A5 FF E9 00 85 FF CA D0 32
1189: D8 4C 2E 09 20 2F 12 20 67
1191: 41 12 AD 15 D0 49 03 8D 52
1199: 15 D0 A9 AE 85 FC A9 16 1A
11A1: 85 FD 20 4A 12 20 06 12 D9
11A9: C9 0D F0 4B 8D A5 16 A9 AF
11B1: C3 85 FC A9 16 85 FD 20 5B
11B9: 4A 12 A9 12 8D AD 16 20 43
11C1: E4 FF C9 59 F0 09 C9 4E DB
11C9: D0 F5 A9 92 8D AD 16 20 3E
11D1: 2F 12 20 41 12 A9 D4 85 8A
11D9: FC A9 16 85 FD 20 4A 12 96
11E1: 20 06 12 C9 0D F0 10 8D 7F
11E9: A7 16 A9 96 8D 9B 16 20 47
11F1: 25 12 CE 9B 16 D0 F8 AD 21
11F9: 15 D0 49 03 8D 15 D0 20 BF
1201: 57 12 4C 2E 09 20 E4 FF F2
1209: F0 FB C9 0D F0 0F C9 20 B6
1211: 90 F3 C9 A0 B0 04 C9 81 FF
1219: B0 EB 20 D2 FF 60 A0 DC 86
1221: 88 D0 FD 60 AE 97 16 20 55
1229: 1F 12 CA D0 FA 60 A0 28 1A
1231: A9 0B 99 BF DB A9 20 99 7E
1239: BF 07 88 C0 00 D0 F1 60 6C
1241: A2 18 A0 00 18 20 F0 FF C5
1249: 60 A0 00 B1 FC 20 D2 FF EB
1251: C8 C9 00 D0 F6 60 20 2F 5B
1259: 12 A9 8D 85 FC A9 17 85 6B
1261: FD 20 41 12 20 4A 12 A9 F8
1269: AA 8D E7 07 A9 0C 8D E7 BB
1271: DB 60 AD 15 D0 49 03 8D 1B
1279: 15 D0 20 2F 12 20 41 12 34
1281: A9 E8 85 FC A9 16 85 FD D9
1289: 20 4A 12 20 E4 FF F0 FB F7
1291: C9 59 D0 11 20 91 0D 20 75
1299: 57 12 AD 15 D0 49 03 8D 70
12A1: 15 D0 4C 0D 0B 20 57 12 75
12A9: AD 15 D0 49 03 8D 15 D0 FC
12B1: 4C 2E 09 A9 18 8D 00 D0 55
12B9: AD 10 D0 29 02 8D 10 D0 E1
12C1: AD 7F 18 F0 05 A9 B8 8D EC
12C9: 00 D0 A9 32 8D 01 D0 4C 22
12D1: 2E 09 20 27 16 A2 18 A0 C1
12D9: 13 A9 05 91 FC 88 C0 FF 73
12E1: D0 F7 20 3E 16 CA D0 EF AA
12E9: 4C 2E 09 20 C8 14 AD 82 9A
12F1: 18 D0 0D A9 18 8D 00 D0 08
12F9: AD 10 D0 29 02 8D 10 D0 22
1301: A9 00 8D 82 18 20 7C 14 83
1309: CE FA 07 AD 7F 18 D0 04 F3
1311: 20 BC 15 60 20 5A 16 A9 9D
1319: 00 8D 7F 18 60 20 B8 14 8B
1321: AD 82 18 D0 0D A9 B8 8D 37
1329: 00 D0 AD 10 D0 29 02 8D 41
1331: 10 D0 A9 00 8D 82 18 20 04
1339: 40 14 EE FA 07 AD 7F 18 C3
1341: F0 04 20 9D 15 60 20 5A E3
1349: 16 A9 14 8D 7F 18 60 A9 4C
1351: 83 85 FE A9 18 85 FF A9 4A
1359: 05 8D 81 18 20 59 15 4C 60
1361: 2E 09 A9 83 85 FE A9 18 0C
1369: 85 FF 20 0A 15 4C 2E 09 B1
```


www.commodore.ca

```
1371: 20 27 16 A2 18 A0 13 B1 EE
1379: FC 8D 77 18 88 B1 FC C8 93
1381: 91 FC 88 88 C0 FF D0 F5 A8
1389: C8 AD 77 18 91 FC 20 3E 7C
1391: 16 CA D0 E1 4C 2E 09 20 C8
1399: 27 16 A2 18 A0 00 B1 FC E0
13A1: 8D 77 18 C8 B1 FC 88 91 50
13A9: FC C8 C8 C0 14 D0 F5 88 5C
13B1: AD 77 18 91 FC 20 3E 16 F1
13B9: CA D0 E1 4C 2E 09 A0 13 6E
13C1: A2 17 18 20 31 16 B1 FC A9
13C9: 8D 77 18 20 4C 16 B1 FC 18
13D1: 8D 72 18 20 3E 16 AD 72 7E
13D9: 18 91 FC 20 32 14 CA D0 82
13E1: ED 20 3E 16 AD 77 18 91 13
13E9: FC 88 C0 FF D0 D2 4C 2E 4E
13F1: 09 A0 13 A2 17 20 27 16 C5
13F9: B1 FC 8D 77 18 20 3E 16 3A
1401: B1 FC 8D 72 18 20 4C 16 4A
1409: AD 72 18 91 FC 20 24 14 28
1411: CA D0 ED 20 4C 16 AD 77 42
1419: 18 91 FC 88 C0 FF D0 D3 AD
1421: 4C 2E 09 18 A5 FC 69 50 19
1429: 85 FC A5 FD 69 00 85 FD 3C
1431: 60 38 A5 FC E9 50 85 FC 29
1439: A5 FD E9 00 85 FD 60 A9 54
1441: 08 8D 77 18 EE 02 D0 AD D5
1449: 02 D0 F0 0F CE 77 18 D0 4B
1451: F3 AD 02 D0 C9 55 F0 0E E3
1459: 4C 7B 14 AD 10 D0 49 02 0F
1461: 8D 10 D0 4C 4D 14 AD 10 3B
1469: D0 29 02 F0 0D AD 10 D0 F1
1471: 49 02 8D 10 D0 A9 15 8D 77
1479: 02 D0 60 A9 08 8D 77 18 7B
1481: CE 02 D0 AD 02 D0 F0 0F A3
1489: CE 77 18 D0 F3 AD 02 D0 2D
1491: C9 0D F0 0E 4C B7 14 AD 2D
1499: 10 D0 49 02 8D 10 D0 4C 80
14A1: 89 14 AD 10 D0 29 02 D0 C9
14A9: 0D AD 10 D0 49 02 8D 10 2E
14B1: D0 A9 4D 8D 02 D0 60 EE 29
14B9: 76 18 AD 76 18 C9 28 D0 47
14C1: 05 A9 00 8D 76 18 60 CE BB
14C9: 76 18 AD 76 18 C9 FF D0 2F
14D1: 05 A9 27 8D 76 18 60 A9 CD
14D9: 00 8D 72 18 AD 76 18 0A 38
14E1: 0A 0A 2E 72 18 0A 2E 72 59
14E9: 18 0A 2E 72 18 0A 2E 72 6F
14F1: 18 85 FE 18 AD E4 17 6D BD
14F9: 72 18 85 FF AD 76 18 6D B3
1501: E5 17 8D FA 07 60 20 D8 E6
1509: 14 A0 00 8C 78 18 8C 77 DE
1511: 18 20 31 16 20 29 15 20 0F
1519: 29 15 20 29 15 EE 78 18 35
1521: AD 78 18 C9 14 D0 EA 60 59
1529: A2 08 A9 00 8D 72 18 AC 42
1531: 78 18 B1 FC 29 0F C9 01 73
1539: D0 09 AD 72 18 1D 9C 16 1B
1541: 8D 72 18 20 4C 16 CA D0 77
1549: E9 AD 72 18 AC 77 18 91 39
1551: FE EE 77 18 60 20 D8 14 3C
1559: A0 00 8C 78 18 8C 77 18 33
1561: 20 31 16 20 78 15 20 78 0F
1569: 15 20 78 15 EE 78 18 AD 59
1571: 78 18 C9 14 D0 EA 60 A2 9E
1579: 08 AD 81 18 AC 78 18 91 97
1581: FC AC 77 18 B1 FE 3D 9C 45
1589: 16 F0 07 AC 78 18 A9 01 7F
1591: 91 FC 20 4C 16 CA D0 E1 20
1599: EE 77 18 60 A9 00 8D 7F 2F
15A1: 18 8D 81 18 20 C8 14 20 FD
15A9: 56 15 A9 14 8D 7F 18 A9 A1
15B1: 05 8D 81 18 20 B8 14 20 EA
15B9: 56 15 60 A9 14 8D 7F 18 68
15C1: A9 00 8D 81 18 20 B8 14 7F
15C9: 20 56 15 A9 05 8D 81 18 2B
15D1: A9 00 8D 7F 18 20 C8 14 9D
15D9: 20 56 15 60 20 2F 12 AD D4
15E1: 15 D0 49 03 8D 15 D0 20 A7
15E9: 41 12 A0 00 B9 11 17 C9 89
15F1: 00 F0 07 20 D2 FF C8 4C F1
15F9: ED 15 20 E4 FF C9 00 F0 BC
1601: F9 C9 30 90 F5 C9 3A B0 30
1609: F1 38 E9 2F 8D 97 16 38 BF
1611: A9 0B ED 97 16 8D 97 16 9C
1619: AD 15 D0 49 03 8D 15 D0 6C
1621: 20 57 12 4C 2E 09 AD 7F 5B
1629: 18 85 FC A9 D8 85 FD 60 2A
1631: 18 AD 7F 18 69 98 85 FC 13
1639: A9 DB 85 FD 60 18 A5 FC 5D
1641: 69 28 85 FC A5 FD 69 00 62
1649: 85 FD 60 38 A5 FC E9 28 1A
1651: 85 FC A5 FD E9 00 85 FD E4
1659: 60 A9 D7 85 FD A9 FF 85 ED
1661: FC A2 18 A0 28 B1 FC 29 B9
1669: 0F C9 01 F0 04 49 05 91 18
1671: FC 88 D0 F1 20 3E 16 CA F8
1679: D0 E9 60 20 27 16 A2 18 AC
1681: A0 13 B1 FC 49 04 91 FC BF
1689: 88 C0 FF D0 F5 20 3E 16 0E
1691: CA D0 ED 4C 2E 09 0C 00 AA
1699: 00 00 FF 00 01 02 04 08 A8
16A1: 10 20 40 80 23 00 20 32 08
16A9: 38 00 01 01 92 98 20 46 75
16B1: 4F 52 45 47 52 4F 55 4E 25
16B9: 44 20 43 48 41 52 2E 3E A9
16C1: 9F 00 98 20 52 45 56 45 4D
16C9: 52 53 45 3F 20 28 59 2F C4
16D1: 4E 29 00 20 42 41 43 4B 7B
16D9: 47 52 4F 55 4E 44 20 43 0E
16E1: 48 41 52 2E 3E 9F 00 9E 68
16E9: 20 20 20 20 12 20 43 4C 2C
16F1: 45 41 52 20 46 4F 4E 54 23
16F9: C0 C0 41 52 45 20 59 4F 1D
1701: 55 20 53 55 52 45 3F 20 16
1709: 28 59 2F 4E 29 20 92 00 E3
```


www.commodore.ca

```
1711: 98 20 43 55 52 53 4F 52 A9
1719: 20 56 45 4C 4F 43 49 54 51
1721: 59 20 28 30 2D 39 29 20 A2
1729: 00 98 20 53 41 56 45 3E 50
1731: 9F 00 98 20 4C 4F 41 44 AA
1739: 3E 9F 00 9E 20 20 20 20 36
1741: 12 20 3C 3C 3C 3C 3C 20 C0
1749: 44 49 53 4B 20 45 52 52 7F
1751: 4F 52 20 3E 3E 3E 3E 3E 4A
1759: 20 92 00 9E 20 20 20 20 2B
1761: 12 20 3C 3C 3C 3C 3C 20 E0
1769: 44 45 56 49 43 45 20 4E 89
1771: 4F 54 20 50 52 45 53 45 B5
1779: 4E 54 20 3E 3E 3E 3E 3E 73
1781: 20 92 00 98 50 52 49 4E 07
1789: 54 3E 9F 00 98 12 41 42 E9
1791: 43 44 45 46 47 48 49 4A C7
1799: 4B 4C 4D 4E 4F 50 51 52 10
17A1: 53 54 55 56 57 58 59 5A 58
17A9: 21 3F 24 30 31 32 33 34 29
17B1: 35 36 37 38 39 92 91 00 E9
17B9: 98 12 20 20 20 20 20 20 25
17C1: 53 54 52 45 41 4D 45 52 27
17C9: 20 46 4F 4E 54 20 20 42 A4
17D1: 59 20 42 4F 42 20 53 50 E2
17D9: 49 52 4B 4F 20 20 20 20 90
17E1: 20 92 00 20 80 7F 80 00 35
17E9: FF C0 00 C0 C0 00 C0 C0 AD
17F1: 00 C0 C0 00 C0 C0 00 C0 B5
17F9: C0 00 C0 C0 00 FF C0 00 FC
1801: 7F 80 1F E0 00 30 30 00 61
1809: 60 18 00 60 18 00 60 18 72
1811: 00 60 18 00 60 18 00 30 32
1819: 30 00 1F E0 FF F8 00 FF 42
1821: F8 00 FF F8 00 FF F8 00 0C
1829: FF F8 00 FF F8 00 FF F8 14
1831: 00 FF F8 00 FF F8 00 FF 23
1839: F8 00 FF F8 FF FF FF 00 2B
```

Beginning address in hex: 2000
Ending address in hex: 29FF

SEQUELA

```
2000: 80 00 00 E0 00 00 FC 00 5E
2008: 00 FF 80 00 FF F0 00 9F 19
2010: FE 00 03 FF C0 03 7F F9 4F
2018: 03 0F FF 03 01 FF 03 00 31
2020: 7F 03 07 F9 03 3F C0 03 A9
2028: FE 00 8F F0 00 FF 80 00 28
2030: FC 00 00 E0 00 00 80 00 8E
2038: 00 00 00 00 00 00 00 00 38
2040: 00 00 00 80 00 01 FF FF C1
2048: FF FF FF FF FF FF FF FF 48
2050: FF FF C0 18 03 C0 18 03 08
2058: C0 18 03 C0 18 03 C0 18 E8
2060: 03 C0 18 03 C0 18 03 E0 FB
2068: 3C 07 F0 7E 0F 7F FF FE A8
2070: 7F F7 FE 3F E3 FC 0F 80 96
2078: F0 00 00 00 00 00 00 00 69
2080: 00 00 00 0F FF F0 3F FF BF
2088: FC 7F FF FE 7F FF FE F0 73
2090: 00 0F E0 00 07 C0 00 03 4B
2098: C0 00 03 C0 00 03 C0 00 E0
20A0: 03 C0 00 03 C0 00 03 E0 0C
20A8: 00 07 F1 00 8F 7F 00 FE AF
20B0: 7F 00 FE 3F 00 FC 0F 00 7A
20B8: F8 00 00 00 00 00 00 00 B1
20C0: 00 00 00 80 00 01 FF FF 42
20C8: FF FF FF FF FF FF FF FF C8
20D0: FF FF E0 00 03 E0 00 03 98
20D8: E0 00 03 E0 00 03 E0 00 81
20E0: 03 E0 00 03 E0 00 03 F0 9C
20E8: 00 07 F8 00 0F 7F FF FE 76
20F0: 7F FF FE 3F FF FC 0F FF BA
20F8: F0 00 00 00 00 00 00 00 E9
2100: 00 00 00 80 00 01 FF FF 81
2108: FF FF FF FF FF FF FF FF 08
2110: FF FF C0 18 03 C0 18 03 C7
2118: C0 18 03 C0 18 03 C0 18 A8
2120: 03 C0 18 03 C0 18 03 C0 9B
2128: 18 03 C0 3C 03 C0 00 03 07
2130: C0 00 03 E0 00 07 70 00 4C
2138: 0E 00 00 00 00 00 00 00 46
2140: 00 00 00 80 00 01 FF FF C1
2148: FF FF FF FF FF FF FF FF 48
2150: FF FF 80 18 03 00 18 03 07
2158: 00 18 03 00 18 03 00 18 A6
2160: 03 00 18 03 00 18 03 00 99
2168: 18 03 00 3C 03 00 00 03 C5
2170: 00 00 03 00 00 07 00 00 7A
2178: 0E 00 00 00 00 00 00 00 86
2180: 00 00 00 0F FF F0 3F FF BF
2188: FC 7F FF FE 7F FF FE F0 73
2190: 00 0F E0 00 07 C0 00 03 4B
2198: C0 00 03 C0 00 03 C0 00 E0
21A0: 03 C0 60 03 C0 60 03 E0 CC
21A8: 60 07 F0 60 8F 7F E0 FE 50
21B0: 7F E0 FE 3F E0 FC 0F E0 1D
21B8: F0 00 00 00 00 00 00 00 A9
21C0: 00 00 00 80 00 01 FF FF 42
21C8: FF FF FF FF FF FF FF FF C8
21D0: FF FF 80 18 01 00 18 00 82
21D8: 00 18 00 00 18 00 00 18 21
21E0: 00 00 18 00 00 18 00 80 91
21E8: 18 01 FF FF FF FF FF FF 02
21F0: FF FF FF FF FF FF 80 00 71
21F8: 01 00 00 00 00 00 00 00 F9
2200: 00 00 00 00 00 00 00 00 00
2208: 00 00 00 00 00 00 00 00 08
2210: 00 00 00 00 00 80 00 01 91
2218: FF FF FF FF FF FF FF FF 18
2220: FF FF FF FF 80 00 01 00 A1
2228: 00 00 00 00 00 00 00 00 28
2230: 00 00 00 00 00 00 00 00 30
2238: 00 00 00 00 00 00 00 00 38
2240: 00 00 00 0E 00 00 3E 00 8C
2248: 00 7E 00 00 7E 00 00 F2 38
```


www.commodore.ca

```
2250: 00 00 E0 00 00 C0 00 00 F1
2258: C0 00 00 C0 00 00 C0 00 9A
2260: 00 E0 00 00 F0 00 01 7F B2
2268: FF FF 7F FF FF 3F FF FF 27
2270: 0F FF FF 00 00 01 00 00 80
2278: 00 00 00 00 00 00 00 00 78
2280: 00 00 00 80 00 01 FF FF 02
2288: FF FF FF FF FF FF FF FF 88
2290: FF FF 80 FE 01 01 F7 00 0A
2298: 03 E3 80 07 C1 C0 0F 80 19
22A0: E0 1F 00 70 3E 00 38 7C 04
22A8: 00 1C F8 00 0E F0 00 07 C3
22B0: E0 00 03 C0 00 01 80 00 D6
22B8: 00 00 00 00 00 00 00 00 B8
22C0: 00 00 00 80 00 01 FF FF 42
22C8: FF FF FF FF FF FF FF FF C8
22D0: FF FF C0 00 01 C0 00 00 53
22D8: C0 00 00 C0 00 00 C0 00 1B
22E0: 00 C0 00 00 C0 00 00 C0 23
22E8: 00 00 C0 00 00 C0 00 00 6A
22F0: C0 00 00 E0 00 00 F0 00 83
22F8: 00 00 00 00 00 00 00 00 F8
2300: 80 00 01 FF FF FF FF FF 81
2308: FF FF FF FF FF FF FF 80 88
2310: 00 7F 00 01 FC 00 07 F0 85
2318: 00 1F C0 00 7F 00 00 1F 96
2320: C0 00 07 F0 00 01 FC 80 57
2328: 00 7F FF FF FF FF FF FF A7
2330: FF FF FF FF FF FF 80 00 B0
2338: 01 00 00 00 00 00 00 00 39
2340: 00 00 00 80 00 01 FF FF C1
2348: FF FF FF FF FF FF FF FF 48
2350: FF FE 80 00 F8 00 03 E0 AC
2358: 00 0F 80 00 3E 00 00 F8 1F
2360: 00 03 E0 00 0F 80 00 3E 12
2368: 00 01 FF FF FF FF FF FF 69
2370: FF FF FF FF FF FF 00 00 70
2378: 01 00 00 00 00 00 00 00 79
2380: 00 00 00 0F FF F0 3F FF BF
2388: FC 7F FF FE 7F FF FE F0 73
2390: 00 0F E0 00 07 C0 00 03 4B
2398: C0 00 03 C0 00 03 C0 00 E0
23A0: 03 C0 00 03 C0 00 03 E0 0C
23A8: 00 07 F0 00 0F 7F FF FE 2E
23B0: 7F FF FE 3F FF FC 0F FF 7A
23B8: F0 00 00 00 00 00 00 00 A9
23C0: 00 00 00 80 00 01 FF FF 42
23C8: FF FF FF FF FF FF FF FF C8
23D0: FF FF 80 18 03 00 18 03 87
23D8: 00 18 03 00 18 03 00 18 27
23E0: 03 00 18 03 00 18 03 00 1A
23E8: 1C 07 00 1E 0F 00 0F FE 47
23F0: 00 0F FE 00 07 FC 00 03 06
23F8: F0 00 00 00 00 00 00 00 E9
2400: 00 00 00 0F FF F0 3F FF 3F
2408: FC 7F FF FE 7F FF FE F0 F2
2410: 00 0F E0 00 07 C0 00 03 CA
2418: C0 00 03 C4 00 03 CC 00 70
2420: 03 DC 00 03 F8 00 03 F0 EF
2428: 00 07 70 00 0F FF FF FE AD
2430: FF FF FE BF FF FC 0F FF FA
2438: F0 00 00 00 00 00 00 00 29
2440: 00 00 00 80 00 01 FF FF C1
2448: FF FF FF FF FF FF FF FF 48
2450: FF FF 80 18 03 00 18 03 07
2458: 00 18 03 00 38 03 00 F8 A7
2460: 03 03 F8 03 0F D8 03 3F 8C
2468: 1C 07 FC 1E 0F F0 0F FE B4
2470: E0 0F FE C0 07 FC 80 03 A7
2478: F0 00 00 00 00 00 00 00 69
2480: 00 00 00 7F 01 F0 3C 07 35
2488: FC 78 0F FE 70 0F FE E0 6B
2490: 1F 0F E0 1E 07 C0 1C 03 A4
2498: C0 1C 03 C0 1C 03 C0 38 51
24A0: 03 C0 38 03 E0 38 03 E0 9C
24A8: 78 07 F0 F8 07 7F F0 0E 97
24B0: 7F F0 1E 3F E0 3C 0F 80 2B
24B8: FE 00 00 00 00 00 00 00 B7
24C0: 00 00 00 00 00 0F 00 00 CF
24C8: 07 00 00 03 00 00 03 00 D5
24D0: 00 03 00 00 03 80 00 03 5A
24D8: FF FF FF FF FF FF FF FF D8
24E0: FF FF FF FF 80 00 03 00 64
24E8: 00 03 00 00 03 00 00 03 F1
24F0: 00 00 03 00 00 07 00 00 FA
24F8: 0F 00 00 00 00 00 00 00 08
2500: 00 00 00 00 00 01 0F FF 10
2508: FF 3F FF FF 7F FF FF 7F 46
2510: FF FF F0 00 01 E0 00 00 E2
2518: C0 00 00 C0 00 00 C0 00 5A
2520: 00 C0 00 00 C0 00 00 60 02
2528: 00 00 78 00 01 3F FF FF E0
2530: 1F FF FF 07 FF FF 00 00 56
2538: 01 00 00 00 00 00 00 00 39
2540: 00 00 01 00 00 07 00 00 48
2548: 3F 00 01 FF 00 0F FF 00 97
2550: 7F F9 03 FF C0 9F FE 00 2C
2558: FF F0 00 FF 80 00 FE 00 C8
2560: 00 9F E0 00 03 FC 00 00 E0
2568: 7F 80 00 0F F1 00 01 FF 6A
2570: 00 00 3F 00 00 07 00 00 B6
2578: 01 00 00 00 00 00 00 00 79
2580: 00 00 01 00 00 7F 00 3F 40
2588: FF 1F FF FF FF FF FF FF A7
2590: FF 81 FF C0 00 3F 00 00 12
2598: 0F F0 00 03 FC 00 0F F0 98
25A0: 00 3F 00 00 FF C0 00 FF A0
25A8: FF 81 FF FF FF 1F FF FF 49
25B0: 00 3F FF 00 00 7F 00 00 6F
25B8: 01 00 00 00 00 00 00 00 B9
25C0: 80 00 01 C0 00 07 F0 00 FA
25C8: 1F FC 00 7F FF 01 FF 1F 84
25D0: C7 F9 07 FF E0 01 FF 80 FB
25D8: 00 7E 00 00 3C 00 00 7E 12
25E0: 00 01 FF 80 07 FF E0 9F E9
25E8: E3 F9 FF 80 FF FE 00 3F 85
```


www.commodore.ca

```
25F0: F8 00 0F E0 00 03 80 00 5D
25F8: 01 00 00 00 00 00 00 00 F9
2600: 00 00 00 00 00 01 00 00 01
2608: 07 00 00 1F 00 00 7F 00 AD
2610: 01 FF 00 03 F9 80 0F E0 7E
2618: FF FF 80 FF FE 00 FF FC 94
2620: 00 FF FE 00 80 0F 80 00 2F
2628: 03 E0 00 00 F9 00 00 3F 45
2630: 00 00 0F 00 00 03 00 00 42
2638: 01 00 00 00 00 00 00 00 39
2640: 00 00 00 FC 00 0F FE 00 4B
2648: 07 FF 00 03 FF 80 03 C7 9D
2650: C0 03 C3 E0 03 C1 F0 03 71
2658: C0 F8 03 C0 7C 03 C0 3E 54
2660: 03 C0 1F 03 C0 0F 83 C0 5A
2668: 07 C3 C0 03 E3 C0 01 FF 9C
2670: C0 00 FF E0 00 7F F0 00 82
2678: 3F 00 00 00 00 00 00 00 B7
2680: 00 00 00 00 00 00 00 00 80
2688: 00 00 00 00 00 00 00 00 88
2690: 00 00 00 00 00 00 0F F8 98
2698: 40 7F FE E7 FF FF E7 FF 27
26A0: FF 40 7F FE 00 0F F8 00 67
26A8: 00 00 00 00 00 00 00 00 A8
26B0: 00 00 00 00 00 00 00 00 B0
26B8: 00 00 00 00 00 00 00 00 B8
26C0: 00 00 00 00 00 00 00 00 C0
26C8: 78 00 00 FC 00 00 8E 00 CC
26D0: 01 C6 00 01 C7 01 80 C3 A6
26D8: 43 C0 03 E3 E0 03 E6 60 EE
26E0: 03 40 30 03 00 30 07 00 8E
26E8: 18 07 00 1C 0E 00 0F FE 40
26F0: 00 07 FC 00 01 F0 00 00 E6
26F8: 00 00 00 00 00 00 00 00 F8
2700: 00 00 00 07 01 C0 0F 87 5F
2708: F0 1C 8F F8 18 0E 38 38 34
2710: 1C 1C 30 1C 1C FF FF FF B0
2718: FF FF FF 30 18 0C 30 18 B4
2720: 0C FF FF FF FF FF FF 38 64
2728: 38 0C 38 38 1C 1C 70 18 9D
2730: 1F F1 38 0F E1 F0 03 80 DE
2738: E0 00 00 00 00 00 00 00 19
2740: 00 00 00 0F FF F0 3F FF 7F
2748: FC 7F FF FE 7F FF FE FB 3E
2750: C0 1F F1 E0 0F E0 F0 07 EA
2758: E0 78 07 E0 3C 07 E0 1E DB
2760: 07 E0 0F 07 E0 07 87 F0 BE
2768: 03 CF F8 01 FF 7F FF FE B3
2770: 7F FF FE 3F FF FC 0F FF 3A
2778: F0 00 00 00 00 00 00 00 69
2780: 00 00 00 00 00 00 00 00 80
2788: 00 00 00 00 00 00 00 00 88
2790: 00 00 C0 00 18 C0 00 18 42
2798: FF FF FC FF FF FE FF FF 94
27A0: FF FF FF FF C0 00 00 C0 22
27A8: 00 00 00 00 00 00 00 00 A8
27B0: 00 00 00 00 00 00 00 00 B0
27B8: 00 00 00 00 00 00 00 00 B8
27C0: 00 00 00 F8 00 78 FC 00 2F
27C8: FC FE 00 FE FF 00 FE FF C2
27D0: 80 9E E7 C0 0F E3 C0 07 53
27D8: E1 E0 07 E0 F0 07 E0 F0 4D
27E0: 07 E0 78 07 E0 38 07 E0 49
27E8: 3C 0F E0 1E 1F E0 1F FE 51
27F0: E0 0F FE F0 07 FC F8 01 CE
27F8: F8 00 00 00 00 00 00 00 F1
2800: 00 00 00 1E 00 1F 3F 00 7C
2808: 0F 7F 00 07 7F 00 07 F9 1E
2810: 00 07 F0 00 07 E0 00 07 F6
2818: E0 08 07 E0 0C 07 E0 0E EA
2820: 07 E0 1F 07 E0 1F 87 F0 A6
2828: 3F C7 F8 79 E7 7F F8 F7 F9
2830: 7F F0 7F 3F E0 3F 1F 80 1F
2838: 1F 00 00 00 00 00 00 00 57
2840: 00 00 00 04 00 00 07 00 4B
2848: 00 07 C0 00 07 E0 00 06 FD
2850: F8 00 06 3E 00 06 0F 80 23
2858: 06 03 E0 06 00 F8 06 00 47
2860: 3C 86 00 0F FF FF FF FF 32
2868: FF FF FF FF FF FF FF FF 68
2870: 86 00 00 06 00 00 06 00 03
2878: 00 00 00 00 00 00 00 00 78
2880: 00 00 00 1E 07 FF 3F 07 EB
2888: FF 7F 07 FF 7F 07 FF F9 8F
2890: 07 07 F0 07 07 E0 07 07 8C
2898: E0 07 07 E0 07 07 E0 07 5E
28A0: 07 E0 07 07 E0 07 07 F0 76
28A8: 0F 07 F8 1F 07 7F FE 07 63
28B0: 7F FE 07 3F FC 0F 1F F0 91
28B8: 1F 00 00 00 00 00 00 00 D7
28C0: 00 00 00 0F FF F0 3F FF FF
28C8: FC 7F FF FE 7F FF FE F8 BB
28D0: 78 1F F0 38 0F E0 1C 07 A4
28D8: E0 1C 07 E0 1C 07 E0 1C DD
28E0: 07 E0 1C 07 E0 1C 07 F0 E0
28E8: 3C 0F F8 7C 3F 7F F8 3E 9F
28F0: 7F F8 3E 3F F0 3C 0F C0 E3
28F8: 30 00 00 00 00 00 00 00 29
2900: 00 00 00 00 00 1F 00 00 1F
2908: 0F 00 00 07 00 00 07 70 95
2910: 00 07 FC 00 07 FF 00 07 22
2918: FF C0 07 FF E0 07 8F F0 48
2920: 07 01 FC 07 00 7E 07 00 B1
2928: 3F 87 00 0F E7 00 03 FF E8
2930: 00 00 FF 00 00 3F 00 00 6F
2938: 1F 00 00 00 00 00 00 00 57
2940: 00 00 00 0F 80 F0 3F E3 E3
2948: FC 7F F7 FE 7F FF FE F8 33
2950: 7F 1F F0 3E 0F E0 1C 07 31
2958: E0 1C 07 E0 1C 07 E0 1C 5D
2960: 07 E0 1C 07 E0 1C 07 F0 60
2968: 3E 0F F8 7F 1F 7F FF FE CB
2970: 7F F7 FE 3F E3 FC 0F 80 96
2978: F0 00 00 00 00 00 00 00 69
2980: 00 00 00 0C 03 F0 3C 0F CB
2988: FC 7C 1F FE 7C 1F FE FC B7
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 113 and 116 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
2990: 3E 1F F0 3C 0F E0 38 07 4A
2998: E0 38 07 E0 38 07 E0 38 F1
29A0: 07 E0 38 07 E0 38 07 F0 D8
29A8: 1C 0F F8 1E 1F 7F FF FE 88
29B0: 7F FF FE 3F FF FC 0F FF 7A
29B8: F0 00 00 00 00 00 00 00 A9
29C0: 00 00 00 00 00 00 00 18 D8
29C8: 00 00 18 00 18 18 18 1E 47
29D0: 18 78 07 99 E0 01 FF 80 64
29D8: 00 FE 00 00 3C 00 00 3C 50
29E0: 00 00 FE 00 01 FF 80 07 68
29E8: 99 E0 1E 18 78 18 18 18 5A
29F0: 00 18 00 00 18 00 00 00 21
29F8: 00 00 00 00 00 00 00 00 F8
```

INFRARAID
FROM PAGE 110

```
·0 REM ********************************  CE
·1 REM *                              *  OO
·2 REM *      -- INFRARAID --         *  CO
·3 REM * ERROR TRAPPING FOR THE C-64 *  MA
·4 REM *                              *  OO
·5 REM * AUTHOR: TIMOTHY VANDEVENTER *  DN
·6 REM *         3851 E. EATON HWY.   *  JG
·7 REM *         SUNFIELD, MI  48890  *  OL
·8 REM *                              *  OO
·9 REM ********************************  CE
·10 POKE53281,0:POKE53280,0:PRINT"[CLEAR]
 [GREEN]"                                 DN
·15 FORI=50176TO50670                     BF
·20 READA:CS=CS+A:LN=100+INT((I-50176)/7)
 *10                                      NC
·30 IFA>255ORA<0THENPRINT:PRINT"[RED]BAD
 DATA IN LINE"LN:STOP                     NM
·40 POKEI,A:PRINTLN;:X=X+1:IFX=7THENX=0:P
 RINT                                     PC
·50 NEXT                                  IA
·60 IFCS<>65731THENPRINT:PRINT"[RED]CHECK
 SUM ERROR. CHECK ALL DATA.":STOP         DP
·70 PRINT:PRINT"[CLEAR][4"[DOWN]"][GREEN]
 INFRARAID LOADED!"                       BL
·80 PRINT"[DOWN][DOWN]DO YOU WISH TO SAVE
  TO [RED]D[GREEN]ISK OR [RED]T[GREEN]APE
 ?"                                       GI
·82 GETC$:IFC$<>"T"ANDC$<>"D"THEN82       GI
·84 IFC$="T"THEN92                        FM
·86 POKE43,0:POKE44,196:POKE45,239:POKE46
 ,197                                     LH
·88 SAVE"INFRARAID(50176)",8,1            MM
·90 GOTO98                                AH
·92 POKE43,0:POKE44,196:POKE45,239:POKE46
 ,197                                     LH
·94 SAVE"INFRARAID(50176)",1,1            MB
·98 POKE43,1:POKE44,8:POKE45,1:POKE46,8   HB
·99 PRINT"[DOWN]TO ACTIVATE, ENTER [RED]S
 YS 50176":END                            FB
·100 DATA 173,0,3,205,235,197,240         CO
·110 DATA 3,141,225,197,173,1,3           NB
·120 DATA 205,236,197,240,15,141,226      JB
·130 DATA 197,173,235,197,141,0,3         CO
·140 DATA 173,236,197,141,1,3,32          II
·150 DATA 99,197,76,125,197,142,237       LJ
·160 DATA 197,32,189,197,32,99,197        MI
·170 DATA 224,128,240,6,165,58,201        NM
·180 DATA 255,208,3,76,17,197,32          CO
·190 DATA 125,197,240,74,224,15,208       PH
·200 DATA 12,41,1,240,66,173,238          MH
·210 DATA 197,208,45,104,104,96,224       LC
·220 DATA 20,208,10,41,2,240,50           JO
·230 DATA 173,238,197,208,29,96,224       KM
·240 DATA 4,208,10,41,4,240,36            PM
·250 DATA 173,238,197,208,15,96,224       MJ
·260 DATA 5,208,26,41,8,240,22            HL
·270 DATA 173,238,197,208,1,96,173        FD
·280 DATA 227,197,133,20,173,228,197      IL
·290 DATA 133,21,32,163,168,76,174        CA
·300 DATA 167,169,32,162,255,232,157      DO
·310 DATA 0,4,157,250,4,157,244           FO
·320 DATA 5,157,238,6,224,250,208         MC
·330 DATA 239,162,255,232,181,217,9       LM
·340 DATA 128,149,217,224,25,208,245      ND
·350 DATA 169,14,141,32,208,169,6         NG
·360 DATA 141,33,208,169,14,141,134       JG
·370 DATA 2,24,162,1,134,214,160          MG
·380 DATA 0,132,211,32,240,255,169        IO
·390 DATA 0,133,19,133,184,165,57         KL
·400 DATA 133,20,165,58,133,21,32         GH
·410 DATA 19,166,144,44,160,2,132         LL
·420 DATA 15,177,95,170,200,177,95        KN
·430 DATA 132,73,32,205,189,169,32        GO
·440 DATA 164,73,41,127,32,210,255        LN
·450 DATA 201,34,208,6,165,15,73          BC
·460 DATA 255,133,15,200,240,7,32         HN
·470 DATA 60,197,177,95,208,11,174        CP
·480 DATA 237,197,169,14,141,134,2        BH
·490 DATA 108,225,197,16,220,201,255      DF
·500 DATA 240,216,36,15,48,212,56         FI
·510 DATA 233,127,170,132,73,160,255      JL
·520 DATA 202,240,8,200,185,158,160       MA
·530 DATA 16,250,48,245,200,185,158       KG
·540 DATA 160,48,183,32,210,255,208       JC
·550 DATA 245,132,73,169,14,141,134       GJ
·560 DATA 2,152,24,101,95,197,122         PC
·570 DATA 240,16,200,24,105,1,197         KN
·580 DATA 122,208,13,177,95,240,4         CA
·590 DATA 201,58,208,5,169,1,141          HE
·600 DATA 134,2,164,73,96,173,229         CG
·610 DATA 197,133,69,173,230,197,133      LJ
·620 DATA 70,32,231,176,160,0,152         MH
```


www.commodore.ca

```
·630 DATA 145,71,174,237,197,200,138      KN
·640 DATA 145,71,96,173,231,197,133       HP
·650 DATA 69,173,232,197,133,70,32        KK
·660 DATA 231,176,169,237,141,40,3        GG
·670 DATA 169,246,141,41,3,169,237        KJ
·680 DATA 141,40,3,160,0,177,71           IM
·690 DATA 41,32,240,5,169,239,141         KB
·700 DATA 40,3,177,71,41,64,240           HA
·710 DATA 10,169,188,141,41,3,169         OI
·720 DATA 54,141,40,3,200,177,71          MF
·730 DATA 174,237,197,96,173,233,197      PG
·740 DATA 133,69,173,234,197,133,70       JM
·750 DATA 32,231,176,160,0,177,71         AP
·760 DATA 141,228,197,141,238,197,200     PL
·770 DATA 177,71,141,227,197,13,238       GP
·780 DATA 197,141,238,197,96,139,227      EG
·790 DATA 0,0,197,210,212,210,197         JL
·800 DATA 204,40,196,128,0                ME
```

MICROSIM
FROM PAGE 89

```
·10 REM MICRO FLIGHT SIMULATOR                                  OF
·15 REM BY TIM GERCHMEZ                                         GN
·20 IFPEEK(49152)=169ANDPEEK(49158)=69THEN30                    PC
·25 FORT=49152TO49152+36:READA:POKET,A:NEXT                     NJ
·30 POKE53265,27:POKE56333,127:POKE788,0:POKE789,192:POKE53274,129   BK
·35 POKE53281,0:POKE646,1:POKE650,128:POKE49169,2:POKE49167,255:POKE49160,11   CK
·40 PRINTCHR$(147)CHR$(9)CHR$(15)CHR$(8)"[DOWN][DOWN]";:R1$="[10"[RIGHT]"]":LF$="[7"[LEFT]"]"   OM
·45 L6$="[6"[LEFT]"]":DN$="[HOME][11"[DOWN]"]"                  NF
·46 DIMJ(127):J(109)=45:J(126)=85:J(125)=68:J(123)=44:J(119)=46:J(110)=43   HH
·47 JY=56320:KB=631:KJ=198                                      JB
·50 PRINT"[RED][7"[s *]"][c E][22"[s *]"][c E][9"[s *]"][UP][UP][7"[RIGHT]"]";   FP
·55 PRINT"[s B][WHITE]MICRO FLIGHT SIMULATOR[RED][s B][HOME][7"[RIGHT]"][s U][22"[s C]"][s I][HOME]"   CK
·60 PRINT"[DOWN][DOWN][CYAN][s U][6"[s C]"][s I]":PRINT"[s B][6" "][s B]":PRINT"[s J][6"[s C]"][s K]":PRINT"[RIGHT][RIGHT][YELLOW]TIME[CYAN]"   GK
·65 PRINT"[s U][7"[s C]"][s I]":PRINT"[s B][7" "][s B]":PRINT"[s J][7"[s C]"][s K]":PRINT"[RIGHT][RIGHT][YELLOW]ERPM[CYAN]"   IP
·70 PRINT"[s U][8"[s C]"][s I]":PRINT"[s B][8" "][s B]":PRINT"[s J][8"[s C]"][s K]":PRINT"[RIGHT][YELLOW]AIRSPEED"   NF
·75 PRINT"[3"[RIGHT]"]KTS[CYAN]":PRINT"[s U][11"[s C]"][s I]":PRINT"[s B][11" "][s B]":PRINT"[s J][11"[s C]"][s K]"   PF
·80 PRINT"[RIGHT][YELLOW]GROUNDSPEED":PRINT"[5"[RIGHT]"]MPH"   JK
·85 PRINT"[HOME][3"[DOWN]"][CYAN]"R1$"[s U][9"[s C]"][s I]":PRINTR1$"[s B][9" "][s B]":PRINTR1$"[s J][9"[s C]"][s K]"   KA
·90 PRINTR1$"[RIGHT][c 1]ALTIMETER[CYAN]"   FL
·95 PRINTR1$"[s U][6"[s C]"][s I]":PRINTR1$"[s B][6" "][s B]":PRINTR1$"[s J][6"[s C]"][s K]"   JO
·100 PRINTR1$"[RIGHT][c 3]RUDDER[CYAN][3"[UP]"][RIGHT][RIGHT][s U][5"[s C]"][s I][DOWN]"LF$"[s B][5" "][s B][DOWN]"LF$"[s J][5"[s C]"][s K][DOWN]"L6$"ELEV."   KB
·105 PRINTR1$"[s U][7"[s C]"][s I]":PRINTR1$"[s B][7" "][s B]"   MF
·110 PRINTR1$"[s J][7"[s C]"][s K]":PRINTR1$"[RIGHT][WHITE]HEADING[CYAN]"   JA
·115 R2$="[11"[RIGHT]"]"   OA
·120 R1$="[19"[RIGHT]"]":PRINTR1$"[DOWN][s U][4"[s C]"][s I][WHITE][UP][UP][5"[LEFT]"]BI>[CYAN][DOWN][DOWN]":PRINTR1$"[s B]FUEL[s B]"   HG
·125 PRINTR1$"[s B][4"[s C]"][s B]":PRINTR1$"[s B][GREEN]F[CYAN][3" "][s B]":PRINTR1$"[s B][4" "][s B]":PRINTR1$"[s B][4" "][s B]"   IJ
·130 PRINTR1$"[s B][RED]E[CYAN][3" "][s B]":PRINTR1$"[s J][4"[s C]"][s K]"   AG
·135 PRINT"[UP][UP][4"[RIGHT]"][WHITE][sQ]":PRINT"[RIGHT][RIGHT][c 8]STALL[3"[RIGHT]"][WHITE]OIL TEMP[UP]"LF$"[LEFT][LEFT][CYAN][s J][8"[s C]"][s K]";   HA
·140 PRINT"[UP]"LF$"[3"[LEFT]"][s B][WHITE]C[6" "]H[CYAN][s B][UP]"LF$"[3"[LEFT]"][s U][8"[s C]"][s I]";   DB
·145 PRINT"[UP][UP][5"[LEFT]"][s J][s C][s K][DOWN][LEFT][LEFT]P[UP][UP][LEFT][LEFT][s B][RVSON][WHITE] [RVSOFF][CYAN][s B][UP][3"[LEFT]"][s U][s C][s I]"   BF
·150 PRINTDN$"[DOWN]"R1$"[RIGHT][YELLOW]FLAPS"   MD
·155 PRINT"[CYAN][HOME][3"[DOWN]"]"R1$"[RIGHT][RIGHT][s U][3"[s C]"][s I][DOWN]"L6$"[RIGHT][s B][RVSON][WHITE][3" "][CYAN][RVSOFF][s B][DOWN]"L6$"[RIGHT][s J][3"[s C]"][s K]";   DE
·160 PRINT"[DOWN]"L6$"[RIGHT][RIGHT]CWI"   BA
·165 GOSUB255   CK
·170 POKE198,0   KB
·175 GETA$:IFA$<>""THEN180   CP
·176 POKEKJ,1:POKEKB,J(PEEK(JY)):GETA$:IFA$=""THENGOSUB450:GOTO245   DO
·180 IFA$="O"THENIFGD=1ANDGS=0THEN935   HK
·185 IFA$="X"THENGOSUB605  :GOTO245   GD
·190 IFA$="H"THENGOSUB635  :GOTO245   IO
·195 IFA$="+"ORA$="-"THENGOSUB680  :GOTO2
```


www.commodore.ca

```
45                                                              OG
·200 IFA$="R"THENFP=1:GOTO245                                   MI
·205 IFA$="L"THENFP=0:GOTO245                                   OP
·210 IFA$="<"ORA$=","THENRU=RU+(RU>-45):P
RINTRU$;RU;S$                                                   JO
·215 IFA$=">"ORA$="."THENRU=RU-(RU<45):PR
INTRU$;RU;S$                                                    FI
·220 IFA$="U"THENEL=EL-(EL<45):PRINTEL$;E
L;S$:GOTO245                                                    NN
·225 IFA$="D"THENEL=EL+(EL>-45):PRINTEL$;
EL;S$:GOTO245                                                   LE
·226 IFA$>"0"ANDA$<"8"THENGOSUB1200                             DE
·228 IFA$="0"THENGOSUB251                                       NP
·230 IFRU=0THENPRINTP8$                                         PK
·235 IFRU<0THENPRINTP7$                                         CC
·240 IFRU>0THENPRINTP9$                                         PK
·245 IFDOTHENRETURN                                             FN
·250 GOTO175                                                    CL
·251 FY=FY+6:IFFY>6THENFY=0                                     FB
·252 IFFY=0 THEN PRINTDN$RX$"[WHITE]NIGHT
":POKE49160,0:GOTO254                                           JG
·253 PRINTDN$RX$"[WHITE]DAY  ":POKE49160,
11                                                              AM
·254 PRINTRX$"FLYING":RETURN                                    FC
·255 REM DEFINE VARIABLES + SET UP                              MD
·260 MJ$="[8"[DOWN]"]":TM$="[HOME][4"[DOW
N]"][RIGHT]":ER$="[HOME]"+MJ$+"[RIGHT]":
AS$=DN$+"[DOWN][RIGHT]"                                         DL
·265 GS$=DN$+"[6"[DOWN]"][RIGHT]":AT$="[H
OME][4"[DOWN]"]"+R2$                                            OB
·270 CL$="[HOME][4"[DOWN]"]"+R1$+"[3"[RIG
HT]"]"                                                          JG
·275 EL$="[HOME][8"[DOWN]"]"+R1$+"[RIGHT]
"                                                               HJ
·280 FP$=DN$+"[DOWN][DOWN]"+R1$+"[RIGHT]"
:FP=1                                                           EG
·285 P7$=FP$+"[RED][4"[RIGHT]"][DOWN][3"
"][LEFT][UP][s N][DOWN][LEFT][LEFT][s Q]
[DOWN][LEFT][LEFT][s N]":P8$=FP$+"[4"[RI
GHT]"][3" "][DOWN][DOWN][3"[LEFT]"][3" "
][UP][3"[LEFT]"][GREEN][s C][s Q][s C]"                         NA
·290 P9$=FP$+"[RED][4"[RIGHT]"][DOWN][3"
"][UP][3"[LEFT]"][s M][DOWN][s Q][DOWN][
s M]"                                                           MN
·295 IFQ0>120ORAS>120THENAS=120:Q0=120                          CL
·300 RU$="[HOME]"+MJ$+R2$:HE$=DN$+"[DOWN]
"+R2$:S$="[LEFT]  "                                             KC
·305 PR$=DN$+"[6"[DOWN]"]"+R2$+"[4"[RIGHT
]"]"                                                            GP
·306 A$="[6"[DOWN]"]":B$="[25"[RIGHT]"]":
I1$=DN$+A$+B$:I2$=DN$+"[DOWN][DOWN]"+A$+
B$                                                              MB
·307 I3$=DN$+"[4"[DOWN]"]"+A$+B$:I4$=DN$+
"[6"[DOWN]"]"+A$+B$                                             IG
·308 IN$="[s Z]":PRINTI1$"[RED]"IN$"[WHIT
E]1"I2$"[RED]"IN$"[WHITE]2"I3$IN$"3"I4$"
[GREEN]"IN$"[WHITE]4[RED]"                                      AI
·310 DIMFL$(2):FL$(1)=DN$+MJ$+"[3"[DOWN]"
]"+R1$+"[4"[RIGHT]"]":FL$(2)=FL$(1)                             FO
·315 OT$=DN$+MJ$+"[DOWN][DOWN]"+R2$+"[5"[
RIGHT]"]"                                                       ML
·320 ER=800:AS=90:GS=AS:AT=GS:RU=AT:HE=RU
:EL=HE:F1=235:F2=2:F3=0:F4=6:W=2                                EB
·325 SW=56180:SI=F3:GD=1:Q0=0:DG=Q0:DO=0:
ML=0:I1=1:I2=1:I3=0:I4=1:BO=49169                               DN
·327 I5=0:I6=0:I7=0:KX=0:FY=6                                   CC
·330 S=54272:FORT=STOS+24:POKET,0:NEXT:PO
KES+6,240:POKES,F1:POKES+1,F2                                   EG
·335 POKES+4,33:POKES+24,15:POKES+13,240:
POKES+7,F3:POKES+8,F4:POKES+11,17                               GP
·340 PRINT"[WHITE]"TM$;TI$:PRINTP8$"[WHIT
E]"                                                             MA
·345 FORX=90TO1STEP-W:POKESW,2                                  HK
·350 PRINTER$;ER;S$;AS$;AS;S$;GS$;GS;S$;A
T$;AT;S$;RU$;RU;S$;HE$;HE;S$;FL$(1)"[BAC
KARROW]"                                                        OA
·355 PRINTEL$;EL;S$:EL=EL-W                                     AI
·360 PRINTOT$"[RED][UPARROW][WHITE]";:IFX
/18=INT(X/18)THENPRINTOT$" ";:OT$=LEFT$(
OT$,LEN(OT$)-1)                                                 PO
·365 ER=ER-W:AS=AS-W:GS=GS-W:AT=AT-W:RU=R
U-W:HE=HE+W:F1=F1-.5:POKES,F1:POKESW,1                          OD
·370 F2=F2-.021:POKES+1,F2                                      JL
·372 IFX=300RX=600RX=90THENPRINTFL$(1)" "
:FL$(1)=FL$(1)+"[UP]":FL$(2)=FL$(2)+"[UP
]"                                                              FE
·375 NEXT:TI$="[6"0"]":PRINTTM$;TI$;FP$;"
[YELLOW]UP  "                                                   OH
·380 POKES+15,75:POKES+20,240:POKES+18,17
:FORT=1TO300:NEXT                                               AE
·385 POKES+15,50:FORX=1TO300:NEXT:POKES+1
8,16                                                            HH
·390 RX$=R1$+"[8"[RIGHT]"]"                                     AL
·392 PRINTDN$RX$"[WHITE][LEFT][UP][UP][c
D]":PRINTDN$RX$"[LEFT][UP][c D]":PRINTDN
$RX$"[LEFT][c D]"                                               HA
·395 PRINT"[HOME][RED][DOWN][DOWN]"RX$"[c
R]";:FORT=1TO22:PRINT"[DOWN][LEFT][s B]
";:NEXT                                                         PJ
·400 RX$=RX$+"[RIGHT]":PRINT"[HOME][3"[DO
WN]"][WHITE]"RX$;"+ - ENGINE":PRINTRX$"<
 > RUDDER"                                                      AP
·405 PRINTRX$"U D ELEVATOR":PRINTRX$"[UP]
R L FLAPS":PRINTRX$"H HEAR ATIS"                                BM
·410 PRINTRX$"X REFUEL":PRINTRX$"O DEMO"                        GE
·415 PRINTRX$"[RED][LEFT][c Q][11"[s C]"]
";:POKE1463,67:POKE1463+S,2                                     KN
·420 PRINTRX$"[RED][RIGHT]CABIN":PRINTRX$
"PRESSURE":PRINTRX$"ON[WHITE]":PRINTPR$"
[RVSON][RED] [RVSOFF][WHITE]"                                   BF
·425 POKES+15,255:POKES+18,16:POKES+18,12
9                                                               AI
·430 PRINTTM$;TI$:IFTI$<>"[4"0"]16"THEN43
0  "                                                            GG
·435 PRINTDN$;RX$;"[5" "]"                                      MC
·440 PRINTRX$"[8" "]":PRINTRX$"  ":POKES+
```


www.commodore.ca

```
 18,128: POKES+15,90:POKES+18,17                                    KN
·445 PRINTPR$"[RVSON][WHITE] [RVSOFF]"                              FO
·446 T=INT(8000*RND(1))+1000:U=INT(300*RN
 D(1))+200:Y1=T+U:Y2=T-U:GOSUB875                                   OH
·448 RETURN                                                         IM
·450 PRINT"[WHITE]"EL$;EL;S$;TM$;TI$;ER$;
 ER;S$;AS$;AS;S$;GS$;GS;S$;AT$;AT;S$                                EM
·465 PRINTRU$;RU;S$;HE$;HE;S$;:PC=RND(1)                            IH
·470 IFAT>31000THENPOKES+4,32:GOSUB895:GO
 SUB845:GOTO820                                                     DA
·475 IFAS>100THENIFEL>30THENGOSUB900:GOSU
 B845:GOTO820                                                       CC
·480 IFPC>.997THENIFLEN(OT$)<38THENPRINTO
 T$" ":OT$=OT$+"[RIGHT]":PRINTOT$"[RED][U
 PARROW][WHITE]"                                                    GA
·485 IFF2<1THENGS=0                                                 AO
·487 KX=KX+1:KX=KX-.2*(I5=1)-.5*(I6=1)-.5
 *(I7=1)                                                            EM
·490 IFPC>.98ANDDG=0ANDAT>5000THENPRINTCL
 $"[RED][RVSON][3" "][RVSOFF][WHITE]":DG=
 1:D3=HE:D8=0:GOSUB890                                              PC
·495 IFDG=1THEND8=D8+1:GOSUB910                                     KI
·500 IFGS>10RAS>1THENIFPC>.88THENGS=GS-1:
 Q0=Q0-2:F1=F1-10:ER=ER+20*(ER>0)                                   GJ
·505 IFF1<0THENF1=255:F2=F2-1:IFF2<0THENF
 2=0                                                                KN
·510 POKES,F1:POKES+1,F2                                            CE
·515 IFSITHENPOKESW,2:POKES+18,17:POKES+1
 5,50:AT=AT+D9*(AT>0):D9=D9+24:GOTO525                              AM
·520 POKESW,1:POKES+18,16:D9=15:GOTO530                             KK
·525 POKES+1,D9 AND3                                                HJ
·530 IFGD=0THENGOSUB745:IFGD=0THENSI=-(AS
 <10):IFAS<30THENIFEL>20THENSI=1                                    BF
·535 IFGD=0ANDER<300THENPOKES+4,32:GOSUB8
 95 :GOSUB845:GOTO820                                               FM
·540 IFAT<=0ANDGD=0THEN820                                          OF
·545 IFAS<Q0THENAS=Q0                                               LA
·550 IFEL>40ANDGD=0THENSI=1                                         IE
·555 IFFPTHENPRINTFP$"[YELLOW]UP  ":GOTO5
 65                                                                 AH
·560 PRINTFP$"[YELLOW]DOWN":                                        JO
·565 IFGD=0THENGS=0:IFAT<=10THENAT=10                               KH
·570 IFEL<=40ANDAS>=10THENSI=0                                      GI
·575 IFGD=1THENIFFP=0THENIFEL>20THENIFGS>
 60THENGD=0:AT=30:AS=GS:GS=0:GOSUB602                               BA
·580 IFAT<=0THENAT=0                                                MK
·585 IFKX>300THENKX=0:GOSUB800                                      AF
·590 IFGD=0THENAS=AS-1:IFPC>.7THENAS=AS+1
 :IFPC>.8THENAS=AS+1                                                NL
·595 IFGD=0THENIFAS>120THENAS=120                                   HP
·600 RETURN                                                         IM
·602 Q0=AS:TI$="[6"0"]":POKEBO,FY:CO=0:MX
 =0:RETURN                                                          GO
·605 IFGD<>1THEN625                                                 PH
·610 PRINTFL$(I4)" ";                                               JL
·615 IFLEN(FL$(1))<49THENFL$(1)=FL$(1)+"[
 UP]":GOTO615                                                       DN
·620 FL$(2)=FL$(1):PRINTFL$(I4)"[WHITE][B
 ACKARROW]";:RETURN                                                 AK
·625 PRINTDN$"[13"[DOWN]"]"R2$"[5"[RIGHT]
 "][WHITE]GROUND ONLY":                                             CP
·630 FORX=1TO1000:NEXT:PRINT"[11"[LEFT]"]
 [11" "]";:RETURN                                                   AK
·635 PRINTDN$RX$"[YELLOW]TERMINAL":PRINTR
 X$"INFORMATION":PRINTRX$"SERVICE"                                  JM
·640 T1=65:FORX=1TO10:T1=T1+(RND(1)>.5):N
 EXT:PRINTRX$"[WHITE][DOWN]TEMP:"T1                                 BA
·645 PRINTRX$"ALT:29.95":T1=INT(15*RND(1)
 )+10:PRINTRX$"VIS:"T1                                              ME
·646 PRINTRX$"TNMT:"ML                                              LN
·650 IFRND(1)>.5THENA$="ALPHA[3"."]":GOTO
 665                                                                NF
·655 IFRND(1)>.5THENA$="BRAVO[3"."]":GOTO
 665                                                                OA
·660 A$="CHARLEY[3"."]"                                             OC
·665 PRINTRX$"[YELLOW][DOWN]THIS":PRINTRX
 $"IS":PRINTRX$;A$:Z0=PEEK(161)+2                                   OC
·670 PRINT"[WHITE]"TM$;TI$;:ON-(PEEK(161)
 <>Z0)GOTO670:PRINTDN$;                                             KJ
·675 FORX=1TO12:PRINTRX$"[11" "]":NEXT:IF
 DOTHENGOSUB1030                                                    MI
·677 RETURN                                                         IM
·680 IFA$="-"THEN715                                                JB
·685 IFF2>=5THENRETURN                                              NN
·690 F1=F1+10:IFF1>255THENF1=0:F2=F2+1                              DB
·695 IFGDTHENGS=GS+1:Q0=0:GOTO710                                   HC
·700 Q0=Q0+3:IFQ0>120THENQ0=120                                     OI
·705 IFQ0>=ASTHENAS=Q0                                              HB
·710 ER=ER+20:GOTO740                                               HM
·715 IFF2<=.25THEN740                                               CM
·720 IFF2=.25 ANDF1<=235THEN740                                     MF
·725 IFGD=1THENGS=GS-1:AS=0:Q0=0:IFGS<0TH
 ENGS=0                                                             AE
·730 IFGD=0THENQ0=Q0-3:IFQ0<0THENQ0=0                               PF
·735 F1=F1-10:ER=ER-20:IFF1<0THENF1=255:F
 2=F2-1                                                             OO
·740 POKES,F1:POKES+1,F2:PRINT"[RED]"AS$;
 AS;S$;GS$;GS;S$;ER$;ER;S$:RETURN                                   GF
·745 IF((EL>-4ANDEL<5)AND(AS<80))THENIF(A
 T<20ANDFP=1)AND(I1=1)THEN790                                       EH
·747 Z7=AT                                                          NI
·750 AT=AT+(SGN(EL)*(.1*AS)+2.5*EL)+2*PC:
 IFAT>32000THENAT=32000                                             BE
·760 AT=AT-2*(FP=0):AT=AT+2*(RU<-30 OR RU
 >30):IFEL=-1THENAT=Z7:AT=AT-INT(2*PC)                              DJ
·762 IFAS<40ANDEL>30THENGOSUB845:GOTO820                            OK
·763 IF(RU>40ORRU<-40)ANDAS>50THENGOSUB86
 6:GOSUB845:GOTO820                                                 GK
·765 IFEL=0THENAT=Z7:T=INT(PC*5):AT=AT+(T
 *(AS<40)):AT=AT-(T*(AS>95))                                        LI
·766 IFAT<Y1ANDAT>Y2THENPOKEBO,1:GOTO770                             IF
·767 POKEBO,FY                                                      PL
·770 HE=HE+((AS/100)*RU):HE=INT(HE):IFHE<
 0THENHE=359                                                        ED
·775 IFHE>359THENHE=0                                               IB
·780 IFAS<30ANDEL=0THENZZ=30-AS:AT=AT-ZZ                            FN
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 113 and 116 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·785 CO=CO+1:MX=AS*(TI/60/60/60):MT=MT+MX GM
/CO:ML=INT(MT):AT=INT(AT):RETURN
·790 GD=1:GS=AS:AS=0:Q0=0:IFABS(RU)>15THE JE
N820
·795 POKEBO,2:AT=0:GOSUB875:RETURN HM
·800 PRINTFL$(I4)" ";:IFGD=0THENFL$(I4)=L DI
EFT$(FL$(I4),LEN(FL$(I4))-1)
·810 IFLEN(FL$(I4))<46THENGOSUB845:GOTO82 JB
0
·815 M2=M1:PRINTFL$(I4)"[WHITE][BACKARROW ME
]":RETURN
·820 PRINT"[HOME][18"[DOWN]"]"RX$"[WHITE] BH
CRASH!!":POKE54290,16:POKE54276,32
·825 IFLEN(FL$(I4))<46THENPRINTRX$"[DOWN] CD
[DOWN]FUEL GONE!"
·830 POKE54287,5 PP
·835 POKE54290,129 ID
·840 FORX=1TO5000:NEXT:RUN NH
·845 AS=300:AT=INT(AT):FORX=5TO0STEP-1:PO MD
KES+1,X:FORY=255TO0STEP-5:POKES,Y
·850 IFAT<=0THENRETURN OL
·855 AT=AT-100:PRINTTM$;TI$;AT$;AT;S$;AS$ EO
;AS;S$
·860 NEXT:IFX=3THENGOSUB870 HK
·865 NEXT:RETURN EJ
·866 POKES+1,F2+1:FORT=1TO600:NEXT:POKES+ MA
1,F2+2:FORT=1TO600:NEXT:POKES+1,F2+3
·867 FORT=1TO600:NEXT:RETURN JL
·870 POKES+18,129:POKES+15,200:RETURN KN
·875 GOSUB880:FORX=1TO500:NEXT:GOTO880 JE
·880 FORX=20TO1STEP-1:POKES+1,X:NEXT:POKE JL
S+4,32:POKES+4,129:FORX=1TO500:NEXT
·885 POKES+4,128:POKES+4,33:RETURN MF
·890 L4=INT(50*RND(1))+16:V=INT(10*RND(1) JE
)+1:V1=D3+V:V2=D3-V:RETURN
·895 FORT=1TO2000:NEXT OB
·900 PRINT"[HOME][11"[DOWN]"][RED]"RX$"EN NA
GINE":PRINTRX$"FAILURE":POKES+18,33
·905 FORX=1TO2000:NEXT:POKES+4,33:RETURN CG
·910 IFHE>V1ORHE<V2THENDG=0:PRINTCL$"[RVS GP
ON][WHITE][3" "][RVSOFF]":RETURN
·915 IFD8<120-ASTHENRETURN LE
·920 PRINTDN$"[5"[DOWN]"]"RX$"COLLISION!" NA
:GOSUB845:GOTO820
·925 DATA169,1,141,25,208,162,69,160,11,1 NH
73,18,208,48,4,162,255,160,2,142,18
·930 DATA208,140,32,208,173,13,220,41,1,2 CF
40,3,76,49,234,76,188,254
·935 GOSUB1030:DO=1 CB
·940 POKEKJ,1:POKEKB,ASC("+"):GOSUB175:IF LP
GS<65THEN940
·950 POKEKJ,1:POKEKB,ASC("U"):GOSUB175:IF GP
EL<29THEN950
·955 POKEKJ,1:POKEKB,ASC("L"):GOSUB175:GO HJ
SUB450:GOSUB450:GOSUB450
·960 POKEKJ,1:POKEKB,ASC("1"):GOSUB175 FB
·965 POKEKJ,1:POKEKB,ASC("D"):GOSUB175:IF AF
EL>4THEN965
·970 D5=0:PX=0 CB
·975 GOSUB175 CP
·980 IFAS<60THENPOKEKJ,1:POKEKB,ASC("+"): KO
GOSUB175:GOTO980
·985 IFDG=1THENRN=INT(10*RND(1)):RU=RU+RN MJ
:GOSUB450:RU=RU-RN
·990 IFD5=100THENPX=1:POKEKJ,1:POKEKB,ASC PF
("D"):GOSUB175:IFEL<>-6THEN990
·995 IFEL>6 THEN EL=EL-1:GOSUB450:GOTO995 IA
·1000 D5=D5+1:IFPX=0THEN975 CE
·1002 GOSUB175:IFAS<40THENPOKEKJ,1:POKEKB MD
,ASC("+"):GOSUB175:GOTO1002
·1003 IFAT>200THEN1002 NO
·1005 POKEKJ,1:POKEKB,ASC("U"):GOSUB175:I AC
FEL<-2THEN1005
·1007 POKEKJ,1: POKEKB,ASC("1"):GOSUB175 KE
·1010 POKEKJ,1:POKEKB,ASC("R") LN
·1015 GOSUB175:IFGD<>1THEN1015 HO
·1020 GOSUB175:POKEKJ,1:POKEKB,ASC("-"):I OM
FGS>0THEN1020
·1025 RU=0:GOSUB1035:PRINTRX$"[UP][UP][4" AO
"]":DO=0:EL=0:GOTO175
·1030 PRINTDN$"[4"[DOWN]"]"RX$"DEMO":PRIN JD
TRX$"FLIGHT":RETURN
·1035 PRINTDN$"[5"[DOWN]"]"RX$"[6" "]":RE NN
TURN
·1200 IFA$>"4"THEN1400 JC
·1202 IFA$="1"THENI1=1-I1:GOSUB1300:GOSUB DP
1220:RETURN
·1205 IFA$="2"THENI2=1-I2:GOSUB1230:RETUR NB
N
·1210 IFA$="3"THENI3=1-I3:GOSUB1240:RETUR NH
N
·1217 IFA$="4"THENGOSUB1350:GOSUB1260:RET HH
URN
·1220 IFI1=1THENPRINTI1$"[RED][s Z][WHITE MD
]":RETURN
·1225 PRINTI1$"[WHITE][s Z]":IFGD=1THEN82 PB
0
·1227 RETURN IM
·1230 IFI2=1THENPRINTI2$"[RED][s Z][WHITE CL
]":RETURN
·1235 PRINTI2$"[WHITE][s Z]":RETURN CM
·1240 IFI3=1THENPRINTI3$"[RED][s Z][WHITE PG
]":GOSUB1310:RETURN
·1250 PRINTI3$"[WHITE][s Z]":GOSUB1330:RE LM
TURN
·1260 IFI4=1THENPRINTI4$"[GREEN][s Z]":RE OM
TURN
·1270 PRINTI4$"[RED][s Z]":RETURN DH
·1300 POKES+11,33:POKES+8,4:FORU=1TO2000: EH
NEXT:POKES+11,32
·1305 POKES+7,F3:POKES+8,F4:POKES+11,17:R NG
ETURN
```


www.commodore.ca

```
·1310 F1=F1-40:ER=ER+80*(ER>0):IFF1<0THEN F1=255:F2=F2-1:IFF2<0THENF2=0 CB
·1320 POKES,F1:POKES+1,F2:RETURN OF
·1330 F1=F1+40:ER=ER+80:IFF1>255THENF1=0:F2=F2+1 LK
·1335 GOTO1320 FL
·1350 PRINTFL$(I4)" ":I4=I4+1:IFI4=3THENI4=1 AG
·1355 PRINTFL$(I4)"[WHITE][BACKARROW]":RETURN KI
·1400 IFA$="5"THENI5=1-I5 HB
·1402 IFA$="6"THENI6=1-I6 GO
·1404 IFA$="7"THENI7=1-I7 ED
·1406 IF I5=1 THEN PRINTDN$RX$"[UP][UP][LEFT][LEFT][RED][c D]":GOTO1410 EN
·1408 PRINTDN$RX$"[LEFT][LEFT][WHITE][UP][UP][c D]" GK
·1410 IFI6=1THENPRINTDN$RX$"[LEFT][LEFT][UP][RED][c D]":GOTO1414 BO
·1412 PRINTDN$RX$"[LEFT][LEFT][UP][WHITE][c D]" OA
·1414 IFI7=1THEN PRINTDN$RX$"[LEFT][LEFT][RED][c D][WHITE]":RETURN PD
·1416 PRINTDN$RX$"[WHITE][LEFT][LEFT][c D]":RETURN OO
```

MEMORY CHECK
FROM PAGE 46

```
·10 REM *** COMMODORE 64 MEMORY CHECK *** NP
·20 PRINT"[CLEAR]LOADING AND VERIFYING DATA[3"."]" DB
·30 FORJ=49152TO49268:READA:POKEJ,A:X=X+A:NEXTJ FP
·40 IFX<>14524THENPRINT"[DOWN]ERROR IN DATA[3"."]":END IN
·50 PRINT"[DOWN]DATA IS OK[3"."]" AH
·60 PRINT"[DOWN]SYS49152 TO BEGIN MEMORY CHECK[3"."]":NEW IJ
·70 DATA32,68,229,169,0,168,153,40,216,200,192,120 EM
·80 DATA144,248,133,251,162,8,134,252,170,168,169,48 AE
·90 DATA141,57,4,141,59,4,169,50,141,58,4,169 AL
·100 DATA32,141,60,4,169,56,141,61,4,140,139,4 BC
·110 DATA152,129,251,193,251,208,50,200,192,0,208,241 EG
·120 DATA230,251,165,251,208,8,165,252,201,159,176,39 EN
·130 DATA230,252,160,22,185,39,4,201,57,176,13,170 LJ
·140 DATA232,138,153,39,4,162,0,160,0,76,45,192 JO
·150 DATA169,48,153,39,4,136,76,76,192,169,2,141 AL
·160 DATA32,208,96,169,5,141,32,208,96 NC
```

THE HAUNTED CASTLE
FROM PAGE 60

```
·0 DIMM,U,D$(3,1),W$(3),W(3),TR(10,1),S(10,1),S$(1):FORA=0TO3:READD$(A,0):NEXT LO
·1 DEFFNR(X)=INT(RND(1)*X):SD=2040:DR=685:S=53248:SO=54272:POKESO+24,15 FH
·2 POKES+39,9:POKES+16,0:DATA"[HOME][4"[DOWN]"][17"[RIGHT]"]","[HOME][13"[DOWN]"][LEFT]" NH
·3 POKESD,245:DATA"[HOME][24"[DOWN]"][17"[RIGHT]"]","[HOME][12"[DOWN]"]" KI
·4 POKES,171:W$(0)=D$(0,0)+"[RVSON][c 4][6" "]":W$(1)=D$(1,0)+"[RVSON][c 4] [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] ":A=RND(-TI) HM
·5 POKES+1,158:W$(2)=D$(2,0)+"[RVSON][c 4][6" "][HOME]":W$(3)=D$(3,0)+"[RVSON][c 4] [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] " DC
·6 POKES+27,1:D$(0,1)=D$(0,0)+"[6" "]":D$(1,1)=D$(1,0)+" [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] " CE
·7 POKES+28,1:D$(2,1)=D$(2,0)+"[6" "][HOME]":D$(3,1)=D$(3,0)+" [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] [DOWN][LEFT] " PG
·8 D$(1,0)=D$(1,0)+"[c 2][RVSON][s -][DOWN][LEFT][s -][DOWN][LEFT][s -][DOWN][LEFT][s -][DOWN][LEFT][s W]":D$(3,0)=D$(3,0)+"[c 2][RVSON][s W][DOWN][LEFT][s -][DOWN][LEFT][s -][DOWN][LEFT][s -][DOWN][LEFT][s -] ME
·9 D$(2,0)=D$(2,0)+"[c 2][RVSON][s W][5"[s *]"][HOME]":D$(0,0)=D$(0,0)+"[c 2][RVSON][5"[s *]"][s W]":POKES+38,7 AC
·10 PRINT"[CLEAR][3"[DOWN]"][7"[RIGHT]"]PLEASE WAIT, READING DATA.":GOSUB96:GOSUB66:GOTO86 GF
·11 PRINT"[CLEAR][4"[DOWN]"][c 4][RVSON][17" "][6"[RIGHT]"][17" "]";:GOSUB157:U=684 FJ
·12 PRINT"[4"[DOWN]"]":GOSUB157:PRINT"[17" "][6"[RIGHT]"][16" "][HOME] BC
·13 POKE2023,160:POKE56295,11:FORA=679TO695:POKEA,0:NEXT:POKEDR,2:POKE687,1 KF
·14 SL=49176:SR=49196:XR=781:M=49750:P=49496:POKES+21,1 NL
·15 GOSUB44:GOSUB39:POKES+30,0:POKES+31,0:GOSUB80:GOSUB81:GOTO79 DK
·16 SYSM:ONPEEK(U)+1GOTO16,17,19,21,23,25,32 AM
·17 Y=Y-1:FORA=0TO7:POKES+1,PEEK(S+1)-1:SYSP:NEXT:POKES+1,255:GOSUB44 NA
·18 FORA=0TO35:POKES+1,PEEK(S+1)-1:SYSP:NEXT:GOSUB39:GOTO79 FC
```


www.commodore.ca

```
·19 X=X+1:FORA=ØTO7:POKEXR,Ø:SYSSR:SYSP:N
EXT:POKES,Ø:POKES+16,Ø:GOSUB44                LL
·2Ø FORA=ØTO35:POKEXR,Ø:SYSSR:SYSP:NEXT:G
OSUB39:GOTO79                                 NE
·21 Y=Y+1:FORA=ØTO24:POKES+1,PEEK(S+1)+1:
SYSP:NEXT:GOSUB44:POKES+1,89                  LG
·22 FORA=ØTO12:POKES+1,PEEK(S+1)+1:SYSP:N
EXT:GOSUB39:GOTO79                            GO
·23 X=X-1:FORA=ØTO7:POKEXR,Ø:SYSSL:SYSP:N
EXT:POKES+16,1:POKES,82:GOSUB44               LG
·24 FORA=ØTO33:POKEXR,Ø:SYSSL:SYSP:NEXT:G
OSUB39:GOTO79                                 FA
·25 ONSS+1GOTO16,26,29                        EP
·26 Z=Z+1:POKESD,25Ø:POKEDR,1:POKE687,1:P
OKES,127:POKES+1,119:FORA=1TO5                HF
·27 POKES,PEEK(S)-16:POKES+1,PEEK(S+1)-8:
FORT=ØTO9:NEXT:NEXT:POKES,Ø:POKES+1,Ø         OB
·28 GOSUB54:SS=Ø:POKES,238:POKES+1,174:GO
TO15                                          KE
·29 POKESD,245:POKEDR,2:POKE687,1:Z=Z-1:P
OKES,238:POKES+1,174:FORA=1TO5                IB
·3Ø POKES+1,PEEK(S+1)+8:B=PEEK(S)+16:IFB=
27ØTHENB=15:POKES+16,PEEK(S+16)OR1            JC
·31 POKES,B:NEXT:GOSUB54:SS=Ø:POKES,127:P
OKES+1,119:POKES+16,Ø:GOTO15                  EF
·32 ONTR+1GOTO16,33,36                        CB
·33 TC=TC+1:GOSUB8Ø:POKES+3Ø,Ø:TR(Z,Ø)=1Ø
:TR(Z,1)=1Ø:GOSUB84:GOSUB153:POKESO+5,8       BP
·34 POKESO+6,255:POKESO+4,23:C=3.5+TC:GOS
UB154:POKES+21,1:POKESO+6,15                  GH
·35 POKES+3Ø,Ø:GOTO16                         II
·36 ON-(TC=Ø)GOTO9Ø:TC=TC-1:POKES+21,1:PO
KE689,Ø:A=FNR(1Ø):B=FNR(1Ø)                   KK
·37 C=FNR(11):IFTR(C,Ø)<>1ØTHEN37             BD
·38 TR(C,Ø)=A:TR(C,1)=B:POKES+3Ø,Ø:GOSUB8
Ø:GOTO16                                      FL
·39 POKES+31,Ø:C=FNR(3):FORA=ØTOC             OD
·4Ø B=FNR(4):ONPEEK(68Ø+B)+1GOTO4Ø            JK
·41 POKE68Ø+B,Ø:CL=CL+1:IFCL=3THENA=C         PM
·42 NEXT:FORA=ØTO3:IFW(A)=1THENW(A)=Ø:NEX
T:RETURN                                      JH
·43 PRINTD$(A,PEEK(68Ø+A)):NEXT:RETURN        PC
·44 POKES+21,1:CL=Ø:FORA=ØTO3:POKE68Ø+A,1
:NEXT:IFX=ØTHENPOKE683,Ø:W(3)=1:CL=CL+1       MI
·45 IFX=9THENPOKE681,Ø:W(1)=1:CL=CL+1         FM
·46 IFY=ØTHENPOKE68Ø,Ø:W(Ø)=1:CL=CL+1         LC
·47 IFY=9THENPOKE682,Ø:W(2)=1:CL=CL+1         HH
·48 FORA=ØTO3:IFW(A)=1THENPRINTW$(A):GOTO
5Ø                                            BB
·49 PRINTD$(A,1)                              PD
·5Ø NEXT:ON-(TC=3)GOSUB76:IFTR=1THENTR=Ø       KN
·51 IFTR(Z,Ø)=XANDTR(Z,1)=YTHEN61             HF
·52 IFTR=2THENTR=Ø:POKE689,Ø                  PC
·53 IFSS=ØTHEN56                              CI
·54 PRINT"[HOME][5"[DOWN]"]";:FORA=1TO19:
PRINT"[RIGHT][38" "][RIGHT]";                 HI
·55 NEXT:PRINT"[HOME]":SS=Ø                   HC
·56 IFZ=ØTHEN59                               OL
·57 IFS(Z,Ø)=XANDS(Z,1)=YTHENPRINTS$(1)ST
$:SS=2:GOTO63                                 GE
·58 IFZ=1ØTHEN63                              BH
·59 IFS(Z+1,Ø)=XANDS(Z+1,1)=YTHENPRINTS$(
Ø)ST$:SS=1                                    PO
·6Ø GOTO63                                    PD
·61 POKES+4Ø,5:POKESD+1,253:POKES+2,173:P
OKES+3,15Ø:POKES+21,3:TR=1:POKE689,Ø          HG
·62 GOTO52                                    PB
·63 IFTR=1ORFNR(1ØØ)>24ØR(X=ØANDY=ØANDZ=Ø
)THENRETURN                                   JI
·64 POKES+2,173:POKES+3,15Ø:POKES+4Ø,1:PO
KESD+1,251:POKES+21,3:POKE689,1:TR=2          LK
·65 RETURN                                    IM
·66 ST$="[RVSON][CYAN]":FORA=1TO6:SB$="[D
OWN]":FORB=1TOA:ST$=ST$+"  ":SB$=SB$+"[L
EFT][LEFT]":NEXT                              MI
·67 ST$=ST$+SB$:NEXT:S$(Ø)="[HOME][5"[DOW
N]"][RIGHT]":S$(1)="[HOME][18"[DOWN]"][R
IGHT]                                         GD
·68 S$(1)=S$(1)+"[12"[RIGHT]"]":ST$=ST$+"
[HOME]":GOSUB75:FORA=1TO3:B=FNR(1Ø)           KM
·69 C=FNR(1Ø):D=FNR(11):IFTR(D,Ø)=BANDTR(
D,1)=CTHENA=A-1:NEXT                          DL
·7Ø TR(D,Ø)=B:TR(D,1)=C:NEXT:FORA=1TO9:S(
A,Ø)=FNR(1Ø):S(A,1)=FNR(1Ø)                   KK
·71 IFS(A,Ø)=TR(A,Ø)ANDS(A,1)=TR(A,1)THEN
A=A-1:NEXT                                    LN
·72 IFS(A,Ø)=TR(A+1,Ø)ANDS(A,1)=TR(A+1,1)
THENA=A-1:NEXT                                GN
·73 IFS(A,Ø)=S(A-1,Ø)ANDS(A,1)=S(A-1,1)TH
ENA=A-1:NEXT                                  IK
·74 NEXT:RETURN                               EJ
·75 FORA=ØTO1Ø:TR(A,Ø)=1Ø:TR(A,1)=1Ø:NEXT
:RETURN                                       NM
·76 IFX<>ØORY<>ØORZ<>ØTHENRETURN              AB
·77 POKES+21,Ø:PRINT"[HOME][7"[DOWN]"][CY
AN]"SPC(12)"CONGRATULATIONS!                  GJ
·78 PRINT"[DOWN][c 1][3"[RIGHT]"]YOU HAVE
 FOUND ALL THE TREASURES!":GOTO93             DJ
·79 PRINT"[HOME][DOWN][c 6][1Ø"[RIGHT]"]R
OOM #"STR$(Y)MID$(STR$(X),2,1)"  LEVEL"Z
"[LEFT] ":GOTO16                              OD
·8Ø PRINT"[HOME][PURPLE][13"[RIGHT]"]TREA
SURES:"TC:RETURN                              HH
·81 US$="NONE":IFZ<1ØTHENUS$="#"+STR$(S(Z
+1,1))+MID$(STR$(S(Z+1,Ø)),2,1)               CD
·82 DS$="NONE":IFZ>ØTHENDS$="#"+STR$(S(Z,
1))+MID$(STR$(S(Z,Ø)),2,1)                    BH
·83 PRINT"[HOME][3"[DOWN]"][7"[RIGHT]"][c
 3]STAIRS: UP-"US$"  DOWN-"DS$                PG
·84 TR$="[RED]NONE":IFTR(Z,Ø)<>1ØTHENTR$=
"[GREEN]#"+STR$(TR(Z,1))+MID$(STR$(TR(Z,
Ø)),2,1)                                      GM
·85 PRINT"[HOME][DOWN][DOWN][BLUE][8"[RIG
HT]"]TREASURE LOCATION: "TR$:RETURN           HA
·86 POKES+32,Ø:POKES+33,Ø:PRINT"[CLEAR][4
"[DOWN]"][PURPLE]"SPC(11)"THE HAUNTED CA
```


www.commodore.ca

```
STLE                                                              EJ
·87 PRINT"[3"[DOWN]"][c 6][4" "]PRESS THE
 [RED]FIRE[c 6] BUTTON WHEN READY[HOME]"
CHR$(142)CHR$(8)                                                  OB
·88 IF(PEEK(56320)AND16)=16THEN88                                 OC
·89 GOTO11                                                        PG
·90 GOSUB153:POKESO+5,8:POKESO+6,255:POKE
SO+4,23:POKESO+15,9:FORA=1TO25                                    JA
·91 POKES+39,FNR(16):POKES+38,FNR(16):GOS
UB156:NEXT:POKESO+6,15:POKES+21,0                                 LE
·92 PRINT"[HOME][7"[DOWN]"]"SPC(11)"[CYAN
]THE GHOST GOT YOU![DOWN]                                         KJ
·93 PRINTSPC(9)"[DOWN][DOWN][YELLOW]PUSH
UP TO PLAY AGAIN":PRINTSPC(11)"[DOWN]PUS
H DOWN TO STOP                                                    PO
·94 A=PEEK(56320):IFA=126THENRUN                                  HP
·95 ON(A=125)+1GOTO94:SYS2048                                     MH
·96 IFPEEK(49759)=96THENRETURN                                    PA
·97 FORA=15680TO16255:READB:POKEA,B:NEXT:
FORA=49152TO49759:READB:POKEA,B:NEXT                              DA
·98 RETURN                                                        IM
·99 DATA0,,,,,,,252,,3,51,,15,255,192,,16
8,,,168,,,168,,,252,,3                                            MB
·100 DATA255,,3,255,192,15,255,224,47,255
,32,32,,,3,255,,3,207,,15,3,192,60                                BA
·101 DATA3,192,42,2,160,,,,,,,144,,,,,,,,
252,,3,51,,15,255,192,                                            AH
·102 DATA168,,,168,,,168,,,252,,3,255,,3,
255,,3,255,,3,239,,,32,,3                                         NL
·103 DATA255,,3,207,,3,207,,3,207,,2,170,
128,,,,,,144,,,,,,,                                               AB
·104 DATA252,,3,51,,15,255,192,,168,,,168
,,,168,,,252,,3,255,,3,255,                                       AK
·105 DATA11,255,,11,255,128,,,128,3,255,,
3,207,,,252,,                                                     AD
·106 DATA252,,,170,,,,,,,144,,,,,,,,63,,
,204,192,3,255,240                                                BO
·107 DATA0,42,,,42,,,42,,,63,,,255,192,,2
55,192,,255,224,2,255,224,2                                       PL
·108 DATA0,,,255,192,,243,192,,63,,,63,,,
170,,,,,,,,144,,,,                                                BI
·109 DATA0,,,63,,,204,192,3,255,240,,42,,
,42,,,42,,,63,,,255,192                                           EB
·110 DATA0,255,192,,255,192,,251,192,,8,,
,255,192,,243,192,,243,192,                                       NK
·111 DATA243,192,2,170,128,.....144.....
,,,63,,,204,192,3,255                                             IE
·112 DATA240,,42,,,42,,,42,,,63,,,255,192
,3,255,192,11,255,240,8,255                                       JA
·113 DATA248,,,8,,255,192,,243,192,3,192,
240,3,192,60,10,128,168,,,,,                                      IL
·114 DATA0,144,,48,,,120,,,252,,1,182,,1,
254,,3,255,,3,255,128,7,255                                       AD
·115 DATA192,7,255,192,15,255,224,15,255,
224,7,255,224,7,255,192,3,255,128,3                               EK
·116 DATA255,,7,252,,15,224,,62,,,,,,,,,,
,,,12,,,30,,,63                                                   AE
·117 DATA0,,109,128,,127,192,,255,192,1,2
55,224,1,255,240,3,255,240,3,255                                  CK
·118 DATA240,3,255,248,3,255,248,1,255,24
8,,255,240,,63,248,,15,252,,,254                                  EL
·119 DATA0,,31,,,,,,,,,,,,,,,,,,7,128,,31
,192,,127,224,1                                                   IJ
·120 DATA255,240,7,255,240,9,255,192,14,2
55,48,22,252,240,27,115,240,29,79,240                             BM
·121 DATA30,63,240,31,127,240,31,127,224,
31,127,192,15,127,,7,124,,3,112,,1                                MI
·122 DATA64,,,,,                                                  NA
·123 DATA1,254,2,253,4,251,8,247,16,239,3
2,223,64,191,128,127,222,1,208,96,254                             NE
·124 DATA1,208,96,189,,208,201,,208,9,173
,16,208,61,1,192,141,16,208,222,                                  MA
·125 DATA208,96,189,,208,201,255,208,9,17
3,16,208,29,,192,141,16,208,254,                                  LF
·126 DATA208,96,162,,142,172,2,142,176,2,
173,31,208,9,254,201,255,208,6,169,5                              KA
·127 DATA141,172,2,96,173,30,208,9,254,20
1,255,208,6,169,6,141,172,2,96,173,                               IC
·128 DATA220,41,15,201,14,240,21,201,13,2
40,53,201,11,240,85,201,7,240,6,169,1                             BH
·129 DATA141,176,2,96,76,18,193,173,1,208
,201,88,240,4,32,16,192,96,173,,208                               KF
·130 DATA201,155,144,234,201,190,144,1,96
,173,168,2,201,1,208,222,169,1,141                                ED
·131 DATA172,2,96,173,1,208,201,223,240,4
,32,20,192,96,173,,208,201,155,144                                LN
·132 DATA198,201,190,144,1,96,173,170,2,2
01,1,208,186,169,3,141,172,2,96,173,                              ME
·133 DATA208,201,29,240,29,173,173,2,201,
2,240,9,169,1,141,173,2,32,24,192,96                              KC
·134 DATA169,250,141,248,7,169,1,141,175,
2,76,218,192,173,16,208,9,254,201,255                             DG
·135 DATA240,218,173,1,208,201,145,144,87
,201,168,144,1,96,173,171,2,201,1,208                             BG
·136 DATA75,169,4,141,172,2,96,173,,208,2
01,59,240,29,173,173,2,201,1,240,9                                MJ
·137 DATA169,2,141,173,2,32,44,192,96,169
,245,141,248,7,169,1,141,175,2,76,32                              NE
·138 DATA193,173,16,208,9,254,201,255,208
,218,173,1,208,201,145,144,17,201,168                             DI
·139 DATA144,1,96,173,169,2,201,1,208,5,1
69,2,141,172,2,96,173,176,2,208,29                                IA
·140 DATA238,174,2,173,174,2,201,5,208,19
,169,,141,174,2,173,173,2,201,1,240                               OA
·141 DATA10,201,2,240,56,76,222,193,76,22
2,193,173,175,2,201,1,240,7,201,2,240                             EA
·142 DATA21,76,222,193,169,1,141,175,2,17
3,248,7,201,248,240,6,206,248,7,76                                PO
·143 DATA222,193,169,2,141,175,2,173,248,
7,201,250,240,226,238,248,7,76,222                                NH
·144 DATA193,173,175,2,201,1,240,7,201,2,
240,21,76,222,193,169,1,141,175,2,173                             JG
·145 DATA248,7,201,247,240,6,238,248,7,76
,222,193,169,2,141,175,2,173,248,7                                BM
·146 DATA201,245,240,226,206,248,7,169,,1
41,31,208,141,30,208,96,173,177,2                                 EE
```


www.commodore.ca

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 113 and 116 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
·147 DATA201,1,208,103,238,178,2,173,178,
 2,201,2,208,93,169,,141,178,2,173,16     BI
·148 DATA208,9,254,201,255,208,9,173,16,2
 08,9,253,201,255,208,39,173,16,208,9     KO
·149 DATA254,201,255,240,9,173,16,208,9,2
 53,201,255,240,8,173,2,208,205,,208      LF
·150 DATA144,13,162,2,32,24,192,169,252,1
 41,249,7,76,64,194,162,2,32,44,192       CN
·151 DATA169,251,141,249,7,173,3,208,205,
 1,208,176,8,162,2,32,20,192,76,85,194    BL
·152 DATA162,2,32,16,192,96,32,231,193,32
 ,64,192,32,88,193,96                     NH
·153 FORA=SOTOSO+23:POKEA,0:NEXT:RETURN   IA
·154 POKESO+15,C:FORB=1TO160STEP2:FORT=0T
 029:NEXT:POKESO+1,B:POKES+40,FNR(16)     DJ
·155 NEXT:RETURN                          EJ
·156 POKESO+1,FNR(256):POKESO,FNR(200):RE
 TURN                                     LB
·157 FORA=1TO7:PRINT"[RVSON] "SPC(38)" ";
 :NEXT:RETURN                             AD
```

KNOCKOUT
FROM PAGE 70

Beginning address in Hex: C000
Ending address in Hex: C61A
SYS to start: 49152

```
C000: A9 00 A0 3F 99 C0 3C 88 A8
C008: 10 FA A0 00 B9 1B C5 99 E7
C010: 00 3D C8 D0 F7 A2 00 8E 10
C018: 0F C5 B9 00 3D F0 16 0A F4
C020: 90 0B 48 AD 0F C5 1D 99 3D
C028: C4 8D 0F C5 68 E8 E0 08 89
C030: D0 ED AD 0F C5 48 98 49 9B
C038: FF A8 68 99 FF 3D 98 49 02
C040: FF A8 C8 D0 D0 A9 80 85 03
C048: FB A9 3D 85 FC A9 FC 85 D9
C050: FD A9 3F 85 FE A9 04 8D F6
C058: 0F C5 A2 00 A0 02 B1 FB 20
C060: 91 FD 88 10 F9 18 A5 FB 3C
C068: 69 03 85 FB A5 FC 69 00 62
C070: 85 FC 38 A5 FD E9 03 85 41
C078: FD A5 FE E9 00 85 FE E8 72
C080: E0 15 D0 D8 E6 FB D0 02 D5
C088: E6 FC C6 FD D0 02 C6 FE C9
C090: CE 0F C5 D0 C5 A0 00 B9 25
C098: E7 C4 99 00 D4 C8 C0 19 56
C0A0: D0 F5 A9 01 8D 1A C5 78 F7
C0A8: A9 7F 8D 0D DC A9 01 8D 81
C0B0: 1A D0 A9 EB 8D 12 D0 A9 4B
C0B8: 1B 8D 11 D0 A9 F1 8D 14 80
C0C0: 03 A9 C3 8D 15 03 58 A9 D8
C0C8: 0B 8D 20 D0 8D 21 D0 A9 7B
C0D0: 00 8D 15 D0 A9 61 A0 C4 B4
C0D8: 20 1E AB 20 E3 C3 8D 04 1C
C0E0: C5 F0 15 A9 74 A0 C4 20 50
C0E8: 1E AB 20 E3 C3 AA BD A1 84
C0F0: C4 8D 05 C5 0A 8D 06 C5 71
C0F8: A9 93 20 D2 FF A9 00 85 58
C100: FD A9 DB 85 FE A2 04 A0 4F
C108: E7 A9 07 91 FD 88 D0 FB 85
C110: 91 FD C6 FE CA D0 F4 A9 9F
C118: 40 A0 1C 99 55 04 99 75 17
C120: 07 88 D0 F7 A9 7D 85 FD 23
C128: A9 04 85 FE A2 13 A9 5D 17
C130: A0 00 91 FD A0 1D 91 FD AD
C138: 18 A5 FD 69 28 85 FD A5 AE
C140: FE 69 00 85 FE CA D0 E6 AF
C148: A9 A0 8D 55 04 8D 72 04 7D
C150: 8D 75 07 8D 92 07 A9 00 2B
C158: A0 03 99 3A D8 88 10 FA 3C
C160: A9 3A 8D 3B 04 A0 03 B9 6E
C168: C9 C4 99 00 C5 B9 CD C4 A2
C170: 99 07 C5 88 10 F1 A9 00 0B
C178: 8D 28 D0 8D 2A D0 A0 02 2A
C180: 99 08 DC 88 10 FA A0 0F 42
C188: 99 0B C5 88 10 FA A9 01 31
C190: 8D 27 D0 8D 29 D0 AE 3C 88
C198: 04 E0 20 F0 F9 A9 0F 8D CE
C1A0: 17 D0 8D 1D D0 8D 15 D0 77
C1A8: 20 D9 C3 20 AE C3 AD 1E C4
C1B0: D0 A2 01 BD 0D C5 F0 0A B0
C1B8: DE 0D C5 BD 0B C5 A8 4C ED
C1C0: E0 C1 8A F0 13 AD 04 C5 69
C1C8: F0 0E AD 10 C5 29 0C 4A CA
C1D0: 4A A8 B9 DB C4 A8 D0 08 9F
C1D8: BD 00 DC 29 0F 49 0F A8 AC
C1E0: BD 00 C5 18 79 B3 C4 C9 38
C1E8: 44 90 07 C9 FD B0 03 9D DD
C1F0: 00 C5 BD 02 C5 18 79 BE 8C
C1F8: C4 C9 47 90 07 C9 BC B0 9D
C200: 03 9D 02 C5 CA 10 AC AD 9D
C208: 1E D0 F0 09 20 3A C3 AD BC
C210: 1E D0 4C 18 C2 8D 17 C5 90
C218: EE 12 C5 AD 12 C5 29 03 90
C220: F0 03 4C E9 C2 A0 01 B9 68
C228: 10 C5 29 03 AA BD DF C4 37
C230: 99 13 C5 BD E3 C4 99 15 B7
C238: C5 88 10 EB AC 11 C5 AD B3
C240: 10 C5 AE 02 C5 EC 03 C5 42
C248: 90 0D 0D 15 C5 48 98 2D DB
C250: 14 C5 A8 68 4C 61 C2 2D D8
C258: 13 C5 48 98 0D 16 C5 A8 A3
C260: 68 AE 00 C5 EC 01 C5 90 81
C268: 0B 09 08 48 98 29 F7 A8 2F
C270: 68 4C 7C C2 29 F7 48 98 66
C278: 09 08 A8 68 8D 10 C5 8C 8A
```


www.commodore.ca

```
C280: 11 C5 A0 01 B9 10 C5 AA 33
C288: 98 F0 1D AD 04 C5 F0 18 AF
C290: AD 03 C5 38 ED 05 C5 CD C5
C298: 02 C5 B0 27 18 6D 06 C5 89
C2A0: CD 02 C5 90 1E 4C AF C2 A3
C2A8: B9 00 DC 29 10 D0 14 8A E7
C2B0: 29 03 18 69 01 29 03 8D 19
C2B8: 0F C5 8A 29 0C 0D 0F C5 2F
C2C0: 4C C6 C2 8A 29 0C 29 0F 8E
C2C8: 99 10 C5 AA BD A3 C4 99 A2
C2D0: 07 C5 B9 0D C5 C9 0A B0 AE
C2D8: 0D 8A 29 08 4A 4A 4A AA 2B
C2E0: BD D1 C4 99 09 C5 88 10 36
C2E8: 9B A2 0E A0 E6 88 D0 FD 14
C2F0: CA 10 F8 AD 1A C5 D0 03 26
C2F8: 4C B1 C1 A9 00 8D 0B D4 CF
C300: 20 D9 C3 A2 01 BD 10 C5 F4
C308: 29 0C A8 B9 A3 C4 9D F8 9E
C310: 07 98 29 08 4A 4A 4A A8 68
C318: B9 D1 C4 9D FA 07 CA 10 E2
C320: E4 A2 17 A0 09 18 20 F0 91
C328: FF A9 82 A0 C4 20 1E AB A3
C330: 20 E4 FF C9 88 D0 F9 4C 9E
C338: CF C0 8D 0F C5 AE 17 C5 B6
C340: D0 0C A2 80 8E 0B D4 E8 97
C348: 8E 0B D4 8E 17 C5 A0 01 C3
C350: B9 0D C5 09 06 99 0D C5 58
C358: B9 10 C5 29 0C 4A 4A AA 5C
C360: BD D7 C4 99 0B C5 88 10 BD
C368: E7 AD 0F C5 29 0C F0 68 61
C370: A2 00 A0 01 AD 0F C5 39 70
C378: D3 C4 F0 2E A9 0F 99 0D 8F
C380: C5 A9 F3 99 09 C5 99 FA E0
C388: 07 FE 18 C5 BD 18 C5 C9 D1
C390: 64 90 17 EE 1A C5 BD D5 FE
C398: C4 A8 A9 0B 99 00 04 A9 02
C3A0: 0F C8 99 00 04 68 68 4C 33
C3A8: FB C2 E8 88 10 C6 A2 00 52
C3B0: A0 0A 18 20 F0 FF A9 01 2F
C3B8: 8D 86 02 AE 18 C5 A9 00 05
C3C0: 20 CD BD A2 00 A0 1C 18 E3
C3C8: 20 F0 FF A9 00 8D 86 02 99
C3D0: AE 19 C5 A9 00 20 CD BD B3
C3D8: 60 A0 15 8C 04 D4 88 8C 69
C3E0: 04 D4 60 20 E4 FF C9 31 1A
C3E8: F0 04 C9 32 D0 F5 29 01 CA
C3F0: 60 A9 01 8D 19 D0 AD 1A 3B
C3F8: C5 D0 59 AD 00 C5 8D 00 E9
C400: D0 8D 04 D0 AD 02 C5 8D 36
C408: 01 D0 8D 05 D0 AD 01 C5 B1
C410: 8D 02 D0 8D 06 D0 AD 03 85
C418: C5 8D 03 D0 8D 07 D0 A0 45
C420: 03 B9 07 C5 99 F8 07 88 CB
C428: 10 F7 AD 0A DC F0 0E C9 8D
C430: 02 90 03 EE 1A C5 29 0F CC
C438: 09 30 8D 3A 04 AD 09 DC D0
C440: AA 29 F0 4A 4A 4A 4A 09 37
C448: 30 8D 3C 04 8A 29 0F 09 12
C450: 30 8D 3D 04 AD 0D DC 29 10
C458: 01 F0 03 4C 31 EA 4C BC BE
C460: FE 8E 08 93 90 50 4C 41 F7
C468: 59 45 52 53 3F 20 28 31 65
C470: 2F 32 29 00 93 4C 45 56 76
C478: 45 4C 3F 20 28 31 2F 32 24
C480: 29 00 50 52 45 53 53 20 58
C488: 46 37 20 46 4F 52 20 41 6F
C490: 20 52 45 4D 41 54 43 48 B6
C498: 00 01 02 04 08 10 20 40 18
C4A0: 80 12 0E F5 FF FE FF F5 2C
C4A8: F6 F7 F6 FA F9 F8 F9 FA 71
C4B0: FC FD FC 00 00 00 00 FF A8
C4B8: FF FF 00 01 01 01 00 FF BB
C4C0: 01 00 00 FF 01 00 00 FF C2
C4C8: 01 46 FA 49 B9 F5 FA F4 F3
C4D0: FB F4 FB 04 08 0A 1C 06 F5
C4D8: 05 0A 09 05 06 09 0A FB 0B
C4E0: FF FF FF 04 00 00 00 00 E4
C4E8: 8F 00 00 00 01 0A 00 29 AC
C4F0: 00 00 00 22 00 00 20 00 33
C4F8: 00 00 00 00 00 00 00 0F 08
C500: 00 00 00 00 00 00 00 00 00
C508: 00 00 00 00 00 00 00 00 08
C510: 00 00 00 00 00 00 00 00 10
C518: 00 00 00 00 00 00 00 00 18
C520: 00 00 00 00 00 00 00 00 20
C528: 00 00 00 00 00 00 00 00 28
C530: 00 00 00 00 00 00 00 10 40
C538: 00 00 0C 00 00 10 00 00 54
C540: 00 00 00 00 00 00 00 00 40
C548: 00 00 00 00 00 00 00 00 48
C550: 00 00 00 00 00 00 00 00 50
C558: 00 00 00 00 7C 00 00 FE D3
C560: 00 7F FE 00 7F FE 00 3C 99
C568: F8 00 1E 7C 00 0F 00 00 0B
C570: 0F 00 00 1F E0 00 3F E0 9F
C578: 00 3F F0 00 3F E0 00 1F E7
C580: E0 00 0F 00 00 0F 00 00 7F
C588: 1E 7C 00 3C F8 00 7F FE D6
C590: 00 7F FE 00 00 FE 00 00 0E
C598: 7C 00 00 00 00 00 00 03 18
C5A0: E0 00 07 F0 01 FF F0 03 6E
C5A8: FF F0 07 87 C0 0F 03 E0 DB
C5B0: 0F 00 00 1F E0 00 3F E0 DF
C5B8: 00 3F F0 00 3F E0 00 1F 28
C5C0: E0 00 0F 00 00 0F 00 00 BF
C5C8: 1E 7C 00 3C F8 00 7F FE 17
C5D0: 00 7F FE 00 00 FE 00 00 4E
C5D8: 7C 00 00 00 00 00 00 00 55
C5E0: 00 00 00 3E 00 00 7F 00 9E
C5E8: 3F FF 00 FF FF 03 E0 7C 88
C5F0: 07 C0 3E 1F E0 00 3F E0 17
C5F8: 00 3F F0 00 3F E0 00 1F 68
C600: E0 00 0F 00 00 0F 00 00 FE
C608: 1E 7C 00 3C F8 00 7F FE 56
C610: 00 7F FE 00 00 FE 00 00 8D
C618: 7C 00 00 94
```


www.commodore.ca

SKETCHER FROM PAGE 92

An assembler is required for entry of this program!
See introductory article beginning on page 92.

```
1 *
2 * SKETCHER
3 *
4           ORG   $8000
5 *
6 COLOR     EQU   $10
7 BASE      EQU   $2000
8 SCROLY    EQU   $D011
9 VMCSB     EQU   $D018
10 COLMAP   EQU   $0400
11 *
12 HMAX     EQU   320
13 VMAX     EQU   200
14 HMID     EQU   160
15 VMID     EQU   100
16 *
17 SCRLEN   EQU   8000
18 MAPLEN   EQU   1000
19 *
20 TEMPA    EQU   $FB
21 TEMPB    EQU   TEMPA+2
22 *
23 TABPTR   EQU   TEMPA
24 TABSIZ   EQU   $9000
25 *
26 HPSN     EQU   TABSIZ+2
27 VPSN     EQU   HPSN+2
28 CHAR     EQU   VPSN+1
29 ROW      EQU   CHAR+1
30 LINE     EQU   ROW+1
31 BYTE     EQU   LINE+1
32 BITT     EQU   BYTE+2
33 *
34 MPRL     EQU   BITT+1
35 MPRH     EQU   MPRL+1
36 MPDL     EQU   MPRH+1
37 MPDH     EQU   MPDL+1
38 PRODL    EQU   MPDH+1
39 PRODH    EQU   PRODL+1
40 *
41 FILVAL   EQU   PRODH+1
42 JSV      EQU   FILVAL+1
43 *
44 CIAPRA   EQU   $DC00
45 *
46          JMP   START
47 *
48 * BLOCK FILL ROUTINE
49 *
50 BLKFIL   LDA   FILVAL
51          LDX   TABSIZ+1
52          BEQ   PARTPG
53          LDY   #0
54 FULLPG   STA   (TABPTR),Y
55          INY
56          BNE   FULLPG
57          INC   TABPTR+1
58          DEX
59          BNE   FULLPG
60 PARTPG   LDX   TABSIZ
61          BEQ   FINI
62          LDY   #0
63 PARTLP   STA   (TABPTR),Y
64          INY
65          DEX
66          BNE   PARTLP
67 FINI     RTS
68 *
69 * 16-BIT MULTIPLICATION ROUTINE
70 *
71 MULT16   LDA   #0
72          STA   PRODL
73          STA   PRODH
74          LDX   #17
75          CLC
76 MULT     ROR   PRODH
77          ROR   PRODL
78          ROR   MPRH
79          ROR   MPRL
80          BCC   CTDOWN
81          CLC
82          LDA   MPDL
83          ADC   PRODL
84          STA   PRODL
85          LDA   MPDH
86          ADC   PRODH
87          STA   PRODH
88 CTDOWN   DEX
89          BNE   MULT
90          RTS
91 *
92 * PLOT ROUTINE
93 *
94 * ROW=VPSN/8 (8-BIT DIVIDE)
95 *
96 PLOT      LDA   VPSN
97          LSR   A
98          LSR   A
99          LSR   A
100         STA   ROW
101 *
102 * CHAR=HPSN/8 (16-BIT DIVIDE)
103 *
104         LDA   HPSN
105         STA   TEMPA
106         LDA   HPSN+1
107         STA   TEMPA+1
108         LDX   #3
109 DLOOP    LSR   TEMPA+1
110         ROR   TEMPA
111         DEX
112         BNE   DLOOP
113         LDA   TEMPA
114         STA   CHAR
115 *
116 * LINE=VPSN AND 7
117 *
118         LDA   VPSN
119         AND   #7
120         STA   LINE
121 *
122 * BITT=7-(HPSN AND 7)
123 *
124         LDA   HPSN
125         AND   #7
126         STA   BITT
127         SEC
128         LDA   #7
129         SBC   BITT
130         STA   BITT
131 *
132 * BYTE=BASE+ROW*HMAX+8*CHAR+LINE
133 *
134 * FIRST MULTIPLY ROW * HMAX
135 *
136         LDA   ROW
137         STA   MPRL
138         LDA   #0
139         STA   MPRH
140         LDA   #<HMAX
141         STA   MPDL
142         LDA   #>HMAX
143         STA   MPDH
144         JSR   MULT16
145         LDA   MPRL
146         STA   TEMPA
147         LDA   MPRL+1
148         STA   TEMPA+1
149 *
150 * ADD PRODUCT TO BASE
151 *
152         CLC
153         LDA   #<BASE
154         ADC   TEMPA
155         STA   TEMPA
156         LDA   #>BASE
157         ADC   TEMPA+1
158         STA   TEMPA+1
159 *
160 * MULTIPLY 8 * CHAR
161 *
162         LDA   #8
163         STA   MPRL
164         LDA   #0
165         STA   MPRH
166         LDA   CHAR
167         STA   MPDL
168         LDA   #0
169         STA   MPDH
170         JSR   MULT16
171         LDA   MPRL
172         STA   TEMPB
173         LDA   MPRH
174         STA   TEMPB+1
175 *
176 * ADD LINE
177 *
178         CLC
179         LDA   TEMPB
180         ADC   LINE
181         STA   TEMPB
182         LDA   TEMPB+1
183         ADC   #0
184         STA   TEMPB+1
185 *
186 * BYTE = TEMPA + TEMPB
187 *
188         CLC
189         LDA   TEMPA
190         ADC   TEMPB
191         STA   TEMPB
192         LDA   TEMPA+1
193         ADC   TEMPB+1
194         STA   TEMPB+1
195 *
196 * POKE BYTE,PEEK(BYTE)OR2^BIT
197 *
198         LDX   BITT
199         INX
200         LDA   #0
201         SEC
202 SQUARE    ROL
203         DEX
204         BNE   SQUARE
205         LDY   #0
206         ORA   (TEMPB),Y
207         STA   (TEMPB),Y
208         RTS
209 *
210 * MAIN ROUTINE STARTS HERE
211 *
212 * FIRST DEFINE BIT MAP AND ENABLE
213 * HIGH-RESOLUTION GRAPHICS
214 *
215 START     LDA   #$18
216         STA   VMCSB
217 *
218         LDA   SCROLY
219         ORA   #32
220         STA   SCROLY
221 *
222 * SELECT GRAPHICS BANK 1
223 *
224         LDA   $DD02
225         ORA   #$03
226         STA   $DD02
227 *
228         LDA   $DD00
229         ORA   #$03
230         STA   $DD00
231 *
232 * CLEAR BIT MAP
233 *
234         LDA   #0
235         STA   FILVAL
236         LDA   #<BASE
237         STA   TABPTR
```


www.commodore.ca

```
238             LDA   #>BASE
239             STA   TABPTR+1
240             LDA   #<SCRLEN
241             STA   TABSIZ
242             LDA   #>SCRLEN
243             STA   TABSIZ+1
244             JSR   BLKFIL
245 *
246 * SET BKG AND LINE COLORS
247 *
248             LDA   #COLOR
249             STA   FILVAL
250             LDA   #<COLMAP
251             STA   TABPTR
252             LDA   #>COLMAP
253             STA   TABPTR+1
254             LDA   #<MAPLEN
255             STA   TABSIZ
256             LDA   #>MAPLEN
257             STA   TABSIZ+1
258             JSR   BLKFIL
259 *
260 * PRINT DOT AT MIDSCREEN
261 *
262             LDA   #VMID
263             STA   VPSN
264             LDA   #<HMID
265             STA   HPSN
266             LDA   #>HMID
267             STA   HPSN+1
268             JSR   PRINT
269 *
270 * READ JOYSTICK
271 *
272 * FIRST CHECK TRIGGER BUTTON
273 *
274 READJS      LDA   CIAPRA
275             AND   #$10
276             BEQ   START
277 *
278 * NOW READ JOYSTICK
279 *
280             LDA   #$0F
281             PHA
282             AND   CIAPRA
283             STA   JSV
284             PLA
285             SEC
286             SBC   JSV
287             STA   JSV
288 *
289             TAX
290             BEQ   READJS
291             LDA   RELADS-1,X
292             STA   MODREL+1
293 MODREL      BNE   *
294 MODR1
295 *
296 NIL1        JMP   READJS
297 *
298 * ROUTINES TO MOVE JOYSTICK
299 *
300 UP          JSR   MOVEUP
301             JMP   DOIT
302 *
303 DOWN        JSR   MOVEDN
304             JMP   DOIT
305 *
306 LEFT        LDX   HPSN
307             LDY   HPSN+1
308             TXA
309             BNE   DECLSB
310             DEY
311 DECLSB      DEX
312             STX   HPSN
313             STY   HPSN+1
314             JMP   DOIT
315 *
316 UPANDL      JSR   MOVEUP
317             JMP   LEFT
318 *
319 DNANDL      JSR   MOVEDN
320             JMP   LEFT
321 *
322 NIL2        JMP   READJS
323 *
324 RIGHT       LDX   HPSN
325             LDY   HPSN+1
326             INX
327             BNE   NOINC
328             INY
329 NOINC       STX   HPSN
330             STY   HPSN+1
331             JMP   DOIT
332 *
333 UPANDR      JSR   MOVEUP
334             JMP   RIGHT
335 *
336 DNANDR      JSR   MOVEDN
337             JMP   RIGHT
338 *
339 * SUBROUTINES TO MOVE UP & DOWN
340 *
341 MOVEUP      LDX   VPSN
342             DEX
343             STX   VPSN
344             RTS
345 *
346 MOVEDN      LDX   VPSN
347             INX
348             STX   VPSN
349             RTS
350 *
351 * "DOIT" SUBROUTINE
352 *
353 DOIT        JSR   PRINT
354             JMP   READJS
355 *
356 * MORE SUBROUTINES START HERE
357 *
358 * MAKE SURE DOT IS WITHIN RANGE
359 *
360 CHECK       LDA   VPSN
361             BEQ   RAISE
362             CMP   #VMAX-1
363             BCS   LOWER
364             JMP   HCHECK
365 RAISE       INC   VPSN
366             JMP   HCHECK
367 LOWER       LDA   #VMAX-1
368             STA   VPSN
369 *
370 HCHECK      BIT   HPSN+1
371             BPL   OKLOW
372             LDA   #1
373             STA   HPSN
374             LDA   #0
375             STA   HPSN+1
376             RTS
377 *
378 OKLOW       LDA   #<HMAX-2
379             CMP   HPSN
380             LDA   #>HMAX-2
381             SBC   HPSN+1
382             BCC   TOOHI
383             RTS
384 *
385 TOOHI       LDA   #<HMAX-2
386             STA   HPSN
387             LDA   #>HMAX-2
388             STA   HPSN+1
389             RTS
390 *
391 * PRINT DOT ON SCREEN
392 *
393 PRINT       JSR   CHECK
394             JSR   PLOT
395 *
396             LDA   HPSN
397             PHA
398             LDA   HPSN+1
399             PHA
400 *
401             LDA   HPSN
402             BNE   SKIP
403             DEC   HPSN+1
404 SKIP        DEC   HPSN
405             JSR   CHECK
406             JSR   PLOT
407 *
408             PLA
409             STA   HPSN+1
410             PLA
411             STA   HPSN
412             RTS
413 *
414 RELADS      DFB   UP-MODR1
415             DFB   DOWN-MODR1
416             DFB   NIL1-MODR1
417             DFB   LEFT-MODR1
418             DFB   UPANDL-MODR1
419             DFB   DNANDL-MODR1
420             DFB   NIL2-MODR1
421             DFB   RIGHT-MODR1
422             DFB   UPANDR-MODR1
423             DFB   DNANDR-MODR1
424 *
```

ALARM CLOCK

FROM PAGE 99

```
·100 POKE56,158:CLR                                          KA
·110 PRINT"[CLEAR][11"[DOWN]"][10"[RIGHT]
 "]ONE MOMENT PLEASE[3"."]"                                  HA
·120 FORAD=40449TO40739:READOP:CK=CK+OP:P
 OKEAD,OP:NEXT                                               PG
·130 IFCK<>31161THENPRINT"[CLEAR]ERROR IN
  DATA STATEMENTS.":STOP                                     NG
·140 PM=0:INPUT"[CLEAR]WHAT IS THE HOUR";
 H:IFH<0ORH>23THEN140                                        HG
·150 IFH=0THENH=12:GOTO180                                   CP
·160 IFH>12THENH=H-12:PM=-1                                  IH
·170 IFPM=0THENGOSUB420                                      NO
·180 IFH=12THENPM=NOT PM                                     DE
·190 NUM=H:GOSUB370:POKE56331,-128*PM+16*
 FD+SD                                                       KN
·200 INPUT"[DOWN]WHAT IS THE MINUTE";M:IF
 M<0ORM>59THEN140                                            NO
·210 NUM=M:GOSUB370:POKE56330,16*FD+SD                       EG
·220 INPUT"[DOWN]WHAT IS THE SECOND";S:IF
 S<0ORS>59THEN140                                            BE
```


www.commodore.ca

```
·230 NUM=S:GOSUB370:POKE56329,16*FD+SD                    AE
·240 PRINT"[DOWN][DOWN]CONTINUOUS TIME DI
 SPLAY? ([RVSON]Y[RVSOFF]/[RVSON]N[RVSOFF
 ])"                                                      PB
·250 GETA$:IFA$=""THEN250                                 HK
·260 IFA$="N"THENPOKE40739,0                              PM
·270 PM=0:INPUT"[CLEAR]WHAT IS THE ALARM
 HOUR";AH:IFAH<0ORAH>23THEN270                            HB
·280 IFAH>12THENAH=AH-12:PM=-1                            DC
·290 IFPM=0THENGOSUB420                                   NO
·300 NUM=AH:GOSUB370:POKE40737,-128*PM+16
 *FD+SD                                                   ME
·310 INPUT "[DOWN]WHAT IS THE ALARM MINUT
 E";AM:IFAM<0ORAM>59THEN270                               LC
·320 NUM=AM:GOSUB370:POKE40738,16*FD+SD                   PH
·330 PRINT"[DOWN][DOWN]AUDIO ALARM? ([RVS
 ON]Y[RVSOFF]/[RVSON]N[RVSOFF])"                          KN
·340 GETA$:IFA$=""THEN340                                 HK
·350 IFA$="N"THENPOKE40680,0                              OP
·360 SYS40449:PRINT"[CLEAR]":END                          PL
·370 REM ** SUBROUTINE                                    CJ
·380 A$=STR$(NUM):IFLEN(A$)=3THENFD=VAL(M
 ID$(A$,2,1)):GOTO400                                     NJ
·390 FD=0                                                 HB
·400 SD=VAL(RIGHT$(A$,1))                                 EL
·410 RETURN                                               IM
·420 REM ** SUBROUTINE                                    CJ
·430 PRINT"[DOWN][RVSON]A[RVSOFF]M OR [RV
 SON]P[RVSOFF]M"                                          CC
·440 GETA$:IFA$=""THEN440                                 HH
·450 IFA$="P"THENPM=-1                                    AK
·460 RETURN                                               IM
·40449 DATA120,173,20,3,73,40                             PD
·40455 DATA141,20,3,173,21,3                              OB
·40461 DATA73,116,141,21,3,88                             OG
·40467 DATA169,0,141,8,220,96                             CN
·40473 DATA173,35,159,240,121,160                         BH
·40479 DATA31,173,32,208,153,0                            DN
·40485 DATA216,200,192,40,208,248                         GL
·40491 DATA173,11,220,41,16,74                            IO
·40497 DATA74,74,74,9,176,201                             CB
·40503 DATA176,208,2,169,160,141                          KG
·40509 DATA31,4,173,11,220,41                             BC
·40515 DATA15,9,176,141,32,4                              OF
·40521 DATA173,11,220,41,128,240                          PE
·40527 DATA8,169,144,141,39,4                             PB
·40533 DATA76,93,158,169,129,141                          HO
·40539 DATA39,4,169,186,141,33                            GH
·40545 DATA4,173,10,220,41,112                            HH
·40551 DATA74,74,74,74,9,176                              ME
·40557 DATA141,34,4,173,10,220                            LB
·40563 DATA41,15,9,176,141,35                             NM
·40569 DATA4,169,186,141,36,4                             EM
·40575 DATA173,9,220,41,112,74                            HE
·40581 DATA74,74,74,9,176,141                             CA
·40587 DATA37,4,173,9,220,41                              NE
·40593 DATA15,9,176,141,38,4                              AL
·40599 DATA173,11,220,41,159,205                          JN
·40605 DATA33,159,208,100,173,10                          OL
·40611 DATA220,41,127,205,34,159                          IP
·40617 DATA208,90,173,2,159,208                           IO
·40623 DATA36,169,181,141,0,212                           NB
·40629 DATA169,23,141,1,212,169                           EK
·40635 DATA0,141,5,212,169,240                            AC
·40641 DATA141,6,212,169,33,141                           JE
·40647 DATA4,212,173,32,208,141                           IG
·40653 DATA3,159,169,255,141,2                            LG
·40659 DATA159,206,32,159,208,38                          JB
·40665 DATA174,32,208,236,3,159                           LJ
·40671 DATA208,14,173,4,159,141                           EK
·40677 DATA32,208,169,15,141,24                           HK
·40683 DATA212,76,250,158,173,3                           EC
·40689 DATA159,141,32,208,169,0                           KJ
·40695 DATA141,24,212,169,30,141                          ON
·40701 DATA32,159,76,26,159,0                             NK
·40707 DATA0,10,173,2,159,240                             PP
·40713 DATA16,173,3,159,141,32                            GP
·40719 DATA208,169,0,141,24,212                           DM
·40725 DATA169,0,141,2,159,173                            HI
·40731 DATA8,220,76,49,234,30                             CM
·40737 DATA0,0,255                                        OE
```

BASIC RELOCATOR
FROM PAGE 30

```
·1 SYS2063:END:REM"[61"A"]                                JE
·2 REMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 AAAAAAAAAAAAAAA                                          KE
·900 REM                                                  JD
·910 REM SAVE THIS PROGRAM TO DISK BEFORE
  RUNNING IT                                              OB
·920 REM MAKE SURE THE REM STATEMENTS ARE
  PACKED WITH THE PROPER AMOUNT OF A'S                    LC
·930 REM TO RUN, TYPE 'RUN1000'.
  NOT 'RUN'                                               NF
·940 REM                                                  JD
·1000 POKE2049,133:CK=0: FORX=2063TO2182:
  READ A: CK=CK+A: POKEX,A: NEXT                          GM
·1010 IF CK<>11741 THEN PRINT"ERROR IN DA
 TA STATEMENTS": STOP                                     DA
·1020 DATA 172,122,160,56,173,130,8,133,4
 4,233,8,133,25,24,165,45                                 OH
·1030 DATA 133,27,133,29,165,46,133,28,10
 1,25,133,30,133,46,165,27                                JO
·1040 DATA 208,4,198,28,198,30,198,27,198
 ,29,177,27,145,29,165,27                                 DA
·1050 DATA 208,244,165,28,201,8,208,230,2
 30,29,177,29,200,17,29,240                               PH
·1060 DATA 21,24,177,29,101,25,145,29,133
 ,2,136,177,29,133,29,165                                 PJ
·1070 DATA 2,133,30,76,73,8,174,110,160,1
 42,120,2,142,123,2,162                                   JO
·1080 DATA 49,142,119,2,162,82,142,121,2,
 162,117,142,122,2,162,5                                  HA
·1090 DATA 134,198,96                                     JO
·1100 DATA 16: REM PAGE TO TRANSFER TO                    FA
·1110 DATA 34,0,0,0                                       AJ
```


www.commodore.ca

INFRARAID

Continued from page 112

(Note that the last zero in line 10 is printed in white, indicating where TXTPTR is pointing.)

Try other examples such as

```
10 A=X/0

10 A=A$

10 ?:?CHR$(-1):?
```

A special case that needs to be discussed further is that of the READ-DATA statements. Try these examples and note any differences in how they are handled:

```
10 READ A

10 READ A
20 PRINT

10 READ A
20 DATA "STRING"
```

In the first example, READ is being pointed at by TXTPTR; therefore it is displayed in white. In the second example, however, READ is not displayed in white because TXTPTR does not point to it. What happens is that BASIC searches the entire text for DATA statements and if one is not found, TXTPTR is pointing to the end of the program, but CURLIN, the current BASIC text line number, is still pointing to the line where the READ originated. When *Infraraid* prints the current line it compares the address of the character it is printing to TXTPTR; when, and only then, will that character be printed in white. In the example it never matched the addresses, so no character in that line was highlighted. This is always the case unless the READ statement is on the last text line in a program and there are no DATA statements.

In the third example above the DATA line is displayed, since the error occurred in that line rather than the READ line. The DATA item searched for is numeric, and since the first character in the DATA line is non-numeric, an error was generated. Note that BASIC displayed a ?SYNTAX ERROR even though technically it should have been a ?TYPE MISMATCH ERROR. Also note that this could be quite handy debugging programs with large amounts of data where there is both numeric and non- numeric data, since *Infraraid* shows you exactly which data caused the error.

This feature of pinpointing errors alone can be quite helpful in debugging programs, but *Infraraid* can also trap some errors within the program. The problem I had with my calculator simulation program was not a program bug but a system limitation. I had to keep the program from crashing when handling larger numbers than BASIC would normally allow. *Infraraid* lets you do that using the trap variable TR% (refer to Table 2). Try this example:

```
10 TR%=1
20 PRINT 10↑100
```

Notice that the number 332.192809 was printed rather than an error message. What *Infraraid* did was check the variable TR% when the error was generated in line 20. Since the first bit in TR% was set, *Infraraid* passed control back to the BASIC routine instead of recognizing the error. The BASIC routine then executed as if no overflow error occurred and it printed the number it had previously calculated. Note that the displayed number is garbage and must be treated as such. When trapping errors in this manner, always check ER% to determine whether an error occurred or not. When a program is run, ER% is set to 0; however, when ER% is set by an error, the program must reset it to 0. In the above example, for instance, you could add:

```
30 IF ER%=15 THEN ER%=0:GOTO (wherever
you want)
```

In addition to trapping certain errors, *Infraraid* also lets the program jump to its own error handling routine when it encounters a trapped error. The programmer simply specifies the line number to jump to in EL% (see Table 3). When *Infraraid* encounters an error trapped by TR% it will perform a GOTO of the line number specified in EL%, if EL% is not equal to 0. This example will demonstrate how it works:

```
10 EL%=1000:REM ERROR ROUTINE STARTS AT
1000
20 TR%=15:REM TRAP ALL POSSIBLE ERRORS
30 PRINT 10 100:REM TRY DIFFERENT ERROR
S IN THIS LINE
40 END
1000 REM ERROR ROUTINE
1010 IF ER%=4 THEN PRINT "FILE NOT FOUN
D"
1020 IF ER%=5 THEN PRINT "DEVICE NOT PR
ESENT"
1030 IF ER%=15 THEN PRINT "OVERFLOW"
1040 IF ER%=20 THEN PRINT "DIVISION BY
ZERO"
1050 ER%=0:REM MUST RESET ER%
1060 GOTO 40
```

As a final note, the RUN/STOP-RESTORE sequence will not disable *Infraraid*. If you wish to do this without reseting the computer use:

```
POKE 768,PEEK(50657):POKE 769,PEEK(5065
8)
```

Infraraid was written with the *Commodore 64 Macro Assembler Development System.* □

SEE PROGRAM LISTING ON PAGE 132

TABLE 3
Variable pointer to the line bumber of your BASIC error routine

Range of values for EL%	Range of line numbers pointed to
0	none (deactivated)
1 to 32767	1 to 32767
−32768 to −1537	32768 to 63999
−1536 to −1	64000 to 65535 (illegal)

www.commodore.ca

STREAMER FONT

Continued from page 29

tinue. If you enter a null string the SAVE command is aborted, and you're taken back to the main program. Press SHIFT-L to LOAD your font back into memory.

Although you can't SAVE a character file on tape, you can tack your font on to the end of the program and SAVE it along with your program. If you have a program that accelerates tape LOADs and SAVEs, this method could be an advantage. There is, however, one drawback with amalgamating your program and file. The font memory, which begins at 8192, is 2K above the program. It can't be lowered to follow immediately after the program, as the rotated display would be usurped by your computer's character information. This is because *Streamer Font*'s characters are stored in memory in a manner similar to sprites, and sprites located below 8192 will not be displayed. The upshot of all this is that you must save nearly 2K of useless memory along with your program and file. Even with this excess baggage, your program/file will still only be 8½K long, a reasonable length thanks to the brevity of machine language.

Here's how to wed your program to your font file. First LOAD *Sequela* using *Flankspeed*. Then type in NEW. Now LOAD *Streamer Font* and enter the following POKES:

```
POKE5802,1:POKE45,0:POKE46,42
```

The first POKE will set the switch so that the font memory is not cleared. The other two will move the BASIC end-of-program pointer to the end of the file. If you don't wish to add *Sequela,* just go through the same steps except loading the file into memory. Now SAVE *Streamer Font* as you normally would. Whenever you design a character set with this new program, it will automatically be SAVEd with it.

Finally we get to what *Streamer Font* is all about: printing banners! There are two print defaults. One is the size. If the size is not set, the program automatically prints the smallest size, 2.4 inches high. This can be altered by pressing SHIFT 1-5, where 1 is the smallest size. Press SHIFT-5 and your characters will be printed 7.2 inches high.

The other default is the composition of your characters. If this is not reset your letters will be printed with the numerical symbol, #, on a background of spaces. To change this, press f6. You will then be asked for the foreground character. Enter the character or graphic symbol of your choice. After that, you'll be asked if you wish this character to be printed in reverse. Answer Y or N. Now you'll be queried for a background character. Type this in and you'll be taken back to the main program.

To send your characters to the printer, press f5 and you'll be prompted to enter your message. Like the SAVE and LOAD filenames, you can only edit your input with the DEL key, and you can abort by entering a null string. Up to 33 characters can be entered at one time, although you can easily string several of these together for very long messages. *Streamer Font* will automatically account for SPACEs in your messages providing that the background character is a space. Since some printers take the time to print spaces, this will make for faster printing. If, on the other hand, you are using a background character other than CHR$(32), you must use an empty character from your font. The left margin, incidentally, is adjusted so that your message appears centered. Once your message is entered, printing begins. When it comes to line spacing, no spaces will be printed between the lines on Commodore printers. If it's necessary to stop printing, you can do so by holding down the RUN/STOP key.

SEQUELA

By the time you've finished typing in 4K of machine language, you may balk at entering another 2.5K. Even so, *Streamer Font* would be incomplete if I didn't include a character set. You'll like *Sequela* as it's an attractive font that can be used for nearly any message. And once you've entered and SAVEd it, you can easily alter it to suit your whims. To enter *Sequela* you must again use *Flankspeed,* but unlike *Streamer Font,* there's no need to change any pointers since it will sit well above *Flankspeed.* LOAD and RUN *Flankspeed* and enter in the hexadecimal addresses.

Like cassette users, disk users will have to follow a few steps to get *Sequela* into the font memory. First use *Flankspeed* to LOAD *Sequela* into memory. Type in NEW and LOAD *Streamer Font*. Now POKE5802,1. Enter RUN and you should see a large A and B on the screen. Press SHIFT-S to SAVE *Sequela* to disk. □

SEE PROGRAM LISTING ON PAGE 124

...COMING IN THE FEBRUARY *AHOY!* (ON SALE JANUARY 7)...

TURNKEY 64
BUILD AN AUTO-EXEC CARTRIDGE FOR YOUR 64

THE DIGITAL BLUE YONDER
SURVEY OF C-64 FLIGHT SIMULATIONS

ARTIFICIAL INTELLIGENCE
LET YOUR COMPUTER TEACH ITSELF

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.

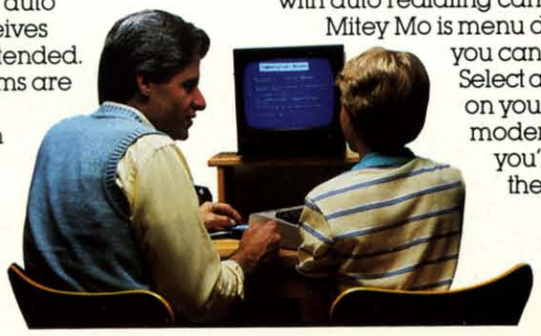
Mitey Mo has just one switch, the Smart 64 software does the rest. With the other modem you'll have to remember to check three switches, otherwise you may be answering when you mean to be originating.

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. 257

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. 136

www.commodore.ca

DON'T MISS THE BOAT!

SUBSCRIBE TO AHOY!

☐ Send me twelve issues of AHOY! for $19.95 ($26.95 Canada & elsewhere). I understand that I will be billed later.

☐ Send me twenty-four issues of AHOY! for $37.95 ($49.95 Canada and elsewhere). I understand that I will be billed later.

NAME_____

ADDRESS_____

CITY_____

STATE_____ ZIP_____

DON'T MISS THE BOAT!

SUBSCRIBE TO THE AHOY! DISK MAGAZINE

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

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

NAME_____

ADDRESS_____

CITY_____

STATE_____ ZIP_____

www.commodore.ca

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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

POSTAGE WILL BE PAID BY ADDRESSEE

Ahoy!

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

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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

POSTAGE WILL BE PAID BY ADDRESSEE

Ahoy!

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

DON'T MISS THE BOAT!

SUBSCRIBE TO AHOY!

☐ Send me twelve issues of AHOY! for $19.95 ($26.95 Canada & elsewhere). I understand that I will be billed later.

☐ Send me twenty-four issues of AHOY! for $37.95 ($49.95 Canada & elsewhere). I understand that I will be billed later.

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

January 1986 *Ahoy!*/Void after January 14, 1986

READER SERVICE CARD

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

101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130

131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160

161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190

191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 210

211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240

241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270

271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300

A. Please check whether you are...

1. ☐ male 2. ☐ female

B. What is your age?

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

C. Education level completed

1. ☐ elementary 2. ☐ high school
3. ☐ junior college 4. ☐ college graduate
5. ☐ masters degree 6. ☐ phd

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

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

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

1. ☐ yes 2. ☐ no

Name_____

Street_____

City_____ State_____ Zip_____

www.commodore.ca

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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

POSTAGE WILL BE PAID BY ADDRESSEE

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

PLACE
STAMP
HERE

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

www.commodore.ca

Reserve my FREE Hour and Rush my Viewtron® Starter Kit

To order, mail this card or call anytime: 1 (800) 543-5500, Dept. 9015.

Name _____
(Please print)

Address _____

City/State/Zip _____

Phone Number (_____) _____

Credit cards only; sorry, no checks.

Charge to: ☐ VISA ☐ MasterCard ☐ American Express

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ ☐☐ ☐☐

Card Number Expiration Date

Signature: _____

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

Description	Cost	Check Here
Viewtron Starter Kit for Commodore 64 or 128	$ 9.95	☐
MODEM PACKAGE FOR COMMODORE 64 OR 128		
300 Baud Volks 6420 by Anchor Automation with FREE Viewtron Starter Kit	$ 49.95	☐

Satisfaction Guaranteed or your money back. (Direct-connect modem, no cable necessary.) 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 2-4 weeks for delivery. Shipping included. Florida, Illinois and New York residents will have the appropriate sales tax added to their order.

Offers expire February 28, 1986.

www.commodore.ca

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 19696 MIAMI, FL

Postage will be paid by addressee

Viewtron®
The New, Easier-to-Use On-Line Service
Viewdata Corporation of America, Inc.
P.O. Box 317678
Cincinnati, Ohio 45231

www.commodore.ca