```
500 IF K<6 AND Z$(K)=CHR$(13) THEN 300
600 IFK=1 AND Z$(K)=CHR$(20)THEN 300
700 IF Z$(K)=CHR$(20) THEN PRINT CHR$(20)
    ;:K=K-1:GOTO 300
800 IF K=6 AND Z$(K)<>CHR$(13) THEN 300
900 PRINT Z$(K);
1000 NEXT K
```

The GET statement can also be used within a program to allow time for a user to perform an outside action. For example, assume the user is running a program that requires a special form to be loaded on the printer. You might use the following program to give the user time to change printer paper:

```
1000 PRINT"PUT SPECIAL FORM IN PRINTER AN
     D TURN ON PRINTER "
1100 PRINT"PRESS C KEY WHEN YOU ARE READY
      TO CONTINUE"
1200 GET A$: IF A$<>"C" THEN 1200
1300 REM  PROGRAM CONTINUES
```

Now the program will execute line 1200 repeatedly until the C key is pressed.

Pausing With GET

The GET statement can be used to put a pause in a program. For example, say you have a long print job to do, but you want to be able to interrupt it at any time. With the GET statement, you can go through the PRINT statement (I'll just print to the screen here) as long as you want.

When you press the S key, printing will stop. When you press C, it will continue where it left off. This example prints integers from 1 to 50000, stopping when you press S and continuing when you press C:

```
100 FOR I=1 TO 50000
110 GET A$: IF A$="S" THEN 130
120 GOTO 140
130 GET A$: IF A$<>"C" THEN 130
140 PRINT I
150 NEXT I
```

Avoiding The GET Statement

Now that we've seen how to use the GET statement, you should also understand that it isn't always necessary to use it. If you want to make it easy for a user to make a default entry, then you can more easily use the INPUT statement. The technique for doing this is to use cursor controls to print the desired character or characters after the prompt from the INPUT statement. Then pressing RETURN will cause the characters to be entered.

For example, suppose that you want to ask a question for which the answer is either yes or no (Y or N), and that you want Y to be the default entry since you expect it will be the most common response. The lines below show how this might be programmed:

```
100 INPUT"PLAY AGAIN  Y{3 LEFT}";A$
200 IF A$="Y" THEN PRINT"{UP}PLAYING . .
    {SPACE}. ":GOTO100
300 PRINT"GAME OVER":END
```

If you just press RETURN, the Y will automatically be entered. If you want to enter N, simply type it over the Y and press RETURN.

HORIZONS: 64

Charles Brannon
Program Editor

Feedback from owners of the MSD disk drive: Most say they are very pleased with the drive and have had no problems with it. Many of the owners bought their MSD drive because of the unavailability or dissatisfaction with the 1541. Other owners wanted a dual-drive system, and opted for MSD. A few readers are using an IEEE-488 interface with the MSD, and find there is a noticeable though not dramatic speed increase.

One thing I neglected to mention in my earlier column is that the MSD drive will format a disk in only 20 seconds. Reader James Baker tried some benchmarks and found that the MSD would set up a data disk for *The Home Accountant* in seven minutes as compared to eleven minutes for the 1541. The MSD also runs the Check Disk program (found on the 1541 Test/Demo disk) about three times faster than his 1541. By the way, the latest 1541s on the market run much better (cooler, and more quietly) than their predecessors.

We've had many requests for the address of Concorde Peripheral systems, manufacturer of the alleged high-speed C-321P Commodore-compatible disk drive, so here it is:

Concorde Peripheral Systems
23152 Verdugo Drive
Laguna Hills, CA 92653

As this column goes to press, we have yet to receive our review drive for evaluation. We talked with Concorde, and they said they were working out a few software compatibility problems, but that the drive should be available by the time you read this. The drive looks quite promising, and has been advertised for only a few dollars more than the 1541, but I'd like to see one first before buying.

New Commodore Printers

The new Commodore MPS-801 and 1526 printers have also been the subject of a lot of mail. First, the MPS-801. This printer is a direct replacement for the 1525. It costs the same, works the same, prints the same mediocre character set, and is about as slow. The good news is that it appears to be built better, and has a more attractive case. The MPS-801 is completely compatible with all 1525 printer codes, and apparently is not as different internally as it looks on the outside.

It works with both the VIC-20 and the 64. It uses a tiny ribbon cartridge, and the paper feed is more reliable than the 1525. Some readers have had trouble finding a replacement ribbon cartridge. Reader E. Thornlimb says that Radio Shack's DMP 110 is much like the MPS-801, and uses the same ribbon cartridge, which is available for $7.95 (catalog number 26-1283).

I don't know why the printer is called the MPS-801 instead of a number like 1525 or 1526, but that may be the manufacturer's product number. Commodore doesn't design and build its own printers. It OEM's them (OEM stands for Original Equipment Manufacturer, which sounds contradictory), then customizes them with a Commodore interface and operating system—yet keeps the price low, which is rather remarkable for an OEM. Radio Shack obviously does the same thing with their DMP-110.

The 1526 printer appears to be an OEM'ed Epson MX-80, but the character set looks more like the MX-70's (which doesn't seem to be around anymore). It has a very good looking character set, especially if you use the carbon film ribbon (though the carbon ribbon is used up quickly). You may remember that the 1526 printer was released earlier, then recalled due to serial interface problems (it would sometimes lock up the serial bus, preventing disk access). The current 1526 still has an occasional serial lock-up when used with the 64, but there are serious serial bus problems with the VIC-20. The box it came in was stamped FOR 64 ONLY, which seems to be a quick fix for the problem.

Even 64 owners have had problems with the 1526, and for good reason. The 1526 is a Commodore-compatible printer, but it isn't compatible with the 1525. The 1526 seems to be a Commodore 4022 printer with a serial interface. The 2022 and 2023 printers were the Commodore standard during the era of the PET/CBM 2001, 3000, 4000, and 8000 series computers (such as the 8032 and SuperPET). A 4022 printer replaced the 2022 with an MX-80 type printer, but still honored all the 2022 codes and standards. This 4022 seems to be now marketed as the 1526 for 64 owners, but is still the 4022 internally. If the 2022 standard was carried over to the 1525, all would be well. VIC and 64 owners could use the business software written for the

COMMODORE 64 GETS AWAY FROM BASICs with

THE Pascal Compiler FOR THE COMMODORE 64™

Limbic Systems, Inc. introduces the PASCAL COMPILER from Oxford Computer Systems (Software), Ltd., developers of *PETSPEED -- the BASIC compiler recommended by Commodore.

THE PASCAL COMPILER OFFERS --

- **EASE OF OPERATION**
  Pascal, a programming language, simplifies development of software for customized applications
- **POWER**
  to write programs efficiently and effectively
- **TEACHING TOOL**
  Pascal is the preferred language to teach programming skills
- **EFFICIENCY**
  debugging time is minimized -- the major task in program development
- **FLEXIBILITY**
  utilities are provided as an aid in programming development

CAPABILITY ECONOMICALLY
THE PASCAL COMPILER PRICED UNDER $50

Limbic Systems Inc.

560 San Antonio Road, Suite 202 Palo Alto, CA 94306
(415) 424-0168

Commodore 64 is a trademark of Commodore Electronics, Inc.
**PETSPEED is a trademark of Oxford Computer Systems (Software), Ltd.*

Let Your CBM-64 "SPEAK"

COMvoice IS AS EASY AS 1-2-3

1) *PLUG COMvoice INTO YOUR CBM-64*
2) *TURN YOUR COMPUTER ON*
3) *TYPE* SPEAK "HELLO, HOW ARE YOU"

AS EASY TO USE AS A PRINT STATEMENT

SPECIAL $99.95
W/EXTERNAL SPEAKER $139.95

DEALER INQUIRIES INVITED

ALSO ASK ABOUT OUR

HOME SECURITY AND ENERGY MANAGEMENT PRODUCTS

VIController
Wireless remote control system for the VIC-20 and CBM-64. Use with BSR and Leviton remote receiver modules. $69.95

COMsense
Input device for the VIC-20 and CBM-64. Provides 4 open/close and 2 analog inputs. $49.95

COMclock/AUTOboot
Clock/calendar cartridge for CBM-64 with battery backup and auto-start software in ROM. $69.95

GENESIS COMPUTER CORPORATION **P.O. Box 1143 Bethlehem, PA 18018 (215) 861-0850**

VIC-20 and CMB-64 are trademarks of Commodore Business Machines Inc.

2022/4022. But the 1525 had smaller ROMs, so luxurious features like formatted output were replaced with more limited, but useful features like dot-graphics mode.

The 1526 has many powerful commands. Although it lacks graphics mode, it does have a programmable character. That's right, one programmable character. You define it, then print CHR$(254) whenever you want it to appear. You could do a high-resolution screen dump with it, but many, many passes would be required to print a single line, and a full page could take an hour. In text mode, though, the 1526 is zippy, printing at 60 cps (characters per second).

A powerful feature for tabulated reports is the formatting channel. You define a string that describes how you want printed data to appear, then all output through another channel conforms to the "picture" you have defined. For example, if "AAA $999.99" were the field, and you sent "TOTAL",450, you'd get "TOT $450.00" on the printer. This is an indirect way to allow PRINT USING, a similar, built-in feature on most larger Microsoft BASICs.

The 1526 can also do automatic form feeds between pages, and has a diagnostic mode that displays error messages on the paper if you program the printer incorrectly. It honors cursor up and cursor down to switch between upper- and lowercase, as well as lowercase mode through a secondary address of 7. But it uses the code CHR$(14) for elongated characters, instead of the CHR$(15) the 1525 uses. The 1525 printer test seems to fail on the 1526 because of the change in codes.

What do we do about the 1526? It's an inexpensive, good quality, powerful printer. But it's incompatible with many programs designed for the 1525, and will not work properly with the VIC-20. For about the same price, you could buy a third-party printer and an interface (see elsewhere in this issue for information on these) that makes it compatible with the 1525. You can still use the 1526 with many programs, including SpeedScript (although some commands, underline for example, won't work). In the future, more programs will probably support either printer. But I wish Commodore would realize that compatibility makes the industry's job easier, and encourages a proliferation of quality programs for their computers. The inexpensive 1520 four color printer/plotter (see the review elsewhere in this issue) also uses a completely different printer standard.

The latest goof, if you will, is the Commodore 264. This, too, is a powerful, inexpensive computer, but it flies in the face of the VIC-20 and 64. Not only are the BASIC and graphics capabilities distinctly different from those found in the VIC-20 and 64, but even some of the BASIC tokens are different. You can still load VIC or 64 programs into the 264, since 2.0 tokens remain the same; but even though 264 BASIC has Super Expander and BASIC 4.0 commands, these extended commands are not token-compatible with either BASIC 4.0 or the Super Expander. Looks like a whole new market will have to spring up to support the 264, and that's only if Commodore is lucky.

Simons' BASIC: 100 New Commands

Yet another Commodore product that has generated a lot of mail is Simons' BASIC. What is it? What can you do with it? People are curious about a product which promises to simplify advanced programming on the 64. As you know, there are no commands in 64 BASIC for graphics, sound, or business applications. Simons' BASIC rectifies the situation with over 100 new commands. It's on cartridge and uses 8K of address space, giving you a FRE(0) of 30717. It was developed by David Simons, a 16-year-old from England. There is indeed a plethora of commands. Here are a few, almost self-explanatory commands:

AUTO RENUMBER FIND CENTRE DIR COLOUR HIRES MULTI LINE CIRCLE PAINT DRAW CHAR FLASH HRDCPY FILL MOB SET DESIGN IF..THEN..ELSE PROC EXEC ON ERROR VOL WAVE ENVELOPE MUSIC PLAY PENX POT JOY

The language is divided into several logical sections:

Programming Aids. These commands are like those found in BASIC Aid, plus many more. You can define the function keys, generate automatic line numbers, renumber your program, merge subroutines, search your program for specific text, trace the execution of your program, dump all variables and their values to the screen, even hide lines with DISAPA, which makes them disappear when listed (although they still RUN fine). A good programming aid package makes writing, editing, and debugging programs much easier.

Input Validation And Text Manipulation. These commands give you control over INPUT, allowing only certain keys. You can read and set the cursor position, center text, align numeric data, check function keys, even interrupt on the pressing of any key. Extended string operations let you insert one string into another, overlay one string within another, search for a substring within a string, and duplicate a character many times.

ULTRA COPY 64

DISK DUPLICATION SYSTEM FOR C-64

- *Analyze disk tracks for data & errors*
- *Skip empty tracks to speed copying*
- *Copy everything incl. DOS flag & false ID*
- *Put errors 20,21,22,23,27 & 29 on copy as required by latest protection schemes*
- *Fast, reliable copying with 1 or 2 drives*

$39.95 plus $3 shipping. Mastercard and Visa

98 % OF SOFTWARE CAN BE ULTRACOPY'ED

C-64 ULTRA RESET SWITCH

- *Built into new 6 foot disk drive cable*
- *Nothing to solder - no connections*
- *Eliminate voltage spikes & switch wear*
- *Recover programs after system crashes*

$16.95 plus $3 shipping. Mastercard and Visa

ULTRABYTE Call (313) 562 - 9855

23400 Michigan, Suite 502, Dearborn, MI 48124

Satisfaction guaranteed, 10 day return privilege

LEARN MACHINE LANGUAGE

- Write Fast-action Arcade-style graphics
- Fully use the Music synthesizer
- Completely understand the Computer
- Develop your skills inventory

Learn with the Tutorial that comes complete with a Full set of professional quality development tools.

DEVELOP-64 4.0 IS NOW FAST!!!

Assembles 2500 lines of code in under 20 seconds!

• Full Macro and conditional assembly capacity • 2600 lines of code in memory, Expandable to 17,000 on disk • Assemble direct to disk or memory • Co-resident full-screen editor (with search, replace, copy, move) and Debugger and Decoder • Decoder disassembles programs on disk or in memory • Built-in disk wedge • Program trace single step, execute • Set 10 breakpoints and/or go-points • Full-screen memory display and modify

PLUS the Machine Language Programmer's Bible: "Inside the Commodore 64"

$69.95

Plus $3.00 postage and handling. (Minn residents add 6%)

French Silk

P.O. Box 7096 Minneapolis, MN 55407

Call Toll-Free 1-800-328-0145

or in Minnesota call: (612) 871-4505

LOW COST SOFTWARE

Are you tired of paying high prices for your software? Let John Henry Software save you money!

We distribute public domain software for your **VIC 20™** or **Commodore 64™**. We've tested and documented each program to guarantee you hours of fun and useful learning experiences. We specialize in prompt delivery of your software, even if you order tapes, and we guarantee our product.

You'll also receive our free program reference book when you place your order.

VIC 20

Group VG	62 Games for Everyone	$7.95
Group VP	54 Programming, Demo, Business and Home	$7.95
Group VE	35 Educational Programs	$7.95

COMMODORE 64

Group CG	26 Games for Everyone	$7.95
Group CP	30 Programming, Demo, Business and Home	$7.95
Group CE	16 Educational Programs	$7.95
Group CA	5 Adventure Games (disk only)	$7.95

When ordering, specify group and tape or disk.
Send check or money order payable to:

John Henry Software
P.O. Box 39021
Cincinnati, Ohio 45239

Don't wait! Order your software today! Or write for your free program reference book. You'll be glad you did!

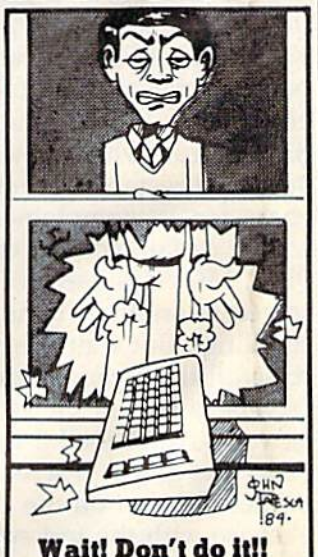
To keep our software prices low, our ad will only appear in the June, August, October and December issues of this magazine.

COMMODORE OWNERS WE'LL CHECK YOU OUT

Mr Tester™
Is your Commodore 64™ Disk Drive, Printer, Memory, Joystick, Monitor and Sound Chip operating correctly?

You may never know for sure. Mr. Tester is a complete diagnostic that tests:

1. **Full joystick operation in all axis .**
2. **Continuous or standard comprehensive memory test.**
3. **Commodore™ SID chip test for sound analysis.**
4. **Screen alignment and color test.**
5. **Complete read/write Disk Track and Block Test.**
6. **Diskette format analysis to check Floppys.**
7. **Complete printer test.**
8. **Complete keyboard test.**
9. **Cassette read/write test.**



Wait! Don't do it!!

All this for only

$29.95

order from

M-W Dist. Inc.
1342B Route 23
Butler, N.J. 07405
201-838-9027

Numeric Aids. MOD (short for modulo) allows you to "return the remainder when one integer is divided by another." For example, MOD(4,3) returns 1, MOD(3,2) returns 2, MOD(2,2) returns 0. You can use MOD to extract the low byte of a 16-bit number with MOD(n,256). DIV gives you the integer result of a division; FRAC gives you the fractional part of a number (FRAC(1.5) gives .5). You can do binary to decimal and hexadecimal to decimal conversions, and EXclusive OR two numbers.

There are two disk commands, which replace the need for a DOS wedge: DISK "command", such as DISK "S0:SNERD" to scratch the file called "SNERD". DIR will display the directory.

Graphics. This is one of the biggest sections in the language. You can draw, paint, and color in high resolution. PLOT turns on a single dot, LINE connects two points to draw a line, CIRCLE and ARC let you draw circles and parts of circles and ellipses. PAINT will fill in any shape with a color. With DRAW, you can design a shape with direction vectors, even ROTate it. You can easily put text on the graphics screen, with either character set. There are too many commands to cover here, but you'll find almost anything you need. The wealth of graphics commands exceeds that found on the Atari or even IBM Advanced BASIC.

Text Screen Manipulation. You can easily use Extended Background Color Mode with BCKGNDS. FLASH will alternate a character drawn in a certain color between normal and reverse field. BFLASH can be used to alternate the border between two colors. Both commands operate in the background: Your program will continue to run while they do their work. FCHR lets you fill a rectangular region of the screen with a character. Its complement, FCOL, fills an area with a certain color. Combined, they give you FILL. MOVE is used to copy one area of the screen to another (handy for multidirectional scrolling). But wait, there are built-in commands to scroll the screen left or right. You can also INVerse a part of the screen. SCRSV and SCRLD let you save or load a text screen. COPY reproduces the graphics screen on a 1525 compatible printer (but not on the 1526, alas). HRDCPY does likewise with the text screen.

Sprites. There are plenty of commands for sprite programming. DESIGN reserves space. You can draw a sprite or a character with the @ sign, embedding the shape right within your program. The word MOB is used in many of the commands. Apparently, Commodore called their sprites MOBs (for Movable OBjects) until the more popular term "sprite" (earlier used by Texas Instruments and a few others) replaced MOB. So CMOB sets the color of a multicolor sprite. MOB SET lets you initialize the sprite pointer, priority of sprite over background, and whether the sprite is normal or multicolor. MMOB moves a sprite to any screen position, can change its X or Y expansion, even move a sprite automatically at various speeds. RLOCMOB merely relocates a sprite to a new position. DETECT allows collision detection with CHECK. Finally, MOB OFF removes a sprite from the screen. You can also easily set up character graphics. MEM moves the character patterns from ROM to RAM. The memory configuration moves all around, though. The text screen is stored at $C000, but is bumped up to $CC00 after MEM. DESIGN is used again to replace a character, followed by @ and the character definition.

Structured Programming. If all that wasn't enough, Simons' BASIC may change the very style of your programming. Program control of execution is very important. Normally, you control execution with statements like GOSUB, IF.˙.THEN, and GOTO. Simons' BASIC extends IF.˙.THEN to allow ELSE, which will be executed when the IF fails. REPEAT.˙.UNTIL will cause a section of code to run until a certain condition is met. A strange command, RCOMP, lets you redo the condition of the last IF.˙.THEN, making the statement following it act as if the IF.˙.THEN were repeated. Program looping is simplified (or is it?) with LOOP.˙.EXIT IF.˙.END LOOP. LOOP and END LOOP bracket the code to be repeated.

The loop will continue until you leave it with EXIT IF, which is followed by a condition. A very powerful capability is PROC, which lets you define a subroutine that can be called by name with EXEC. You can also label a section of code with PROC and use CALL to jump to it (like GOTO). This makes program execution totally independent of line numbers. A simple system of LOCAL and GLOBAL variables are also supported. You can temporarily reuse the same variable names within a procedure without changing the original value, which is restored when you use GLOBAL.

Many people think structured programming inhibits their creativity. I don't flowchart or plan out my programming very much, and resist any suggestions to do so. Like many people, I prefer to just sit down at the keyboard and begin crafting. Structured programming as a discipline does encourage, even enforce this "plan before you do" approach, but adding the capability to structure your work enhances your options. I wish standard BASIC had IF.˙.THEN.˙.ELSE. Simons' BASIC may not keep me from using GOTO, but I definitely prefer branching to a

1541 DISK DRIVE ALIGNMENT PROGRAM

Finally, a complete disk drive alignment program! **No special equipment needed.** A two disk program allows anyone with average mechanical skills to properly align the 1541 disk drive. Complete instruction manual. **$39.95 + shipping**

PROGRAM PROTECTION FOR THE C-64

This is the book you've been waiting for! All the latest tips and secrets. A complete reference guide to software protection on the C-64. Covers the disk drive, bad tracks and sectors, modified directories, cartridges and much, much more. A complete and up to date guide to program protection of all types. Covers both basic and machine language protection schemes. A **complete memory map** and a disk with many helpful programs is included.

$29.95 + shipping

C. S. M. SOFTWARE

P. O. Box 563
Crown Point, IN 46307
(219) 663-4335

VISA AND MASTER CARDS ACCEPTED

Synapse Demo Disk $4

Preview the hottest games from Synapse for just $4*! The new self-running Commodore 64® demo disk contains up to eight of our most popular titles, and is available now by mail. Send your check or money order today!

Synapse
5221 Central Ave.
Richmond, CA 94804

Name _____ **Age** _____

Computer _____

Address _____

City _____ **State** _____ **Zip** _____

Phone _____

synapse

Please allow 3 to 4 weeks for delivery.
*$5 outside the continental U.S.

Program Your Own EPROMS

promenade™

▶ VIC 20
▶ C 64 **$99.50**

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

- 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
In California: 800-421-7748

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

VISA MasterCard

The Simpler, the Better

When it's on, it's on. No software to mess around with. This high quality, low-profile CP Numeric Keypad is the one for your Commodore 64 and VIC-20. It is guaranteed to be 100% compatible with all the software you have, now and forever, in any format. The Keypad easily connects in parallel with the existing keyboard connector. Now you can zip through your numeric work sheet, input your numbers and figures comfortably, quickly, and more easily than ever before at only $69.95.

Computer Place (213) 325-4754
23914 Crenshaw Blvd. Torrance, CA 90505

Dealer inquiries welcome

Commodore 64 and VIC-20 are trademarks of Commodore Business Machines, Inc.

named section of code rather than using a meaningless and arbitrary GOTO 5500.

Anyway, you can also trap and handle error conditions with ON ERROR GOTO. If there is any error, program execution will divert to a special error handling routine that you write. You can also use it to skip over lines that would cause an error, such as printing to a nonexistent printer. OUT disables error trapping. The error number is returned in the variable ERRN, and ERRLN gives the line number where the error occured.

Music. These commands simplify the use of the SID chip, but not too much. You still have to know how to shape the envelope and choose waveforms. Commands just replace POKEs, such as VOLume, and WAVE. The MUSIC command compiles a string of notes, and PLAY executes the music. I'm not sure what the difference is between MUSIC and PLAY, except that PLAY can play music in the background while your program continues to execute.

Finally, Simons' BASIC simplifies the use of joysticks, paddles, and a light pen. No more PEEKing; just use POT to read the paddle (potentiometer), JOY to read the joystick (which returns a number from 1 to 8, giving one of the directions), and PENX and PENY (horizontal/vertical position of the light pen).

Can you memorize 100 commands? The core of BASIC isn't even that big. I find that there are more commands than you really need. A language should be general purpose; some of the commands are too specific, such as LEFT and RIGHT, which scroll the screen left and right. CENTRE A$ can be easily replaced with the longer PRINT TAB(20-LEN(A$)/2);A$. It's amazing that so many commands can be built into the cartridge, but I think Simons' BASIC goes a little too far. Don't be intimidated by it, however. Again, you can use only the commands you need.

Another problem is that you can only run programs written with Simons' BASIC on another machine with Simons' BASIC. If you've been using it, write and tell me what you think. Incidentally, Simons' BASIC changes the screen color, even if you use RUN/STOP–RESTORE. So if you're already used to light blue on blue, get ready to switch again.

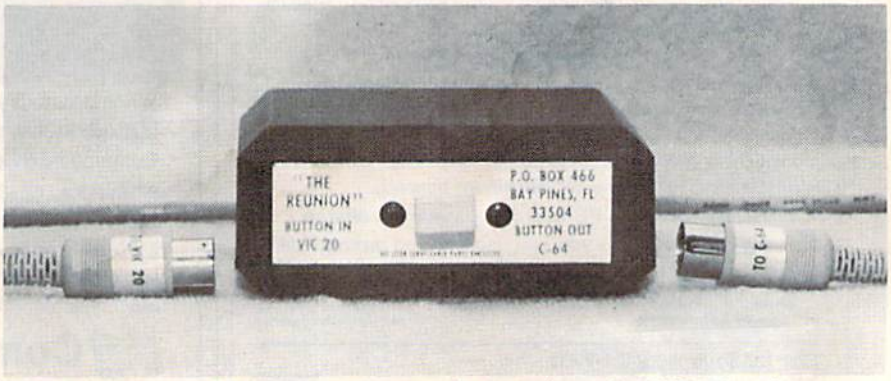
Simons' BASIC is available from Commodore or your local dealer. Suggested retail price is $59.95.

NEWS & PRODUCTS

Simultaneous Interface For VIC-20 And 64

The Reunion, recently introduced by HyTech, simultaneously interfaces a VIC-20 and Commodore 64 to a disk drive and/or printer, providing two computer systems.

In addition, through electronic coupling, *The Reunion* permits saving and loading of VIC-20 and Commodore programs



The Reunion, *by Hytech, simultaneously interfaces the VIC-20 and Commodore 64 with a disk drive and/or a printer, providing two computer systems.*

on the same disk. Cable switching is not necessary.

Any items attached to either computer, such as modems, expansion boards, etc., are unaffected by *The Reunion*. Should either computer be operated while *The Reunion* is set for the other one, no damage will occur. Instead, a "device not present" error will be displayed.

Two models are available at $29.95 each. Model A interfaces the VIC-20 and a Commodore 64 with a disk drive and a printer. Model B interfaces the VIC-20 and Commodore 64 computers with a Datassette recorder.

HyTech
P.O. Box 466
Bay Pines, FL 33504
(813)398-6661

Head Cleaner For Commodore

Nortronics has announced a software-driven head cleaner for Commodore computers.

The cleaner consists of a software program disk that first asks the user which drive he wants to clean. Once the cleaning disk is prepared and inserted into the drive, the program automatically steps the head out to the next unused portion of the disk, loads the head and spins the drive for 30 seconds.

The program keeps track of which one of the four cleaning bands it used last, and automatically advances the head to the next band when the next cleaning is done on the cleaning disk.

The program also tells the user when all four bands have been used and that a replaceable cleaning disk should be used.

The $39.95 retail price includes the program disk, two cleaning disks and a can of aerosol cleaning spray.

Nortronics Company, Inc.
8101 Tenth Ave. N.
Minneapolis, MN 55427
(612) 545-0401

Activision Software For Commodore

Activision has announced six new entertainment titles in disk and cartridge formats for the Commodore 64, ranging from fast action to space and adventure, to sport, humor and strategy.

Zenji is a strategy and puzzle game with an Eastern theme and melody. The object is to connect a glowing maze of elements to a pulsating source in order to create a single unified green image, or "Zenji."

Toy Bizarre chronicles the midnight adventures of a regular guy named Merton who wakes up in a toy factory gone berserk with sound and motion.

Four other previously released titles—*Pitfall, Beamrider, H.E.R.O.* and *The Activision Decathlon*—have been introduced for the Commodore 64, as well.

The games retail for $34.95 in the disk version and $39.95 for the cassette version.

Activision, Inc.
2350 Bayshore Frontage Road
Mountain View, CA 94043
(415) 960-0410

Free Educational Catalog

Opportunities for Learning, Inc., has announced publication of the new secondary school and college edition of *Selected Microcomputer Software*, a catalog of educational computer products.

The catalog features more than 400 software programs, books, and accessories for schools with Commodore 64, PET, Apple II, Atari, TRS-80, and IBM microcomputers.

Software is available in the areas of mathematics, science, reading and language arts, spelling and vocabulary skills, computer literacy and programming, logic and simulations, teacher and administrator utilities, SAT and test preparation, careers and guidance, social studies, foreign language, business education, games, music, and art.

The catalog is free. Also available free are the elementary school edition of *Selected Microcomputer Software* and a catalog of home educational software.

Opportunities for Learning, Inc.
8950 Lurline Avenue
Dept. L79
Chatsworth, CA 91311
(818) 341-2535

COMPUTE!'s GAZETTE *welcomes announcements of new products for VIC-20 and Commodore 64 computers, especially products aimed at beginning to intermediate users. Please send press releases and photos well in advance to: Tony Roberts, Assistant Managing Editor,* COMPUTE!'s GAZETTE, *P.O. Box 5406, Greensboro, NC 27403.*

New product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

A Beginner's Guide To Typing In Programs

What Is A Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in COMPUTE!'s GAZETTE for Commodore are written in a computer language called BASIC. BASIC is easy to learn and is built into all VIC-20s and Commodore 64s.

BASIC Programs

Each month, COMPUTE!'s GAZETTE for Commodore publishes programs for both the VIC and 64. To start out, type in only programs written for your machine, e.g., "VIC Version" if you have a VIC-20. Later, when you gain experience with your computer's BASIC, you can try typing in and converting certain programs from another computer to yours.

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one "right way" of stating something. Every letter, character, or number is significant. A common mistake is substituting a letter such as O for the numeral 0, a lowercase l for the numeral 1, or an uppercase B for the numeral 8. Also, you must enter all punctuation such as colons and commas just as they appear in the magazine. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Braces And Special Characters

The exception to this typing rule is when you see the braces, such as "{DOWN}". Anything within a set of braces is a special character or characters that cannot easily be listed on a printer. When you come across such a special statement, refer to "How To Type In *COMPUTE!'s GAZETTE* Programs."

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could "lock up," or "crash." The keyboard and STOP key may seem "dead," and the screen may go blank. Don't panic – no damage is done. To regain control, you have to turn off your computer, then turn it back on. This will erase whatever program was in memory, *so always SAVE a copy of your program before you RUN it*. If your computer crashes, you can LOAD the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is RUN. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get To Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program, so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter inverse video, lowercase, and control characters? It's all explained in your computer's manuals.

A Quick Review

1. Type in the program a line at a time, in order. Press RETURN at the end of each line. Use backspace or the back arrow to correct mistakes.

2. Check the line you've typed against the line in the magazine. You can check the entire program again if you get an error when you RUN the program.

3. Make sure you've entered statements in braces as the appropriate control key (see "How To Type *COMPUTE!'s GAZETTE* Programs" elsewhere in the magazine).

We regret that we are not able to respond to individual inquiries about programs, products, or services appearing in COMPUTE!'s GAZETTE *for Commodore due to increasing publication activity. On those infrequent occasions when a published program contains a typo, the correction will appear in the magazine, usually within eight weeks. If you have specific questions about items or programs which you've seen in* COMPUTE!'s GAZETTE *for Commodore, please send them to Gazette Feedback, P.O. Box 5406, Greensboro, NC 27403.*

How To Type In COMPUTE!'s GAZETTE Programs

Many of the programs which are listed in COMPUTE!'s GAZETTE contain special control characters (cursor control, color keys, inverse video, etc.). To make it easy to know exactly what to type when entering one of these programs into your computer, we have established the following listing conventions.

Generally, any VIC-20 or Commodore 64 program listings will contain words within braces which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S would mean to type the S key while holding the shift key. This would appear on your screen as a "heart" symbol. If you find an underlined key enclosed in braces (e.g., {10 N}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's).

If a key is enclosed in special brackets, ⟨⟩, you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as necessary.

Rarely, you'll see a solitary letter of the alphabet enclosed in braces. These characters can be entered on the Commodore 64 by holding down the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A. You should never have to enter such a character on the VIC-20, but if you do, you would have to leave the quote mode (press RETURN and cursor back up to the position where the control character should go), press CTRL-9 (RVS ON), the letter in braces, and then CTRL-0 (RVS OFF).

About the *quote mode*: You know that you can move the cursor around the screen with the CRSR keys. Sometimes a programmer will want to move the cursor under program control. That's why you see all the {LEFT}'s, {HOME}'s, and {BLU}'s in our programs. The only way the computer can tell the difference between direct and programmed cursor control is the quote mode.

Once you press the quote (the double quote, SHIFT-2), you are in the quote mode. If you type something and then try to change it by moving the cursor left, you'll only get a bunch of reverse-video lines. These are the symbols for cursor left. The only editing key that isn't programmable is the DEL key; you can still use DEL to back up and edit the line. Once you type another quote, you are out of quote mode.

You also go into quote mode when you INSerT spaces into a line. In any case, the easiest way to get out of quote mode is to just press RETURN. You'll then be out of quote mode and you can cursor up to the mistyped line and fix it.

Use the following table when entering cursor and color control keys:

When You Read:	Press:	See:
{CLR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + CRSR ↑	
{DOWN}	CRSR ↓	
{LEFT}	SHIFT + CRSR ←	
{RIGHT}	CRSR →	
{RVS}	CTRL + 9	
{OFF}	CTRL + 0	
{BLK}	CTRL + 1	
{WHT}	CTRL + 2	
{RED}	CTRL + 3	
{CYN}	CTRL + 4	
{PUR}	CTRL + 5	
{GRN}	CTRL + 6	
{BLU}	CTRL + 7	
{YEL}	CTRL + 8	
⟨1⟩	C= + 1	
⟨2⟩	C= + 2	
⟨3⟩	C= + 3	
⟨4⟩	C= + 4	
⟨5⟩	C= + 5	
⟨6⟩	C= + 6	
⟨7⟩	C= + 7	
⟨8⟩	C= + 8	
{F1}	f1	
{F2}	SHIFT + f1	
{F3}	f3	
{F4}	SHIFT + f3	
{F5}	f5	
{F6}	SHIFT + f5	
{F7}	f7	
{F8}	SHIFT + f7	

The Automatic Proofreader

"The Automatic Proofreader" will help you type in program listings from COMPUTE!'s Gazette without typing mistakes. It is a short error-checking program that hides itself in memory. When activated, it lets you know immediately after typing a line from a program listing if you have made a mistake. Please read these instructions carefully before typing any programs in COMPUTE!'s Gazette.

Preparing The Proofreader

1. Using the listing below, type in the Proofreader. The same program works on both the VIC-20 and Commodore 64. Be very careful when entering the DATA statements — don't type an l instead of a 1, an O instead of a 0, extra commas, etc.

2. SAVE the Proofreader on tape or disk at least twice *before running it for the first time.* This is very important because the Proofreader erases this part of itself when you first type RUN.

3. After the Proofreader is SAVEd, type RUN. It will check itself for typing errors in the DATA statements and warn you if there's a mistake. Correct any errors and SAVE the corrected version. Keep a copy in a safe place — you'll need it again and again, every time you enter a program from COMPUTE!'s Gazette.

4. When a correct version of the Proofreader is RUN, it activates itself. You are now ready to enter a program listing. If you press RUN/STOP-RESTORE, the Proofreader is disabled. To reactivate it, just type the command SYS 886 and press RETURN.

Using The Proofreader

All VIC and 64 listings in COMPUTE!'s Gazette now have a *checksum number* appended to the end of each line, for example *":rem 123". Don't enter this statement when typing in a program.* It is just for your information. The rem makes the number harmless if someone does type it in. It will, however, use up memory if you enter it, and it will confuse the Proofreader, even if you entered the rest of the line correctly.

When you type in a line from a program listing and press RETURN, the Proofreader displays a number at the top of your screen. *This checksum number must match the checksum number in the printed listing.* If it doesn't, it means you typed the line differently than the way it is listed. Immediately recheck your typing. Remember, don't type the rem statement with the checksum number; it is published only so you can check it against the number which appears on your screen.

The Proofreader is not picky with spaces. It will not notice extra spaces or missing ones. This is for your convenience, since spacing is generally not important. But occasionally proper spacing *is* important, so be extra careful with spaces, since the Proofreader will catch practically everything else that can go wrong.

There's another thing to watch out for: if you enter the line by using abbreviations for commands, the checksum will not match up. But there is a way to make the Proofreader check it. After entering the line, LIST it. This eliminates the abbreviations. Then move the cursor up to the line and press RETURN. It should now match the checksum. You can check whole groups of lines this way.

Special Tape SAVE Instructions

When you're done typing a listing, you must disable the Proofreader before SAVEing the program on tape. Disable the Proofreader by pressing RUN/STOP-RESTORE (hold down the RUN/STOP key and sharply hit the RESTORE key). This procedure is not necessary for disk SAVEs, *but you must disable the Proofreader this way before a tape SAVE.*

SAVE to tape erases the Proofreader from memory, so you'll have to LOAD and RUN it again if you want to type another listing. SAVE to disk does not erase the Proofreader.

Since the Proofreader is a machine language program stored in the cassette buffer, it will be erased during a tape SAVE or LOAD. If you intend to type in a program in more than one sitting or wish to make a safety SAVE, follow this procedure:

1. LOAD and RUN the Proofreader.
2. Disable it by pressing RUN/STOP-RESTORE.
3. Type the following three lines in direct mode (without line numbers):

```
A$="PROOFREADER.T":B$="{10 SPACES}":FO
RX=1TO4:A$=A$+B$:NEXTX
FORX=886 TO 1018:A$=A$+CHR$(PEEK(X)):N
EXTX
OPEN1,1,1,A$:CLOSE1
```

After you type the last line, you will be asked to press RECORD and PLAY. We recommend you start at the beginning of a new tape.

You now have a new version of the Proofreader (PROOFREADER.T, as renamed in the above code). Turn your computer off and on, then LOAD the program you were working on. Put the cassette containing PROOFREADER.T into the tape unit and type:

OPEN1:CLOSE1

You can now get into the Proofreader by typing SYS 886. To test this, PRINT PEEK (886) should return the number 173. If it does not, repeat the steps above, making sure that A$ (PROOFREADER.T) contains 13 characters and that B$ contains 10 spaces.

The new version of Automatic Proofreader will load itself into the cassette buffer whenever you type OPEN1:CLOSE1 and PROOFREADER.T is the next program on your tape. It will not disturb the contents of BASIC memory.

Automatic Proofreader For VIC And 64

```
100 PRINT"{CLR}PLEASE WAIT...":FORI=886TO
    1018:READA:CK=CK+A:POKEI,A:NEXT
110 IF CK<>17539 THEN PRINT"{DOWN}YOU MAD
    E AN ERROR":PRINT"IN DATA STATEMENTS.
    ":END
120 SYS886:PRINT"{CLR}{2 DOWN}PROOFREADER
     ACTIVATED.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```

MLX Machine Language Entry Program

For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost failsafe entry of machine language programs published in COMPUTE!'s GAZETTE. You need to know nothing about machine language to use MLX—it was designed for everyone. There are separate versions for the Commodore 64.

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file. You can then use the LOAD command to read the program into the computer, as with any program:

```
LOAD "filename",1,1   (for tape)
LOAD "filename",8,1   (for disk)
```

To start the program, you enter a SYS command that transfers control from BASIC to machine language. The starting SYS number always appears in the appropriate article.

Using MLX

Type in and save the correct version of MLX for your computer (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

You'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the comma, SPACE bar, or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

```
    U I O            7 8 9
  H J K L   become   0 4 5 6
    M , .            1 2 3
```

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session) you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later.

MLX recognizes these commands:

SHIFT-S: Save	**SHIFT-N: New Address**
SHIFT-L: Load	**SHIFT-D: Display**

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

What if you forgot where you stopped typing? Use the Display command to scan memory from the beginning to the end of the program. When you reach the end of your typing, the lines will contain a random pattern of numbers. When you see the end of your typing, press any key to stop the listing. Use the New Address command to continue typing from the proper location.

See program listing on page 148.

Power BASIC

(Article on page 108.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

Program 1:
String Search—BASIC Loader

```
100 PRINT"{CLR}{4 DOWN}{3 SPACES}STRING S
    EARCHER"                          :rem 93
110 PRINT"{2 DOWN}ONE MOMENT PLEASE"
                                      :rem 241
120 TP=PEEK(55)+256*PEEK(56)          :rem 39
130 TP=TP-186:H=INT((TP)/256):L=TP-H*256:
    POKE55,L:POKE56,H                 :rem 215
140 IN=PEEK(55)+256*PEEK(56):FORC=IN TO I
    N+185:READI:POKEC,I:CK=CK+I:NEXT:
                                      :rem 209
150 IFCK<>26449 THEN PRINT"ERROR IN DATA"
    :END                              :rem 130
160 REM****************** STRING SEARCH D
    ATA**********************         :rem 178
180 DATA 160,17,185,216,0,153,60,3,136,20
    8                                 :rem 11
190 DATA 247,160,9,177,45,133,217,200,177
    ,45                               :rem 128
200 DATA 133,218,200,177,45,133,219,24,16
    0,2                               :rem 102
210 DATA 177,47,101,47,105,7,133,220,200,
    177                               :rem 105
220 DATA 47,101,48,133,221,160,0,24,165,4
    7                                  :rem 3
230 DATA 105,7,133,224,165,48,105,0,133,2
    25                                :rem 53
240 DATA 169,0,240,12,160,17,185,60,3,153
                                      :rem 212
250 DATA 216,0,136,208,247,96,24,165,224,
    105                               :rem 118
260 DATA 3,133,224,165,225,105,0,133,225,
    160                               :rem 98
270 DATA 0,177,224,153,229,0,200,192,3,20
    8                                  :rem 7
280 DATA 246,24,165,220,105,2,133,220,165
    ,221                              :rem 150
290 DATA 105,0,133,221,165,229,240,202,20
    8,2                               :rem 98
300 DATA 240,210,162,0,134,227,134,226,24
    ,165                              :rem 146
310 DATA 217,197,226,240,31,24,165,229,19
    7,227                             :rem 228
320 DATA 144,37,164,226,177,218,164,227,2
    09,230                            :rem 19
330 DATA 208,6,230,227,230,226,208,226,23
    0,227                             :rem 207
340 DATA 169,0,133,226,240,218,160,0,169,
    1                                  :rem 7
350 DATA 145,220,200,169,0,145,220,240,19
    7,160                             :rem 202
360 DATA 0,152,145,220,240,242        :rem 179
999 PRINT"DONE":NEW                   :rem 198
```

Program 2:
String Search—Demo Program

```
10 REM STRING SEARCH DEMO{9 SPACES}PROGRA
   M                                  :rem 15
20 A$="DUMMY DATA":REM**MUST BE A STRING*
   *                                 :rem 156
30 Q$="":REM THIS IS TO BE USED AS THE SE
   ARCH STRING *************          :rem 202
40 DIMA$(300),Q%(300):REM SEARCHED STRING
    AND FLAG ARRAY                   :rem 173
45 ML=PEEK(55)+256*PEEK(56):REM START ADD
   RESS                               :rem 164
50 PRINT"{CLR}{2 DOWN}{2 SPACES}STRING SE
   ARCH DEMO"                         :rem 157
100 PRINT"BUILDING ARRAY"              :rem 47
110 Q$="GOOD"                         :rem 177
120 FORL=1TO299                       :rem 123
130 :                                 :rem 206
140 :A$(L)="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
                                      :rem 49
150 :                                 :rem 208
160 NEXTL                              :rem 34
170 A$(1)="GARBAGE GOOD MORE GARBAGE"
                                      :rem 46
180 A$(10)="GARB GOOD MORE GARB"      :rem 197
185 A$(70)="GOOD GARBAGE"              :rem 78
190 A$(100)="GARBAGE GOOD"            :rem 116
195 A$(250)="GARBAGE GOOD MORE GARBAGE"
                                      :rem 155
200 PRINT"ARRAY FINISHED"              :rem 44
300 REM***************{8 SPACES}BASIC SEA
    RCH{2 SPACES}**********************
                                      :rem 161
310 PRINT"BASIC SEARCH":TI$="000000"
                                      :rem 25
320 FORL=1TO299                       :rem 125
330 :FORJ=1TOLEN(A$(L))-LEN(Q$)+1:rem 114
340 ::IFMID$(A$(L),J,LEN(Q$))=Q$THENQ%(L)
    =1:NEXTL                           :rem 89
350 :NEXTJ                             :rem 91
360 NEXTL                              :rem 36
370 PRINTTI;"JIFFIES"                  :rem 67
380 FORL=1TO299                       :rem 131
390 :IFQ%(L)<>0THENPRINTA$(L)         :rem 224
395 NEXTL                              :rem 44
400 REM***************{10 SPACES}ML SEARC
    H{3 SPACES}**********************
                                      :rem 217
410 PRINT"ML SEARCH":TI$="000000"     :rem 81
420 SYS(ML)                           :rem 127
430 PRINTTI;"JIFFIES"                  :rem 64
440 FORL=1TO299                       :rem 128
450 :IFQ%(L)<>0THENPRINTA$(L)         :rem 221
460 NEXTL                              :rem 37
999 END                               :rem 130
```

Disk Purge

(Article on page 110.)

```
30 REM FOR 4040/2031/1540/1541    :rem 222
40 PRINT "{CLR}{RVS}DISKETTE PURGE"
                                      :rem 186
50 PRINT:PRINT"WHICH DRIVE (0/1) ?{RVS}
   {OFF}{LEFT}";                      :rem 134
60 GET DR$:IF DR$<>"0" AND DR$<>"1" THEN
   {SPACE}60                          :rem 130
70 PRINT DR$:D=VAL(DR$)                :rem 87
80 DR$= "I"+DR$:OPEN 15,8,15,DR$      :rem 127
```

```
90 GOSUB 450                          :rem 130
100 OPEN 1,8,3,"#":GOSUB 460          :rem 158
110 PRINT:PRINT "PRESS'{RVS}Y{OFF}'TO DEL
    FILE":PRINT                       :rem 97
120 Z$=CHR$(0)                        :rem 208
130 T=18:S=1                          :rem 137
140 PRINT#15,"U1:"3;D;T;S             :rem 170
150 PRINT#15,"B-P:"3;2+32*R           :rem 193
160 GET#1,A$:IF A$="" THEN A$=Z$      :rem 90
170 IF ASC(A$)<129 THEN 270           :rem 84
180 FOR K=5+32*R TO 20+32*R           :rem 90
190 PRINT#15,"B-P:"3;K                :rem 210
200 GET#1,A$:IF A$="" THEN PRINT"{RVS}
    {OFF}";:GOTO 220                  :rem 236
210 PRINT A$;                         :rem 192
220 NEXT                              :rem 211
230 PRINT ,,"PURGE ? {RVS} {OFF}{LEFT}";
                                      :rem 252
240 GETA$:IFA$<>"Y" AND A$<>"N" THEN240
                                      :rem 41
250 PRINT A$                          :rem 137
260 IF A$="Y" THEN GOSUB 360          :rem 174
270 R=R+1:IF R<8 THEN 140             :rem 41
280 PRINT#15,"B-P:"3;0                :rem 183
290 GET#1,A$:IFA$="" THENA$=Z$        :rem 94
300 T=ASC(A$):IF T=0 THEN PRINT:PRINT "EN
    D OF DIRECTORY.":GOTO 400         :rem 88
310 PRINT#15,"B-P:"3;1                :rem 178
320 GET#1,A$:IFA$=""THENA$=Z$         :rem 88
330 S=ASC(A$):PRINT:PRINT "*** NEXT TRACK
    : ";:NU=T:GOSUB520:PRINT N$;      :rem 95
340 PRINT TAB(20);"NEXT SECTOR: ";:NU=S:G
    OSUB 520:PRINT N$;" ***":PRINT
                                      :rem 108
350 R=0:GOTO140                       :rem 95
360 PRINT#15,"B-P:"3;5+32*R-3         :rem 39
370 PRINT#1,CHR$(0);                  :rem 100
380 PRINT#15,"U2:"3;D;T;S             :rem 177
390 GOSUB460:RETURN                   :rem 208
400 CLOSE 1                           :rem 59
405 PRINT "VALIDATE DISK Y OR N"      :rem 39
406 GET V$:IF V$="Y"THEN410           :rem 213
407 IF V$=""THEN 406                  :rem 238
408 CLOSE15:END                       :rem 137
410 PRINT:PRINT "{RVS}VALIDATING DISK
    {OFF}"                            :rem 223
420 PRINT#15,("V"+DR$)                :rem 168
430 GOSUB460:CLOSE 15                 :rem 199
440 END                               :rem 111
450 REM CHECK DISK STATUS             :rem 234
460 INPUT#15,ER,ER$,TR,SE             :rem 243
470 IF ER=0 THEN RETURN               :rem 61
480 CLOSE 15:CLOSE 1                  :rem 89
490 PRINT:PRINT "{DOWN}DISK ERROR! ";
                                      :rem 87
500 PRINT ER;ER$;TR;SE                :rem 99
510 STOP                              :rem 220
520 REM FORMAT RETURN WITHOUT GOSUBS
                                      :rem 43
530 N$=MID$(STR$(NU),2)               :rem 5
540 IF LEN(N$)<2 THEN N$="0"+N$       :rem 39
550 RETURN                            :rem 122
```

Sno-Cat

(Article on page 62.)

Program 1: Sno-Cat—64 Version

```
2 POKE 53281,1:POKE 53280,0           :rem 136
4 POKE 52,60:POKE 56,60               :rem 194
5 PRINT "{CLR}"                       :rem 153
10 DATA 120,169,73,141,20,3,169,3,141
                                      :rem 12
20 DATA 21,3,88,96,169,208,133,252,169
                                      :rem 91
30 DATA 3,133,251,162,1,160,0,177,251
                                      :rem 7
40 DATA 105,01,144,2,169,251,145,251,224
                                      :rem 162
50 DATA 7,240,8,232,230,251,230,251,24
                                      :rem 59
60 DATA 144,234,173,1,220,41,15,201
                                      :rem 164
62 DATA 11,240,7,201,7,240,16,76,49,234,1
   73,0,208,201,32,240,246,206,0,208
                                      :rem 32
64 DATA 24,144,240,173,0,208,201,255,240,
   233,238,0,208,24,144,227,0,0,0,0,0,0
                                      :rem 60
70 FORU=828TO921:READQ:POKEU,Q:NEXT
                                      :rem 28
80 SYS828:POKE 53248+21,0:VIC=53248
                                      :rem 193
85 GOSUB 400                          :rem 129
86 REM                                :rem 82
87 GOSUB 1000                         :rem 176
89 POKE 856,2                         :rem 161
90 X5=RND(-SQR(RND(1)*TI))            :rem 9
100 S=53248:PRINT "{CLR}{BLK}"        :rem 85
105 POKE 53281,3                      :rem 39
110 FORU=1024 TO 1063:POKEU,160:POKEU+960
    ,160                              :rem 11
115 POKE54272+U,6:POKE54272+U+960,6:NEXT
                                      :rem 62
120 FORU=1024 TO 1984 STEP 40:POKEU,160:P
    OKE U+33,160:POKE U+39,160        :rem 151
125 POKE U+54272,14:POKE 54272+U+33,14:PO
    KE 54272+U+39,14:NEXT             :rem 121
126 POKE 53281,1                      :rem 40
130 FORU=0TO14STEP2:POKEU+VIC,(U*18):NEXT
                                      :rem 92
131 POKE 2040,203                     :rem 75
132 POKE VIC+28,34:POKE VIC+37,5:POKE VIC
    +38,5                             :rem 82
133 FORU=40 TO 47:POKEVIC+U,5:NEXT:rem 39
135 POKE VIC+21,255                   :rem 0
140 FORI=1TO7:POKE 2040+I,202:NEXT
                                      :rem 233
141 DI=254                            :rem 251
145 X=150:DI=0                        :rem 249
146 POKE VIC,X:TI$="000000":A$(2)="SECOND
    ":A$(3)="THIRD ":A$(4)="FOURTH "
                                      :rem 199
147 A1=PEEK(VIC+30):FORI=0TO24:POKE54272+
    I,0:NEXT                          :rem 199
148 POKE 54296,15:POKE 54273,0:POKE54277,
    85:POKE54278,85:POKE 54276,129
                                      :rem 130
149 A$(1)="BRAKE ":G=2:A$(5)="OVER DRIVE"
                                      :rem 183
150 I=0:A=INT(RND(1)*7)+1:IF PEEK(VIC+(A*
    2)+1)>240 THEN I=1                :rem 179
151 A1=PEEK(VIC+30)AND1:IFA1THEN 3000
                                      :rem 93
154 IFBR=0THENPOKE 856,G              :rem 158
155 IFBR=0THENIF(PEEK(56321)AND16)=0THENP
    OKE856,1:BR=1:GOSUB5000
                                      :rem 113
156 IFBRTHEN DI=DI-1                  :rem 163
157 IFBRTHENIF(PEEK(56321)AND16)=16THENPO
    KE856,G:BR=0:POKE54276,129:GOSUB5100
                                      :rem 140
```

```
161 DI=DI+1:PRINT"{RVS}{HOME} DISTANCE TO
    GO:{4 SPACES}{4 LEFT}"400-DI;:IF DI
    {SPACE}=>400 THEN 1500          :rem 144
162 IF TI$="000010"THENG=3          :rem 113
163 IF TI$="000050"THENG=4          :rem 119
165 PRINT TAB(20);"{RVS} GEAR ";A$(PEEK(8
    56));                           :rem 168
166 PRINT "{HOME}{2 DOWN}"TAB(33);"{RVS}"
    TI$"                            :rem 165
167 PRINT "{DOWN}"TAB(34)"TIME"     :rem 130
168 POKE 54272,G*30+20               :rem 95
169 IF TI$="000008" THEN PRINT"{HOME}
    {4 DOWN}"TAB(9);"SHIFTING GEARS"
                                    :rem 158
170 IF TI$="000048" THEN PRINT"{HOME}
    {4 DOWN}"TAB(9);"SHIFTING GEARS"
                                    :rem 154
171 IF TI$="000011" THEN PRINT"{HOME}
    {4 DOWN}"TAB(9);"{14 SPACES}":rem 195
172 IF TI$="000051" THEN PRINT"{HOME}
    {4 DOWN}"TAB(9);"{14 SPACES}":rem 200
174 IFI=0THENGOTO 150               :rem 223
175 POKE 2040+A,202                 :rem 190
177 POKE VIC+(A*2)+1,0              :rem 129
178 A1=PEEK(VIC+30)AND1:IFA1THEN 3000
                                    :rem 102
179 POKE 54272,PEEK(856)*12         :rem 166
180 GOTO 150                        :rem 104
400 REM{5 SPACES}***{4 SPACES}OPENING SCR
    EEN{5 SPACES}***                 :rem 68
405 PRINT "{CLR}"                   :rem 253
470 DATA0,0,120,198,60,204,198,102
                                    :rem 175
480 DATA192,246,102,120,246,102,12,222,10
    2                                :rem 47
490 DATA204,222,102,120,198,60,0,0,0
                                      :rem 1
500 I=200:GOSUB 520                 :rem 254
510 GOTO 560                        :rem 106
520 FORR=I*64TO(I*64)+62:POKER,0:NEXT
                                    :rem 222
530 FORU=I*64+19 TO I*64+45:READQ:POKEU,Q
    :NEXT                           :rem 242
540 RETURN                          :rem 121
560 I=201:GOSUB 520                   :rem 5
570 DATA 0,0,0,120,49,248,204,120,96
                                    :rem 220
580 DATA 192,204,96,192,252,96,192,204,96
                                    :rem 253
590 DATA 204,204,96,120,204,96,0,0,0,0
                                    :rem 55
605 POKE 2040,200:POKE 2041,200:POKE 2042
    ,201:VIC=53248                  :rem 139
610 POKE VIC+21,6:POKE VIC+29,7:POKEVIC+2
    3,7                              :rem 26
620 POKE VIC+2,136:POKEVIC+4,188    :rem 67
625 POKE VIC+41,14:POKE VIC+40,14   :rem 58
630 POKE VIC+3,0:POKEVIC+5,0        :rem 107
632 S=54272:POKES+24,15:POKES+5,85:POKES+
    6,85:POKES+12,85:POKES+13,85    :rem 64
634 POKE S+4,33:POKES+11,17         :rem 237
640 IF PEEK(VIC+3)=>127 THEN POKE VIC+1,P
    EEK(VIC+3):GOTO 660             :rem 129
645 POKE S+1,128-PEEK(VIC+3):POKES+8,128-
    PEEK(VIC+5)                     :rem 242
650 GOTO 640                        :rem 110
660 POKE VIC,PEEK(VIC+2):POKE VIC+21,5:PO
    KE VIC+1,127:POKEVIC+39,6       :rem 121
670 IF PEEK(VIC+5)>=127 THEN 690    :rem 199
675 POKE S+8,128-PEEK(VIC+5)         :rem 51
680 GOTO 670                        :rem 116
690 POKE 856,0:POKE VIC+41,6:POKEVIC+5,12
    7                               :rem 123
693 POKE S+24,0                      :rem 17
700 FORU=0TO5:FORE=0TO14:POKEVIC+39,E:POK
    EVIC+41,E:NEXTE,U:GOSUB720      :rem 36
701 POKE 53254,171:POKE53255,200:POKE5325
    7,100:POKE53256,171:POKE2044,204
                                    :rem 14
702 POKE2043,203:POKE VIC+21,13+16:rem 77
703 POKE VIC+42,12                  :rem 203
704 POKE 198,0:FORDL=1TO400:NEXT    :rem 151
705 SB=1024:CB=SB+54272:FORI=10TO29:POKES
    B+I+160,160:POKECB+I+160,6      :rem 136
706 POKE SB+I+880,160:POKECB+I+880,6:NEXT
    :FORI=4TO22:W=I*40:POKESB+W+10,160
                                    :rem 193
707 POKE CB+W+10,6:POKESB+W+29,160:POKECB
    +W+29,6:NEXT                    :rem 124
708 PRINT"{15 DOWN}{15 RIGHT}PRESS {RVS}F
    IRE"                            :rem 231
710 GET A$:IFA$=""THEN710           :rem 83
711 POKE VIC+21,0:FORU=0TO14:POKEU+VIC,0:
    NEXT                            :rem 31
715 POKEVIC+23,0:POKE VIC+29,0:PRINT"
    {CLR}":RETURN                   :rem 143
720 FORI =202*64 TO 204*64+62       :rem 142
730 READ Q:POKEI,Q:NEXT:RETURN      :rem 201
740 DATA 0,16,0,0,40,0,0,84,0       :rem 108
750 DATA 0,170,0,1,85,0,2,170,128 :rem 65
760 DATA 5,85,64,2,170,128,5,85,64
                                    :rem 148
770 DATA 10,170,160,21,85,80,10,170,160
                                    :rem 112
780 DATA 21,85,80,42,170,168,0,56,0
                                    :rem 185
790 DATA 0,56,0,0,56,0,5,85,64      :rem 188
800 DATA 42,128,168,0,0,0,0,0,0     :rem 210
810 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0:rem 51
820 DATA 126,0,126,24,255,24,126,126,126
                                    :rem 169
830 DATA 31,255,248,126,126,126,24,255,24
                                    :rem 230
840 DATA 126,255,126,16,255,8,112,255,14
                                    :rem 180
850 DATA 23,0,232,119,255,238,22,255,104
                                    :rem 172
860 DATA 118,255,110,22,255,104,119,255,2
    38                              :rem 73
870 DATA 16,0,8,126,0,126,0,0,0,0   :rem 54
880 RETURN                          :rem 128
1000 REM{4 SPACES}***{4 SPACES}INSTRUCTIO
    NS{7 SPACES}***                  :rem 86
1001 PRINT "{CLR}":A1=PEEK(56321)AND16:IF
    A1=0THEN1000                     :rem 88
1002 POKE 856,0                     :rem 241
1003 POKE VIC+23,0:POKE V+29,0      :rem 114
1005 POKE 53281,6:FORI=12TO27:POKE1064+I,
    160:POKE1064+I+160,160:NEXT :rem 129
1008 FORI=1064 TO 1064+160STEP40:POKEI+12
    ,160:POKEI+27,160:NEXT          :rem 86
1010 PRINT "{2 DOWN}"TAB(14);"{RVS}{7}INS
    TRUCTIONS"                      :rem 34
1020 PRINT TAB(14);"{12 T}"          :rem 0
1040 PRINT "{3 DOWN}{7}{3 SPACES}YOU ARE
    {SPACE}ONE OF THE RANGERS"  :rem 162
1050 PRINT "{2 SPACES}WITH THE JOB OF RES
    CUING"                          :rem 132
1060 PRINT "{2 SPACES}SKIERS AT THE MOUNT
     CRUMB SKI LODGE."              :rem 107
1090 PRINT "{2 SPACES}YOU MUST MAKE THE T
    REACHEROUS"                      :rem 37
1100 PRINT "{2 SPACES}CLIMB TO SAVE THEM
    {DOWN}"                         :rem 11
```

```
1110 PRINT "{2 SPACES}{2 DOWN} {RVS}{WHT}
     THIS IS YOU:"                    :rem 216
1120 PRINT TAB(15);"{P}M"             :rem 218
1130 PRINT TAB(15);"{Y}N":VIC=53248
                                      :rem 67
1132 POKE VIC+5,70:POKE VIC+4,40:POKE 204
     2,202:POKEVIC+41,0               :rem 45
1140 POKE VIC+21,5:POKE2040,203       :rem 182
1150 POKE VIC,180:POKE VIC+1,200      :rem 2
1160 PRINT "{8}{DOWN}{2 SPACES}PRESS
     {WHT}{RVS}FIRE{OFF} TO START"
                                      :rem 210
1163 SB=1024:CB=54272+SB:FORI=0TO39:POKES
     B+I,160:POKECB+I,7               :rem 4
1164 POKESB+I+960,160:POKECB+I+960,7:NEXT
                                      :rem 153
1165 FORI=0TO960 STEP 40              :rem 76
1166 POKESB+I,160:POKECB+I,7:POKESB+I+39,
     160:POKECB+I+39,7:NEXT           :rem 47
1170 A=PEEK(56321)AND16:IFATHEN1170
                                      :rem 250
1180 PRINT"{CLR}":POKEVIC+21,0:RETURN
                                      :rem 125
1500 REM{4 SPACES}***{4 SPACES}YOU MADE I
     T{8 SPACES}***                   :rem 87
1501 POKE VIC+11,0:POKE2045,204       :rem 183
1502 T$=TI$                           :rem 62
1505 POKE VIC+21,252                  :rem 47
1510 IF PEEK(VIC+5)=<50THEN1510       :rem 226
1520 SP=53251:FORU=1TO7               :rem 102
1530 IF U=5THEN1550                   :rem 25
1540 POKESP,0                         :rem 248
1550 SP=SP+2:NEXT                     :rem 36
1560 IF PEEK(VIC+11)=<150 THEN 1560
                                      :rem 74
1570 POKE 856,2                       :rem 253
1580 IFPEEK(VIC+11)=<179 THEN 1580:rem 89
1590 POKE 856,0                       :rem 253
1600 PRINT "{HOME}{4 DOWN}"TAB(11);"YOU M
     ADE IT"                          :rem 169
1610 H1(1)=25:L1(1)=30:H2(1)=18:L2(1)=209
                                      :rem 46
1620 H1(2)=33:L1(2)=135:H2(2)=25:L2(2)=30
                                      :rem 46
1630 H1(3)=42:L1(3)=62 :H2(3)=31:L2(3)=16
     5                                :rem 56
1640 H1(4)=50:L1(4)=60 :H2(4)=37:L2(4)=16
     2                                :rem 61
1650 H1(5)=42:L1(5)=62 :H2(5)=31:L2(5)=16
     5                                :rem 66
1660 H1(6)=50:L1(6)=60 :H2(6)=37:L2(6)=16
     2                                :rem 71
1670 S=54272:POKES+5,85:POKES+6,85:POKES+
     12,85:POKES+13,85                :rem 148
1680 POKES+24,15:POKES+4,33:POKES+11,17
                                      :rem 254
1685 FOR X=1TO6:POKES,L1(X):POKES+1,H1(X)
     :POKES+7,L2(X):POKES+8,H2(X):rem 173
1690 IF H1(X)=50THENFORT=1TO200:NEXT
                                      :rem 169
1695 FORT=1TO200:NEXT:NEXTX           :rem 253
1699 FORD=0TO24:POKES+D,0:NEXT        :rem 122
1700 GOTO 3100                        :rem 197
3000 REM{4 SPACES}***{4 SPACES}YOU CRASHE
     D ? ? ?{5 SPACES}***             :rem 87
3002 POKE856,0:S=54272:POKES,240:POKES+1,
     33:POKES+5,8:POKES+22,104:POKES+23,1
                                      :rem 30
3003 POKE S+24,79                     :rem 117
3005 POKES+4,129:FORDL=1TO100:NEXT:POKES+
     4,128                            :rem 31
3006 POKE 856,1                       :rem 248
3010 FORI=200 TO 251:POKE53249,I:NEXT
                                      :rem 92
3060 POKE 856,1:T$=TI$:POKES+4,0      :rem 31
3070 FORY=32TO50:POKE2040,Y:NEXT      :rem 225
3080 POKE VIC+21,254                  :rem 49
3081 S=54272:POKES+24,15:POKES+5,85:POKE
     {SPACE}S+1,1:POKE S+4,17         :rem 24
3083 FORDL=1TO200:NEXT:POKES+4,16:FORDL=1
     TO200:NEXT:POKE S+4,0            :rem 83
3090 PRINT "{HOME}{4 DOWN}"TAB(8);"
     {4 SPACES}YOU CRASHED{2 SPACES}"
                                      :rem 202
3095 POKE 54276,0                     :rem 100
3100 PRINT "{2 DOWN}"TAB(13)"GAME OVER"
                                      :rem 221
3105 POKES+4,0:POKES+4,33:POKE S+5,85:POK
     ES+24,15                         :rem 71
3110 A1=PEEK(56321)AND16:POKE53280,RND(1)
     *16                              :rem 179
3111 SC=ABS(INT(((300-VAL(T$))*DI)/2)):IF
     DI=400THENSC=SC+500              :rem 161
3112 A=INT(RND(1)*7)+1:IF PEEK(VIC+(A*2)+
     1)=251 THEN POKEVIC+(A*2)+1,0:rem 32
3114 IF A1 THEN 3110                  :rem 190
3115 POKE 56321,0:POKE53280,0         :rem 29
3116 POKE S+24,0                      :rem 58
3117 POKE 53281,14                    :rem 143
3120 POKE VIC+21,0                    :rem 193
3125 A1=PEEK(56321)AND16:IF A1=0THEN3125
                                      :rem 205
3130 PRINT "{CLR}{WHT}"               :rem 48
3133 PRINT "{6 DOWN}{13 SPACES}YOUR SCORE
      "SC                             :rem 98
3136 IFSC=>HS THEN HS=SC              :rem 165
3139 PRINT "{BLU}{2 DOWN}{13 SPACES}BEST
     {SPACE}SCORE "HS                 :rem 39
3140 PRINT "{2 DOWN}{4 RIGHT}{BLU}
     {7 SPACES}WOULD YOU LIKE TO":rem 158
3150 PRINT "{DOWN}{4 RIGHT}{11 SPACES}PLA
     Y AGAIN?"                        :rem 244
3160 PRINT "{DOWN}{WHT}{2 RIGHT}
     {14 SPACES}{RIGHT}Y / N"         :rem 222
3165 PRINT{2 SPACES}"{HOME}{18 DOWN}
     {14 RIGHT}{CYN}{2 SPACES}{RVS} {YEL}
     {OFF}↑{RVS}{CYN}{3 SPACES}{YEL}
     {CYN} {OFF}"                     :rem 58
3170 A4=1761:A5=1765                  :rem 12
3175 A=A4                             :rem 195
3180 A1=PEEK(56321)AND15              :rem 43
3190 IF A1=11THENA=A4                 :rem 143
3200 IF A1=7 THEN A=A5                :rem 93
3210 POKE A4,32:POKEA5,32:POKEA,30
                                      :rem 164
3220 A1=PEEK(56321)AND16:IFA1THEN3180
                                      :rem 93
3230 IF A=A4 THEN 86                  :rem 231
3235 POKE 198,0:FORI=1TO30:PRINT"{DOWN}";
     :NEXT                            :rem 168
3240 END                              :rem 160
4000 DATA{2 SPACES}31 , 255 , 224 , 31 ,
     {SPACE}199 , 224 , 31 , 199 , 224
                                      :rem 20
4010 DATA{2 SPACES}31 , 1 , 224 , 31 , 1
     {SPACE}, 224 , 31 , 199 , 224:rem 56
4020 DATA{2 SPACES}31 , 199 , 224 , 31 ,
     {SPACE}255 , 224 , 24 , 0 , 0:rem 61
4030 DATA{2 SPACES}24 , 112 , 0 , 24 , 11
     2 , 0 , 31 , 252 , 0         :rem 194
4040 DATA{2 SPACES}24 , 126 , 0 , 24 , 11
     5 , 0 , 24 , 246 , 0         :rem 208
4050 DATA{2 SPACES}25 , 176 , 0 , 25 , 17
     6 , 0 , 127 , 176 , 0        :rem 21
```

```
4060 DATA{2 SPACES}225 , 255 , 0 , 0 , 0
     {SPACE}, 0 , 0 , 0 , 0          :rem 201
5000 POKE 54276,0:POKE54277,144:POKE 5427
     6,33:POKE 54276,32:POKE54278,144
                                     :rem 204
5005 POKE 54273,20                   :rem 140
5010 RETURN                          :rem 166
5100 POKE 54276,0:POKE 54276,129     :rem 146
5105 POKE 54273,0:POKE 54277,85      :rem 102
5110 RETURN                          :rem 167
```

Program 2:

Sno-Cat—VIC Loader Program

```
5 POKE52,26:POKE56,26:CLR            :rem 226
10 PRINT"{CLR}{5 DOWN}{5 SPACES}PLEASE WA
   IT"                               :rem 9
15 PRINT"{2 DOWN}{2 SPACES}Z MOVES YOU LE
   FT                                :rem 67
17 PRINT"{2 DOWN}{2 SPACES}X MOVES YOU RI
   GHT"                              :rem 184
18 PRINT"{2 DOWN}GET TO THE TOP OF THE
   {3 SPACES}{DOWN}HILL AND SAVE THE
   {7 SPACES}{DOWN}INJURED SKIER{WHT}
   {9 SPACES}";                      :rem 38
20 FORA=6656TO6786:READB:POKEA,B:NEXT
                                     :rem 69
30 FORA=6912TO7100:READB:POKEA,B:NEXT
                                     :rem 46
100 S$="{WHT}LO"+CHR$(34)+"SC"+CHR$(34)+"
    ,8:"+CHR$(131)                   :rem 149
110 FOR I=1TOLEN(S$):POKE630+I,ASC(MID$(S
    $,I)):NEXT:POKE198,I:END         :rem 137
200 DATA120,169,81,141,20,3,169,26
                                     :rem 175
210 DATA141,21,3,88,96,172,251,3     :rem 81
220 DATA185,228,31,201,33,144,5,169
                                     :rem 231
230 DATA1,141,250,3,169,32,153,228
                                     :rem 174
240 DATA31,165,197,201,33,208,13,206
                                     :rem 20
250 DATA251,3,173,251,3,208,3,238:rem 128
260 DATA251,3,16,28,201,26,208,15:rem 125
270 DATA238,251,3,173,251,3,201,16
                                     :rem 175
280 DATA144,3,206,251,3,16,9,201     :rem 74
290 DATA33,208,5,169,0,141,249,3     :rem 86
300 DATA96,72,152,72,206,248,3,208
                                     :rem 189
310 DATA8,169,9,141,248,3,32,13      :rem 35
320 DATA26,172,251,3,185,228,31,201
                                     :rem 225
330 DATA33,144,5,169,1,141,250,3     :rem 73
340 DATA169,10,153,228,31,173,14,144
                                     :rem 24
350 DATA240,3,206,14,144,104,168,104
                                     :rem 14
360 DATA76,191,234                   :rem 172
500 DATA160,15,162,23,169,206,133,251
                                     :rem 69
510 DATA169,31,133,252,133,254,169,228
                                     :rem 134
520 DATA133,253,177,251,145,253,165,251
                                     :rem 181
530 DATA56,233,22,133,251,165,252,233
                                     :rem 73
540 DATA0,133,252,165,253,56,233,22
                                     :rem 228
550 DATA133,253,165,254,233,0,133,254
                                     :rem 74
560 DATA202,208,223,136,16,204,96,234
                                     :rem 77
570 DATA234,234,169,32,160,21,153,234
                                     :rem 77
580 DATA29,136,16,250,76,74,27,165
                                     :rem 203
590 DATA142,74,32,167,27,168,165,251
                                     :rem 45
600 DATA153,234,29,165,252,153,235,29
                                     :rem 84
610 DATA165,253,153,0,30,165,254,153
                                     :rem 23
620 DATA1,30,96,32,148,224,165,142
                                     :rem 182
630 DATA201,0,144,13,169,32,133,251
                                     :rem 219
640 DATA133,252,133,253,133,254,76,160
                                     :rem 127
650 DATA27,201,37,144,19,169,33,133
                                     :rem 239
660 DATA251,169,34,133,252,169,59,133
                                     :rem 94
670 DATA253,169,60,133,254,76,160,27
                                     :rem 42
680 DATA169,35,133,251,169,36,133,252
                                     :rem 92
690 DATA169,46,133,253,169,47,133,254
                                     :rem 101
700 DATA32,58,27,32,0,27,96,32       :rem 242
710 DATA148,224,165,142,74,74,74,74
                                     :rem 250
720 DATA168,185,0,30,121,1,30,201:rem 114
730 DATA64,208,236,152,96            :rem 10
```

Program 3:

Sno-Cat—VIC Main Program

```
0 DIMNO%(14,2):GOSUB5000:GOSUB1050
                                     :rem 223
1 GOSUB1000                          :rem 114
5 SYSSC:IFPEEK(P)<>.THEN3000         :rem 243
10 D=D-1:D$=STR$(INT(D)):PRINT"{HOME}
   {3 DOWN}"SPC(TW-LEN(D$))INT(D):rem 201
20 PRINT"{DOWN}"SPC(18)RIGHT$(TI$,3):SYSS
   C                                 :rem 198
25 IFD=10THENGOSUB2000               :rem 73
30 IFD=.THEN2500                     :rem 151
40 IFD/HU=INT(D/HU)THENG=G+O:PRINT"{HOME}
   {9 DOWN}"SPC(18)GE$(G):POKEV1,TT:POKEV
   O,15:Q=Q-33                       :rem 175
50 FORTD=OTOQ:NEXT:GOTO5             :rem 133
1000 VO=36878:V1=36876:V2=36875:N=36877:T
     I$="000000":D=400:HU=100:O=1:TT=220:
     Q=100:G=0                       :rem 32
1020 SC=7011:P=1018:TW=21            :rem 117
1025 POKEN,0:POKEV1,0                :rem 237
1026 PRINT"{CLR}{3 DOWN}{2 SPACES}CHOOSE
     {SPACE}LEVEL (0-9)":PRINT"{DOWN}
     {3 SPACES}9 IS THE HARDEST":POKE198,
     0                               :rem 196
1027 GETA$:IFA$<"0"ORA$>"9"THEN1027
                                     :rem 167
1028 :POKE7017,VAL(A$)*10+80         :rem 230
1030 PRINT"{CLR}":POKE36879,25:POKE36869,
     255:SYS6656:POKE1018,0:POKE1019,11
                                     :rem 215
1040 FORK=0TO15:FORJ=38400TO38884STEP22:P
     OKEJ+K,13:NEXT:NEXT:POKE36878,32
                                     :rem 27
1045 PRINT"{BLK}{HOME}{2 DOWN}"SPC(17)"DI
     ST                              :rem 85
1046 PRINT"{2 DOWN}"SPC(17)"TIME":rem 208
1047 PRINT"{9 DOWN}{3 LEFT}S{DOWN}{LEFT}N
     {DOWN}{LEFT}O{DOWN}{LEFT}C{DOWN}
     {LEFT}A{DOWN}{LEFT}T{OFF}"      :rem 205
```

```
1048 FORI=7696TO8180STEP22:POKEI,37 :POKE
     I+30720,0:NEXT:RETURN          :rem 58
1050 FORK=7168TO7679:POKEK,PEEK(K+25600):
     NEXT                           :rem 205
1055 FORI=1TO3:READGE$(I):NEXT      :rem 88
1056 FORI=1TO14:READNO%(I,1),NO%(I,2):NEX
     T                              :rem 224
1060 READA:IFA=-1THENRETURN         :rem 216
1065 FORI=0TO7:READB:POKE7168+A*8+I,B:NEX
     T:GOTO1060                     :rem 125
1090 FORI=7696TO8180STEP22:POKEI,37 :POKE
     I+30720,0:NEXT:RETURN          :rem 55
2000 POKE7017,0:POKE7690,38:POKE7712,37:R
     ETURN                          :rem 220
2500 FORN=1TO14:POKEVO,15:POKEV1,NO%(N,1)
     :FORTD=1TONO%(N,2):NEXT:NEXT:rem 103
3000 POKEV1,.:POKEN,130:POKEVO,15:FORTD=1
     TO1500:NEXT                    :rem 243
3010 POKE7017,0:FORT=1TO12:SYSSC:FORK=384
     00+T*22TO38421+T*22:POKEK,0:NEXT:NEX
     T                              :rem 63
3015 PRINT"{HOME}{3 SPACES}GAME OVER":PRI
     NT"PLAY AGAIN (Y,N)":FORT=1TO10:SYSS
     C                              :rem 126
3020 FORTD=1TO100:NEXT:NEXT:GOTO4000
                                    :rem 15
4000 POKE198,0                      :rem 241
4005 GETA$:IFA$=""THEN4005          :rem 181
4010 IFA$="Y"THEN1                  :rem 243
4015 IFA$<>"N"THEN4005              :rem 194
4020 POKE36869,240:PRINT"{CLR}":END
                                    :rem 118
5000 O=7680:PRINT"{CLR}":POKE36879,8:FORA
     =7920TO8084STEP21:POKEA,78:NEXT:FORA
     =8064TO8067                    :rem 208
5010 POKEA,99:NEXT:POKE8041,77:S$="SNOCAT
     ":Q=7881                       :rem 183
5020 FORL=1TO6:B=ASC(MID$(S$,L,1))-64:FOR
     A=7898TO8024STEP21:POKEO,32:POKEA,B:
     O=A                            :rem 75
5025 GOSUB5100:NEXT                 :rem 139
5030 FORA=8045TO8042STEP-1:POKEO,32:POKEA
     ,B:O=A:GOSUB5100:NEXT:FORA=8019TOQST
     EP-23                          :rem 157
5035 POKEO,32:POKEA,B:O=A:GOSUB5100:NEXT
                                    :rem 244
5040 Q=Q+23:O=7680                  :rem 211
5050 POKE36878,15:FORN=1TOL:POKE36876,180
     +N*10:FORT=1TO50:NEXT:POKE36876,0:NE
     XT:NEXT                        :rem 208
5100 FORTD=1TO75:NEXT:RETURN        :rem 85
62012 DATA"2ND{2 DOWN}{4 LEFT}GEAR","3RD"
     ,"4TH"                         :rem 75
62013 DATA215,225,207,75,207,75,207,75,21
     5,225,207,225,201,75,201,75,201,75,
     201,75                         :rem 249
62014 DATA 209,75,209,75,207,225,195,225
                                    :rem 195
63000 DATA10,195,195,235,255,255,235,195,
     195                            :rem 188
63001 DATA11,0,0,48,48,112,176,176,112
                                    :rem 114
63002 DATA33,2,2,10,42,10,42,10,42
                                    :rem 148
63003 DATA34,0,0,128,160,128,160,128,160
                                    :rem 211
63004 DATA35,0,0,0,0,0,15,63,15     :rem 4
63005 DATA36,0,0,0,0,0,192,240,255
                                    :rem 159
63006 DATA42,53,53,53,53,49,48,0,0
                                    :rem 184
63007 DATA43,112,112,112,112,48,48,0,0
                                    :rem 105
63008 DATA59,170,10,42,170,3,3,3,3,32,0,0
     ,0,0,0,0,0,0                   :rem 29
63009 DATA60,168,128,160,168,0,0,0,0,38,1
     74,174,174,255,255,174,174,174
                                    :rem 233
63010 DATA46,63,15,15,3,0,0,0,0,37,192,19
     2,192,192,192,192,192,192      :rem 220
63011 DATA47,255,252,252,252,48,0,0,0,-1
                                    :rem 205
```

Sprite Magic

Note: Be sure to use MLX before typing in this program. (Article on page 70.)

```
49152 :076,032,195,000,001,003,051
49158 :004,032,184,192,169,004,079
49164 :133,252,169,000,133,251,182
49170 :133,167,169,216,133,168,236
49176 :169,021,141,040,002,169,054
49182 :003,141,041,002,160,000,121
49188 :177,253,170,173,048,002,091
49194 :240,003,076,138,192,169,092
49200 :207,145,251,138,010,170,201
49206 :176,008,173,003,192,145,239
49212 :167,076,069,192,173,004,229
49218 :192,145,167,200,192,008,202
49224 :208,221,024,165,251,105,022
49230 :008,133,251,133,167,165,167
49236 :252,105,000,133,252,105,163
49242 :212,133,168,230,253,208,014
49248 :002,230,254,206,041,002,063
49254 :173,041,002,208,183,024,221
49260 :165,251,105,016,133,251,005
49266 :133,167,165,252,105,000,168
49272 :133,252,105,212,133,168,099
49278 :206,040,002,173,040,002,077
49284 :240,003,076,029,192,096,000
49290 :134,097,169,000,141,042,209
49296 :002,006,097,046,042,002,083
49302 :006,097,046,042,002,174,005
49308 :042,002,169,207,145,251,204
49314 :200,169,247,145,251,136,030
49320 :189,003,192,145,167,200,040
49326 :145,167,200,192,008,208,070
49332 :215,076,074,192,169,000,138
49338 :133,254,173,043,002,133,156
49344 :253,006,253,038,254,006,234
49350 :253,038,254,006,253,038,016
49356 :254,006,253,038,254,006,247
49362 :253,038,254,006,253,038,028
49368 :254,096,032,184,192,160,110
49374 :000,177,253,073,255,145,101
49380 :253,200,192,064,208,245,110
49386 :096,032,184,192,160,062,192
49392 :136,136,177,253,010,008,192
49398 :200,200,162,003,177,253,217
49404 :040,042,008,145,253,136,108
49410 :202,208,245,040,192,255,120
49416 :208,230,096,032,184,192,182
49422 :160,000,200,200,177,253,236
49428 :074,008,136,136,162,003,027
49434 :177,253,040,106,008,145,243
49440 :253,200,202,208,245,040,156
49446 :192,063,208,230,096,032,091
49452 :184,192,160,000,177,253,242
49458 :153,203,202,200,192,003,235
49464 :208,246,177,253,136,136,188
49470 :136,145,253,200,200,200,172
49476 :200,192,063,208,241,162,110
49482 :000,160,060,189,203,202,120
49488 :145,253,200,232,224,003,113
```

```
49494 :208,245,096,032,184,192,019
49500 :160,060,162,000,177,253,136
49506 :157,203,202,200,232,224,036
49512 :003,208,245,160,060,177,189
49518 :253,200,200,200,145,253,081
49524 :136,136,136,136,016,243,151
49530 :160,000,185,203,202,145,249
49536 :253,200,192,003,208,246,206
49542 :096,032,184,192,160,000,030
49548 :152,170,232,232,169,003,074
49554 :133,097,169,008,141,055,237
49560 :002,177,253,074,145,253,032
49566 :062,203,202,206,055,002,120
49572 :173,055,002,208,240,200,018
49578 :202,198,097,165,097,208,113
49584 :227,192,063,144,215,160,153
49590 :000,185,203,202,145,253,146
49596 :200,192,063,208,246,096,169
49602 :169,147,032,210,255,173,156
49608 :000,220,133,097,041,015,194
49614 :073,015,170,173,000,208,077
49620 :024,125,066,194,141,000,250
49626 :208,173,016,208,125,077,001
49632 :194,141,016,208,173,001,189
49638 :208,024,125,088,194,141,242
49644 :001,208,032,018,195,173,095
49650 :141,002,041,001,024,109,048
49656 :248,007,141,248,007,173,048
49662 :141,002,041,002,074,073,075
49668 :255,056,109,248,007,141,052
49674 :248,007,165,097,041,016,072
49680 :208,181,173,000,220,041,071
49686 :016,240,249,173,043,002,233
49692 :141,248,007,032,059,196,199
49698 :169,255,141,000,208,169,208
49704 :000,141,016,208,169,128,190
49710 :141,001,208,076,177,194,075
49716 :032,184,192,160,000,152,004
49722 :145,253,200,192,063,208,095
49728 :249,096,000,000,000,000,153
49734 :255,255,255,000,001,001,069
49740 :001,000,000,000,000,255,076
49746 :255,255,000,000,000,000,080
49752 :000,255,001,000,000,255,087
49758 :001,000,000,255,001,018,113
49764 :083,080,082,073,084,069,059
49770 :032,077,065,071,073,067,235
49776 :146,095,069,082,082,079,153
49782 :082,032,079,078,032,083,248
49788 :065,086,069,047,076,079,034
49794 :065,068,095,018,084,146,094
49800 :065,080,069,032,079,082,031
49806 :032,018,068,146,073,083,050
49812 :075,063,095,070,073,076,088
49818 :069,078,065,077,069,058,058
49824 :095,069,078,084,069,082,125
49830 :032,067,079,076,079,082,069
49836 :032,075,069,089,095,169,189
49842 :099,160,194,133,251,132,123
49848 :252,160,040,169,032,153,222
49854 :191,007,136,208,250,177,135
49860 :251,200,201,095,208,249,120
49866 :136,132,097,152,074,073,098
49872 :255,056,105,020,168,162,206
49878 :024,024,032,240,255,169,190
49884 :146,032,210,255,160,000,255
49890 :177,251,032,210,255,200,071
49896 :196,097,144,246,096,133,120
49902 :251,132,252,160,040,169,218
49908 :032,153,191,007,136,208,203
49914 :250,162,024,160,000,024,102
49920 :032,240,255,160,000,177,096
49926 :251,201,095,240,006,032,063
49932 :210,255,200,208,244,096,201
49938 :174,053,002,240,008,160,143
49944 :000,200,208,253,202,208,071
49950 :250,096,169,147,032,210,166
49956 :255,169,000,141,134,002,225
49962 :141,056,002,169,008,032,194
49968 :210,255,169,128,141,138,065
49974 :002,169,048,141,053,002,213
49980 :169,255,141,043,002,169,071
49986 :000,141,048,002,173,006,180
49992 :192,141,038,208,173,004,060
49998 :192,141,037,208,141,039,068
50004 :208,032,007,192,169,255,179
50010 :141,000,208,169,128,141,109
50016 :001,208,173,043,002,141,152
50022 :248,007,169,001,141,021,177
50028 :208,169,000,141,028,208,094
50034 :169,012,141,033,208,141,050
50040 :032,208,141,044,002,141,176
50046 :045,002,032,177,194,032,096
50052 :059,196,032,007,192,032,138
50058 :030,196,173,000,220,072,061
50064 :041,015,073,015,141,046,219
50070 :002,104,041,016,141,047,245
50076 :002,032,228,255,240,006,151
50082 :032,238,196,076,134,195,009
50088 :032,018,195,173,047,002,123
50094 :208,003,032,089,196,032,222
50100 :030,196,173,047,002,073,189
50106 :016,141,052,002,173,046,104
50112 :002,240,195,174,046,002,083
50118 :189,066,194,172,048,002,101
50124 :240,001,010,024,109,044,120
50130 :002,141,044,002,024,173,084
50136 :045,002,125,088,194,141,043
50142 :045,002,174,044,002,016,249
50148 :017,162,000,142,044,002,083
50154 :162,023,173,048,002,240,114
50160 :002,162,022,142,044,002,102
50166 :174,044,002,224,024,144,090
50172 :005,162,000,142,044,002,095
50178 :172,045,002,016,005,160,146
50184 :020,140,045,002,172,045,176
50190 :002,192,021,144,005,160,026
50196 :000,140,045,002,032,030,013
50202 :196,076,134,195,174,045,078
50208 :002,172,044,002,032,240,012
50214 :255,164,211,173,048,002,123
50220 :208,005,169,032,145,209,044
50226 :096,169,032,145,209,200,133
50232 :145,209,096,162,000,160,060
50238 :030,024,032,240,255,169,044
50244 :018,032,210,255,174,043,032
50250 :002,142,248,007,169,000,130
50256 :032,205,189,169,032,032,227
50262 :210,255,096,032,184,192,031
50268 :173,045,002,010,109,045,220
50274 :002,133,097,173,044,002,037
50280 :074,074,074,024,101,097,036
50286 :168,173,044,002,041,007,033
50292 :073,007,170,232,134,097,061
50298 :056,169,000,042,202,208,031
50304 :252,174,048,002,208,047,091
50310 :133,097,173,052,002,208,031
50316 :016,169,000,141,049,002,005
50322 :177,253,037,097,208,005,155
50328 :169,001,141,049,002,165,167
```

```
50334 :097,073,255,049,253,174,035
50340 :049,002,240,002,005,097,047
50346 :145,253,173,056,002,240,015
50352 :003,032,030,202,096,133,160
50358 :098,074,005,098,133,098,176
50364 :174,052,002,208,014,162,032
50370 :000,142,049,002,049,253,177
50376 :208,005,169,001,141,049,005
50382 :002,165,098,073,255,049,080
50388 :253,166,097,202,133,097,136
50394 :173,051,002,074,042,202,250
50400 :208,252,174,049,002,208,093
50406 :002,169,000,005,097,145,136
50412 :253,096,141,050,002,174,184
50418 :010,197,221,010,197,240,093
50424 :004,202,208,248,096,202,184
50430 :138,010,170,189,051,197,241
50436 :072,189,050,197,072,096,168
50442 :039,133,137,134,138,077,156
50448 :074,147,018,145,017,157,062
50454 :029,135,139,049,050,051,219
50460 :052,019,136,140,033,034,186
50466 :035,036,086,083,076,024,118
50472 :088,089,066,032,160,043,006
50478 :045,004,095,070,010,193,207
50484 :234,192,088,193,042,193,226
50490 :134,193,193,193,051,194,248
50496 :217,192,127,197,137,197,107
50502 :143,197,157,197,191,197,128
50508 :244,197,006,198,006,198,157
50514 :006,198,006,198,023,198,199
50520 :034,198,062,198,094,198,104
50526 :094,198,094,198,094,198,202
50532 :174,198,028,200,195,200,071
50538 :218,200,173,197,182,197,249
50544 :133,197,088,196,088,196,242
50550 :232,198,246,198,065,201,234
50556 :081,202,090,202,206,045,182
50562 :002,076,169,197,238,033,077
50568 :208,096,238,045,002,076,033
50574 :169,197,206,044,002,173,165
50580 :048,002,240,017,206,044,193
50586 :002,076,169,197,238,044,112
50592 :002,173,048,002,240,003,116
50598 :238,044,002,104,104,076,222
50604 :224,195,173,029,208,073,050
50610 :001,141,029,208,096,173,058
50616 :023,208,073,001,141,023,141
50622 :208,096,169,016,141,048,100
50628 :002,169,001,141,028,208,233
50634 :032,007,192,162,001,142,226
50640 :051,002,189,003,192,141,018
50646 :032,208,173,004,192,141,196
50652 :037,208,173,005,192,141,208
50658 :039,208,173,006,192,141,217
50664 :038,208,173,044,002,041,226
50670 :254,141,044,002,076,169,156
50676 :197,169,000,141,048,002,033
50682 :141,032,208,141,028,208,240
50688 :173,004,192,141,039,208,245
50694 :096,056,173,050,002,233,104
50700 :049,141,051,002,170,189,102
50706 :003,192,141,032,208,096,178
50712 :169,000,141,044,002,141,009
50718 :045,002,076,169,197,032,039
50724 :218,192,032,007,192,032,197
50730 :218,192,032,007,192,032,203
50736 :184,192,160,000,177,253,246
50742 :153,139,202,200,192,064,236
50748 :208,246,096,032,184,192,250
50754 :160,000,185,139,202,145,129
50760 :253,200,192,064,208,246,211
50766 :096,144,005,028,159,156,154
50772 :030,031,158,129,149,150,219
50778 :151,152,153,154,155,169,000
50784 :161,160,194,032,181,194,250
50790 :032,133,202,162,000,221,084
50796 :079,198,240,008,232,224,065
50802 :016,208,246,076,177,194,007
50808 :056,173,050,002,233,033,155
50814 :168,138,153,003,192,173,185
50820 :048,002,208,009,173,004,064
50826 :192,141,039,208,076,163,189
50832 :198,173,004,192,141,037,121
50838 :208,173,005,192,141,039,140
50844 :208,173,006,192,141,038,146
50850 :208,174,051,002,189,003,021
50856 :192,141,032,208,076,177,226
50862 :194,169,210,160,198,032,113
50868 :181,194,032,228,255,056,102
50874 :233,048,048,248,201,010,206
50880 :176,244,133,097,056,169,043
50886 :009,229,097,010,010,010,051
50892 :141,053,002,076,177,194,079
50898 :067,085,082,083,079,082,176
50904 :032,086,069,076,079,067,113
50910 :073,084,089,032,040,048,076
50916 :045,057,041,063,095,173,190
50922 :043,002,201,255,240,006,213
50928 :238,043,002,032,059,196,042
50934 :096,206,043,002,032,184,041
50940 :192,165,046,197,254,144,226
50946 :004,238,043,002,096,032,161
50952 :059,196,096,160,000,140,147
50958 :055,002,169,164,032,210,134
50964 :255,169,157,032,210,255,074
50970 :032,133,202,172,055,002,110
50976 :133,097,169,032,032,210,193
50982 :255,169,157,032,210,255,092
50988 :165,097,201,013,240,043,035
50994 :201,020,208,013,192,000,172
51000 :240,211,136,169,157,032,233
51006 :210,255,076,013,199,041,088
51012 :127,201,032,144,196,192,192
51018 :020,240,192,165,097,153,173
51024 :000,002,032,210,255,169,236
51030 :000,133,212,200,076,013,208
51036 :199,169,095,153,000,002,198
51042 :152,096,032,231,255,169,009
51048 :133,160,194,032,181,194,230
51054 :032,133,202,162,001,201,073
51060 :084,240,011,162,008,201,054
51066 :068,240,005,104,104,076,207
51072 :177,194,141,054,002,160,088
51078 :000,169,001,032,186,255,009
51084 :169,151,160,194,032,237,059
51090 :194,032,011,199,208,007,029
51096 :173,054,002,201,084,208,106
51102 :237,173,054,002,201,068,125
51108 :208,066,169,064,141,020,064
51114 :002,169,048,141,021,002,041
51120 :169,058,141,022,002,160,216
51126 :000,185,000,002,153,023,033
51132 :002,200,204,055,002,208,091
51138 :244,169,044,153,023,002,061
51144 :169,080,153,024,002,173,033
51150 :050,002,201,083,208,012,250
51156 :169,044,153,025,002,169,006
51162 :087,153,026,002,200,200,118
51168 :200,200,200,200,200,076,020
```

```
51174 :246,199,160,000,185,000,252
51180 :002,153,020,002,200,204,049
51186 :055,002,208,244,152,162,041
51192 :020,160,002,032,189,255,138
51198 :169,160,133,178,096,083,049
51204 :065,086,069,032,065,076,141
51210 :076,032,070,082,079,077,170
51216 :032,072,069,082,069,063,147
51222 :032,040,089,047,078,041,093
51228 :095,032,100,199,032,184,158
51234 :192,169,003,160,200,032,022
51240 :181,194,032,133,202,201,215
51246 :089,208,007,162,000,160,160
51252 :064,076,067,200,024,165,136
51258 :253,105,064,170,165,254,045
51264 :105,000,168,165,253,133,120
51270 :251,165,254,133,252,032,133
51276 :225,200,169,251,032,216,145
51282 :255,176,011,032,183,255,226
51288 :208,006,032,235,200,076,077
51294 :177,194,032,235,200,032,196
51300 :231,255,173,054,002,201,248
51306 :068,240,013,169,114,160,102
51312 :194,032,181,194,032,133,110
51318 :202,076,177,194,169,000,168
51324 :032,189,255,169,015,162,178
51330 :008,160,015,032,186,255,018
51336 :032,192,255,162,015,032,056
51342 :198,255,160,000,032,207,226
51348 :255,201,013,240,007,153,249
51354 :000,002,200,076,146,200,010
51360 :169,095,153,000,002,032,099
51366 :204,255,169,000,160,002,188
51372 :032,181,194,162,015,032,020
51378 :201,255,169,073,032,210,094
51384 :255,169,013,032,210,255,094
51390 :032,231,255,076,116,200,076
51396 :032,100,199,032,225,200,216
51402 :032,184,192,169,000,166,177
51408 :253,164,254,032,213,255,099
51414 :176,136,076,235,200,169,182
51420 :004,141,136,002,000,169,160
51426 :000,141,021,208,169,147,144
51432 :076,210,255,169,001,141,060
51438 :021,208,169,147,032,210,001
51444 :255,032,059,196,032,007,057
51450 :192,076,177,194,248,169,026
51456 :000,141,000,001,141,001,028
51462 :001,224,000,240,021,202,182
51468 :024,173,000,001,105,001,060
51474 :141,000,001,173,001,001,079
51480 :105,000,141,001,001,076,092
51486 :007,201,216,173,001,001,117
51492 :009,048,141,002,001,173,154
51498 :000,001,041,240,074,074,216
51504 :074,074,009,048,141,001,139
51510 :001,173,000,001,041,015,029
51516 :009,048,141,000,001,096,099
51522 :056,165,045,233,002,133,188
51528 :045,165,046,233,000,133,182
51534 :046,169,001,133,097,169,181
51540 :008,133,098,169,000,133,113
51546 :057,133,058,160,000,177,163
51552 :097,200,017,097,240,027,006
51558 :160,002,177,097,133,057,216
51564 :200,177,097,133,058,160,165
51570 :000,177,097,072,200,177,069
51576 :097,133,098,104,133,097,014
51582 :076,093,201,024,165,057,230
51588 :105,001,133,057,165,058,139
51594 :105,000,133,058,032,184,138
51600 :192,160,000,132,098,160,118
51606 :000,024,165,045,105,037,014
51612 :145,045,200,165,046,105,094
51618 :000,145,045,200,165,057,006
51624 :145,045,200,165,058,145,158
51630 :045,200,169,131,145,045,141
51636 :200,132,097,164,098,132,235
51642 :098,177,253,170,032,254,146
51648 :200,164,097,173,002,001,061
51654 :145,045,173,001,001,200,251
51660 :145,045,173,000,001,200,000
51666 :145,045,200,169,044,145,190
51672 :045,200,132,097,164,098,184
51678 :200,152,041,007,208,213,019
51684 :132,098,164,097,136,169,000
51690 :000,145,045,160,000,177,249
51696 :045,072,200,177,045,133,144
51702 :046,104,133,045,230,057,093
51708 :208,002,230,058,164,098,244
51714 :192,064,208,143,160,000,001
51720 :152,145,045,200,145,045,228
51726 :024,165,045,105,002,133,232
51732 :045,165,046,105,000,133,002
51738 :046,076,094,166,032,135,063
51744 :193,173,045,002,010,109,052
51750 :045,002,168,162,000,185,088
51756 :203,202,157,011,203,200,252
51762 :232,224,003,208,244,032,225
51768 :135,193,173,045,002,010,102
51774 :109,045,002,168,162,000,036
51780 :177,253,029,011,203,145,118
51786 :253,200,232,224,003,208,170
51792 :243,096,173,056,002,073,211
51798 :001,141,056,002,096,032,158
51804 :184,192,160,000,162,060,082
51810 :169,003,133,097,177,253,162
51816 :157,203,202,200,232,198,016
51822 :097,165,097,208,243,138,034
51828 :056,233,006,170,016,232,061
51834 :160,062,185,203,202,145,055
51840 :253,136,016,248,096,032,141
51846 :228,255,240,251,096,013,193
```

Campaign Manager

See special instructions in article before entering this program. (Article on page 46.)

```
2049 :011,008,010,000,158,050,238
2055 :048,054,049,000,000,000,158
2061 :032,110,012,032,241,012,196
2067 :032,122,017,032,108,031,105
2073 :069,250,204,204,204,204,136
2079 :220,192,000,000,000,005,192
2085 :229,255,167,255,255,255,173
2091 :255,178,030,128,000,000,122
2097 :219,095,250,031,255,255,130
2103 :255,255,143,045,000,004,245
2109 :245,037,255,255,031,255,115
2115 :255,255,241,197,250,076,061
2121 :255,248,095,095,255,255,252
2127 :255,255,143,191,175,245,063
2133 :255,115,037,245,255,255,223
2139 :255,252,204,254,250,247,017
2145 :035,076,032,015,247,255,245
2151 :255,255,255,250,254,162,254
2157 :250,047,018,000,095,021,028
2163 :255,255,227,255,092,252,171
2169 :204,060,204,000,000,127,204
2175 :175,255,255,255,250,255,036
```

```
2181 :204,060,207,176,000,001,013
2187 :242,255,255,191,255,239,040
2193 :175,250,247,224,000,000,017
2199 :000,001,051,127,255,255,072
2205 :242,255,255,240,000,000,125
2211 :079,160,128,000,119,255,136
2217 :047,225,035,127,000,000,091
2223 :013,255,000,096,000,007,034
2229 :176,000,000,000,126,000,227
2235 :000,211,058,000,112,000,056
2241 :002,000,000,000,001,250,190
2247 :000,016,000,160,000,000,119
2253 :000,000,000,000,000,001,206
2259 :032,000,000,000,000,000,243
2265 :032,227,008,032,041,009,054
2271 :032,078,009,096,173,014,113
2277 :220,041,254,141,014,220,095
2283 :165,001,041,251,133,001,059
2289 :169,209,133,252,169,057,206
2295 :133,254,160,000,132,251,153
2301 :132,253,177,251,145,253,184
2307 :136,208,249,198,252,198,220
2313 :254,169,055,197,254,208,122
2319 :239,165,001,009,004,133,054
2325 :001,173,014,220,009,001,183
2331 :141,014,220,173,024,208,039
2337 :041,240,009,014,141,024,246
2343 :208,096,169,057,133,252,186
2349 :133,254,169,080,133,251,041
2355 :169,208,133,253,032,068,146
2361 :009,169,024,133,251,169,044
2367 :216,133,253,198,254,160,253
2373 :039,177,251,145,253,136,046
2379 :016,249,096,169,255,141,233
2385 :003,056,169,240,141,002,180
2391 :056,169,015,141,001,056,013
2397 :162,000,142,000,056,134,075
2403 :251,138,032,117,009,138,016
2409 :032,114,009,232,224,016,220
2415 :208,243,096,234,074,074,016
2421 :041,003,168,185,000,056,058
2427 :160,003,145,251,136,016,066
2433 :251,230,251,230,251,230,036
2439 :251,230,251,096,169,054,162
2445 :133,252,169,000,133,251,055
2451 :168,170,224,188,208,001,082
2457 :096,189,025,008,072,074,105
2463 :056,106,074,074,145,251,097
2469 :032,181,009,104,041,015,035
2475 :009,032,145,251,032,181,053
2481 :009,232,208,224,201,032,059
2487 :208,004,009,192,145,251,224
2493 :200,192,025,240,001,096,175
2499 :169,000,145,251,168,024,184
2505 :169,026,101,251,133,251,108
2511 :144,002,230,252,096,012,175
2517 :001,003,002,014,160,004,141
2523 :185,212,009,153,032,208,250
2529 :136,016,247,173,017,208,254
2535 :009,064,141,017,208,096,254
2541 :032,247,009,032,110,010,165
2547 :032,185,010,096,169,147,114
2553 :032,210,255,160,003,032,173
2559 :087,010,169,144,032,210,139
2565 :255,169,171,032,210,255,073
2571 :169,163,032,101,010,169,143
2577 :167,032,210,255,162,015,090
2583 :160,003,032,082,010,169,223
2589 :170,032,210,255,169,154,251
2595 :032,210,255,169,160,032,125
2601 :101,010,169,144,032,210,195
2607 :255,169,165,032,210,255,109
2613 :202,208,223,160,003,032,113
2619 :082,010,169,174,032,210,224
2625 :255,169,172,032,101,010,036
2631 :169,173,032,210,255,169,055
2637 :146,032,210,255,096,169,217
2643 :013,032,210,255,169,032,026
2649 :032,210,255,136,208,250,156
2655 :169,018,032,210,255,096,107
2661 :160,025,032,210,255,136,151
2667 :208,250,096,169,004,133,199
2673 :254,169,044,133,253,169,111
2679 :054,133,252,169,000,133,092
2685 :251,169,000,168,162,015,122
2691 :177,251,208,007,032,160,198
2697 :010,202,208,246,096,145,020
2703 :253,200,208,240,041,063,124
2709 :170,189,192,055,041,192,220
2715 :017,247,145,247,096,024,163
2721 :169,026,101,251,133,251,068
2727 :144,002,230,252,169,040,236
2733 :024,101,253,133,253,144,057
2739 :002,230,254,160,000,096,153
2745 :169,015,133,249,169,216,112
2751 :133,254,169,044,133,253,153
2757 :133,247,169,004,133,248,107
2763 :169,034,133,252,169,173,109
2769 :133,251,160,024,177,251,181
2775 :201,000,240,043,133,002,066
2781 :041,063,170,189,192,055,163
2787 :041,015,145,253,169,192,018
2793 :036,002,240,025,048,008,080
2799 :189,120,034,032,147,010,003
2805 :208,015,080,007,169,192,148
2811 :032,155,010,208,006,189,083
2817 :121,034,032,147,010,234,067
2823 :136,016,203,169,025,024,068
2829 :101,251,133,251,144,002,127
2835 :230,252,198,249,208,001,133
2841 :096,169,040,024,101,247,190
2847 :133,247,144,002,230,248,011
2853 :169,040,024,101,253,133,245
2859 :253,144,165,230,254,208,017
2865 :161,173,018,208,072,101,014
2871 :162,074,074,074,168,104,199
2877 :229,162,074,141,032,208,139
2883 :140,036,208,096,031,067,133
2889 :065,077,080,065,073,071,248
2895 :078,032,077,065,078,065,218
2901 :071,069,082,013,000,162,226
2907 :018,160,008,024,032,240,061
2913 :255,162,000,189,071,011,017
2919 :240,006,032,210,255,232,054
2925 :208,245,160,005,169,001,129
2931 :141,134,002,169,018,032,099
2937 :210,255,162,040,173,134,071
2943 :002,073,003,141,134,002,226
2949 :169,163,032,210,255,202,140
2955 :208,250,136,208,235,169,065
2961 :146,076,210,255,169,146,123
2967 :133,254,169,000,133,253,069
2973 :162,000,232,236,137,036,192
2979 :240,047,189,137,036,133,177
2985 :249,041,007,133,247,165,243
2991 :249,074,074,074,074,041,249
2997 :007,133,248,160,002,032,251
3003 :230,011,165,247,160,001,233
3009 :032,230,011,169,255,160,026
3015 :005,145,253,169,005,024,032
3021 :101,253,133,253,076,159,156
3027 :011,169,000,170,168,185,146
3033 :068,034,157,000,120,232,060
3039 :232,200,192,052,208,243,070
3045 :096,145,253,200,200,145,244
3051 :253,096,169,145,133,248,255
```

```
3057 :169,000,133,247,230,247,243
3063 :133,254,170,162,000,189,131
3069 :189,036,133,249,074,074,240
3075 :074,074,133,250,189,240,195
3081 :036,133,251,074,074,133,198
3087 :252,074,074,133,253,160,193
3093 :004,162,004,181,249,072,181
3099 :041,003,024,105,001,145,090
3105 :247,104,074,074,041,003,064
3111 :024,105,003,010,010,010,201
3117 :010,017,247,145,247,136,079
3123 :202,016,226,230,247,160,108
3129 :002,169,015,049,247,170,197
3135 :232,138,010,010,010,010,217
3141 :133,002,138,005,002,145,238
3147 :247,136,208,235,230,247,098
3153 :230,247,230,247,230,247,232
3159 :230,254,166,254,224,051,242
3165 :208,157,096,169,255,141,095
3171 :015,212,169,128,141,018,014
3177 :212,141,024,212,096,162,184
3183 :064,169,000,157,000,143,132
3189 :157,064,143,202,208,247,114
3195 :169,128,141,138,002,169,102
3201 :008,032,210,255,032,149,047
3207 :011,032,250,026,032,108,082
3213 :027,032,128,023,032,139,010
3219 :009,032,030,028,032,217,239
3225 :008,032,237,011,032,217,178
3231 :009,032,237,009,169,158,005
3237 :032,210,255,032,090,011,027
3243 :032,030,020,032,050,011,090
3249 :032,026,031,173,107,031,065
3255 :240,245,032,217,009,032,190
3261 :096,012,162,004,160,005,116
3267 :032,163,028,141,021,143,211
3273 :141,035,037,162,007,160,231
3279 :009,032,163,028,162,000,089
3285 :160,000,201,000,240,007,053
3291 :041,001,240,002,202,200,137
3297 :136,142,015,143,140,079,112
3303 :143,032,046,017,208,003,168
3309 :076,157,012,096,169,000,235
3315 :141,036,037,169,128,133,119
3321 :247,169,143,133,248,169,078
3327 :005,133,002,160,005,162,210
3333 :003,173,027,212,041,003,208
3339 :149,249,202,208,246,169,210
3345 :001,037,250,024,105,001,179
3351 :101,251,101,252,145,247,096
3357 :136,208,228,160,006,173,172
3363 :027,212,041,003,170,192,168
3369 :008,240,010,192,009,240,228
3375 :006,173,021,143,240,002,120
3381 :232,232,232,138,145,247,255
3387 :200,192,011,208,226,173,045
3393 :027,212,041,063,240,249,129
3399 :201,052,176,245,145,247,113
3405 :200,173,015,143,145,247,232
3411 :208,009,173,027,212,041,241
3417 :007,010,010,145,247,032,028
3423 :220,014,198,002,208,157,126
3429 :160,000,140,045,017,169,120
3435 :128,133,247,169,143,133,036
3441 :248,173,045,017,201,005,034
3447 :176,236,170,240,006,032,211
3453 :220,014,202,208,250,238,233
3459 :045,017,160,005,177,247,014
3465 :153,015,143,136,208,248,016
3471 :160,006,162,000,177,247,127
3477 :157,027,143,200,232,224,108
3483 :005,208,245,177,247,141,154
3489 :012,143,141,010,143,200,042
3495 :177,247,141,013,143,032,152
3501 :228,014,032,238,014,208,139
3507 :003,076,106,013,032,046,199
3513 :017,240,169,162,000,134,139
3519 :248,160,006,024,032,240,133
3525 :255,173,021,143,205,035,005
3531 :037,240,002,162,012,134,022
3537 :247,189,158,020,240,006,045
3543 :032,210,255,232,208,245,117
3549 :169,063,032,210,255,166,092
3555 :247,160,010,169,044,157,246
3561 :158,020,232,136,208,249,212
3567 :032,228,255,240,251,201,166
3573 :013,240,039,201,032,240,242
3579 :008,201,065,144,239,201,085
3585 :091,176,235,230,248,166,123
3591 :248,224,011,240,019,164,145
3597 :247,153,158,020,041,063,183
3603 :157,005,004,230,247,169,063
3609 :047,157,006,004,208,208,143
3615 :032,038,015,032,046,017,211
3621 :240,149,173,015,143,041,030
3627 :002,024,109,016,143,010,091
3633 :109,018,143,141,022,143,113
3639 :173,017,143,010,010,105,001
3645 :009,056,237,019,143,141,154
3651 :023,143,173,027,212,041,174
3657 :031,010,109,023,143,105,238
3663 :032,141,008,143,173,015,079
3669 :143,041,004,109,019,143,032
3675 :010,109,019,143,109,020,245
3681 :143,141,024,143,010,109,155
3687 :018,143,105,048,141,009,055
3693 :143,173,020,143,009,008,093
3699 :109,016,143,141,025,143,180
3705 :173,015,143,041,007,024,012
3711 :109,018,143,109,017,143,154
3717 :141,026,143,162,000,173,010
3723 :012,143,232,221,127,036,142
3729 :176,250,142,032,143,142,006
3735 :011,143,142,033,143,032,143
3741 :132,027,173,021,143,205,090
3747 :035,037,240,003,076,241,027
3753 :012,173,015,143,041,003,044
3759 :141,129,143,032,243,027,122
3765 :169,000,141,129,143,174,169
3771 :033,143,189,127,036,168,115
3777 :202,189,127,036,170,202,095
3783 :032,247,027,032,132,027,184
3789 :173,021,143,205,035,037,051
3795 :208,213,032,250,026,032,204
3801 :108,027,096,169,016,024,145
3807 :101,247,133,247,096,032,055
3813 :237,009,032,205,021,032,253
3819 :038,015,096,169,015,133,189
3825 :253,169,022,133,254,169,217
3831 :029,133,167,162,240,160,114
3837 :016,032,184,020,173,021,187
3843 :143,240,013,162,010,189,248
3849 :117,020,041,063,157,156,051
3855 :006,202,208,245,173,021,102
3861 :143,205,035,037,240,003,172
3867 :238,125,006,162,020,160,226
3873 :021,032,163,028,096,174,035
3879 :021,143,189,040,037,032,245
3885 :210,255,169,017,133,253,058
3891 :169,025,133,254,169,000,033
3897 :133,167,162,081,160,016,008
3903 :032,184,020,169,031,032,019
3909 :210,255,169,020,133,253,085
3915 :169,025,133,254,169,009,066
3921 :133,167,162,171,160,016,122
3927 :032,184,020,162,019,232,224
```

```
3933 :160,015,024,032,240,255,051
3939 :162,049,138,032,210,255,177
3945 :232,224,055,208,247,056,103
3951 :032,240,255,224,024,208,070
3957 :230,173,012,143,010,170,087
3963 :189,220,033,041,063,141,042
3969 :171,006,232,189,220,033,212
3975 :041,063,141,172,006,162,208
3981 :018,160,002,024,032,240,105
3987 :255,174,021,143,189,040,201
3993 :037,032,210,255,174,013,106
3999 :143,048,014,160,004,189,205
4005 :049,016,032,210,255,232,191
4011 :136,208,246,240,013,162,152
4017 :000,189,228,016,240,006,088
4023 :032,210,255,232,208,245,085
4029 :169,158,133,247,169,020,061
4035 :133,248,160,000,173,021,162
4041 :143,205,035,037,240,002,095
4047 :160,012,177,247,240,006,025
4053 :032,210,255,200,208,246,084
4059 :173,021,143,240,032,162,222
4065 :010,189,117,020,041,063,153
4071 :157,248,006,202,208,245,017
4077 :169,020,133,253,169,022,235
4083 :133,254,169,009,133,167,084
4089 :162,210,160,016,032,184,245
4095 :020,162,004,160,160,189,182
4101 :016,143,009,048,153,039,157
4107 :007,152,056,233,040,168,155
4113 :202,016,240,162,004,160,033
4119 :160,152,024,125,027,143,142
4125 :168,185,046,007,009,064,252
4131 :153,046,007,152,056,233,170
4137 :040,041,248,168,202,016,244
4143 :232,096,083,069,078,032,125
4149 :071,079,086,032,082,069,216
4155 :080,032,082,069,086,032,184
4161 :032,077,083,032,068,082,183
4167 :062,032,086,061,080,032,168
4173 :071,069,078,032,027,044,142
4179 :000,027,044,000,255,044,197
4185 :068,069,077,079,067,082,019
4191 :065,084,073,067,032,067,227
4197 :065,078,068,073,068,065,006
4203 :084,069,032,044,044,044,168
4209 :044,044,044,000,255,156,144
4215 :047,032,067,072,065,082,228
4221 :032,088,000,255,047,032,067
4227 :083,084,065,077,032,088,048
4233 :000,255,047,032,073,078,110
4239 :084,076,032,088,000,255,166
4245 :047,032,069,088,080,082,035
4251 :032,088,000,255,047,032,097
4257 :065,080,080,076,032,088,070
4263 :000,000,000,000,255,031,197
4269 :085,078,069,077,080,000,050
4275 :255,080,079,086,084,089,084
4281 :000,255,065,071,082,073,219
4287 :067,000,255,069,068,085,223
4293 :067,078,000,255,068,070,223
4299 :069,078,083,000,000,000,177
4305 :000,255,031,073,078,070,204
4311 :076,078,000,255,067,082,005
4317 :073,077,069,000,000,000,184
4323 :000,157,080,082,069,083,186
4329 :073,068,069,078,084,032,125
4335 :000,255,151,080,076,065,098
4341 :089,069,082,032,091,049,145
4347 :000,255,068,069,077,079,031
4353 :067,082,065,084,073,067,183
4359 :000,255,067,079,078,086,060
4365 :069,078,084,073,079,078,218
4371 :000,010,166,000,005,032,232
4377 :000,255,030,032,047,032,165
4383 :078,079,000,255,032,047,010
4389 :032,089,069,083,000,000,054
4395 :000,000,000,173,005,004,225
4401 :072,169,000,133,162,133,206
4407 :198,169,032,197,162,208,253
4413 :252,162,023,189,098,017,034
4419 :041,063,157,004,004,202,026
4425 :016,245,032,026,031,173,084
4431 :107,031,240,248,162,023,122
4437 :104,157,004,004,202,016,060
4443 :250,173,107,031,041,016,197
4449 :096,058,070,073,082,069,033
4455 :066,085,084,084,079,078,067
4461 :032,084,079,032,067,079,226
4467 :078,084,073,078,085,069,070
4473 :058,173,035,037,205,021,138
4479 :143,208,011,238,036,037,032
4485 :173,036,037,201,010,208,030
4491 :001,096,032,237,009,032,034
4497 :205,021,032,038,015,169,113
4503 :007,141,000,143,032,244,206
4509 :020,162,005,160,012,032,036
4515 :163,028,170,208,003,076,043
4521 :003,018,202,208,003,076,167
4527 :147,018,202,208,003,076,061
4533 :197,018,202,208,003,076,117
4539 :239,018,202,208,006,032,124
4545 :043,029,076,155,017,202,203
4551 :208,008,032,022,019,208,184
4557 :205,076,003,018,202,208,149
4563 :014,032,046,017,240,197,245
4569 :173,011,143,141,032,143,092
4575 :076,141,017,202,240,003,134
4581 :076,155,017,076,200,019,004
4587 :162,000,169,128,024,109,059
4593 :032,143,168,169,000,133,118
4599 :253,169,014,133,254,169,215
4605 :030,133,167,076,184,020,095
4611 :032,235,017,174,032,143,124
4617 :189,127,036,202,056,253,104
4623 :127,036,072,105,003,168,014
4629 :162,003,032,163,028,201,098
4635 :000,208,007,032,043,029,090
4641 :104,076,006,018,133,002,116
4647 :104,197,002,176,003,076,085
4653 :155,017,198,002,174,032,111
4659 :143,202,189,127,036,024,004
4665 :101,002,174,000,143,157,122
4671 :000,143,133,251,134,252,208
4677 :032,250,019,169,030,032,089
4683 :210,255,165,251,010,170,112
4689 :189,220,033,032,210,255,252
4695 :189,221,033,032,210,255,003
4701 :169,032,032,210,255,189,212
4707 :000,120,072,170,169,000,118
4713 :032,205,189,104,201,010,078
4719 :176,005,169,032,032,210,223
4725 :255,169,032,032,210,255,046
4731 :169,152,032,210,255,173,090
4737 :032,143,009,048,032,210,091
4743 :255,206,000,143,208,003,182
4749 :076,007,020,076,006,018,088
4755 :174,000,143,169,240,157,006
4761 :000,143,134,252,032,250,196
4767 :019,169,129,032,210,255,205
4773 :162,000,189,112,021,240,121
4779 :006,032,210,255,232,208,090
4785 :245,173,032,143,009,048,059
4791 :032,210,255,206,000,143,005
4797 :208,003,076,007,020,076,067
4803 :155,017,174,000,143,169,085
```

```
4809 :255,157,000,143,134,252,118
4815 :032,250,019,169,154,032,095
4821 :210,255,162,000,189,125,130
4827 :021,240,006,032,210,255,215
4833 :232,208,245,206,000,143,235
4839 :208,003,076,007,020,076,109
4845 :155,017,174,000,143,169,127
4851 :000,157,000,143,134,252,161
4857 :032,250,019,169,155,032,138
4863 :210,255,169,090,162,005,122
4869 :032,210,255,202,208,250,138
4875 :206,000,143,208,003,076,135
4881 :007,020,076,155,017,173,209
4887 :009,143,201,010,176,001,051
4893 :096,206,009,143,032,153,156
4899 :033,174,032,143,189,127,221
4905 :036,133,248,202,189,127,208
4911 :036,133,247,169,150,133,147
4917 :249,169,004,133,250,169,003
4923 :047,133,251,133,252,166,017
4929 :247,228,248,208,003,076,051
4935 :046,017,165,249,024,105,165
4941 :040,133,249,144,002,230,107
4947 :250,189,000,144,133,253,028
4953 :133,254,162,004,006,254,134
4959 :202,208,251,006,254,1[illegible]6,168
4965 :028,169,037,133,251,0[illegible]6,213
4971 :254,176,020,169,032,1[illegible]3,123
4977 :251,006,254,176,012,1[illegible],213
4983 :037,133,252,006,254,176,209
4989 :004,169,032,133,252,160,107
4995 :000,169,032,145,249,200,158
5001 :165,251,145,249,200,165,032
5007 :252,145,249,169,047,133,114
5013 :251,133,252,006,253,176,196
5019 :028,169,042,133,252,006,017
5025 :253,176,020,169,032,133,176
5031 :252,006,253,176,012,169,011
5037 :042,133,251,006,253,176,010
5043 :004,169,032,133,251,160,160
5049 :007,165,251,145,249,200,178
5055 :165,252,145,249,230,247,199
5061 :076,058,019,032,103,023,252
5067 :174,032,143,232,232,232,224
5073 :160,031,024,032,240,255,183
5079 :169,058,032,210,255,162,077
5085 :003,160,013,032,163,028,108
5091 :201,000,208,006,032,043,205
5097 :029,076,200,019,201,010,000
5103 :208,003,076,155,017,141,071
5109 :032,143,076,155,017,169,069
5115 :022,056,229,252,170,160,116
5121 :032,024,032,240,255,096,168
5127 :032,046,017,208,003,076,133
5133 :217,017,032,104,025,032,184
5139 :250,026,032,108,027,032,238
5145 :132,027,076,122,017,169,056
5151 :000,133,253,169,010,133,217
5157 :254,169,030,133,167,162,184
5163 :049,160,020,076,184,020,040
5169 :255,018,144,160,213,211,026
5175 :197,160,202,207,217,189,203
5181 :160,000,255,160,211,212,035
5187 :201,195,203,160,207,210,219
5193 :160,000,255,2[illegible],202,203,070
5199 :204,146,205,018,160,203,247
5205 :197,217,211,000,255,018,215
5211 :155,080,076,091,049,032,062
5217 :080,065,082,084,089,146,131
5223 :000,255,031,068,069,077,091
5229 :079,067,082,065,084,073,047
5235 :067,000,255,082,069,080,156
5241 :085,066,076,073,067,065,041
5247 :078,000,255,018,155,032,153
5253 :073,078,067,085,077,066,067
5259 :069,078,084,146,000,255,003
5265 :031,032,032,032,078,079,173
5271 :078,069,032,032,032,000,138
5277 :255,032,080,076,065,089,242
5283 :069,082,032,049,032,000,171
5289 :255,032,080,076,065,089,254
5295 :069,082,032,050,032,000,184
5301 :000,000,000,134,251,132,186
5307 :252,208,011,200,152,024,010
5313 :101,251,133,251,144,002,051
5319 :230,252,166,253,228,254,046
5325 :208,001,096,230,253,164,133
5331 :167,024,032,240,255,160,065
5337 :000,162,255,177,251,016,054
5343 :016,200,177,251,240,217,044
5349 :032,210,255,202,016,250,170
5355 :240,209,200,208,241,170,223
5361 :200,208,237,169,000,133,164
5367 :253,169,014,133,254,169,215
5373 :146,032,210,255,169,144,185
5379 :032,210,255,169,030,133,064
5385 :167,162,072,160,021,032,111
5391 :184,020,174,021,143,189,234
5397 :037,037,041,063,141,071,155
5403 :004,173,036,037,009,048,078
5409 :141,078,004,173,032,143,092
5415 :010,010,010,024,109,032,234
5421 :143,170,173,032,143,009,203
5427 :048,141,150,004,160,000,042
5433 :189,037,036,041,063,240,151
5439 :007,153,152,004,232,200,043
5445 :208,242,096,009,058,000,170
5451 :255,032,032,032,087,069,070
5457 :069,075,032,032,032,000,065
5463 :009,058,000,009,032,000,195
5469 :009,032,000,255,031,032,196
5475 :032,067,065,077,080,065,229
5481 :073,071,078,000,255,032,102
5487 :032,084,086,032,065,068,222
5493 :083,032,032,157,000,255,164
5499 :032,032,070,085,078,068,232
5505 :082,065,073,083,000,255,175
5511 :032,032,082,069,083,084,005
5517 :032,032,032,032,000,255,012
5523 :018,155,032,077,065,080,062
5529 :032,032,032,032,032,032,089
5535 :000,255,032,080,079,076,169
5541 :076,032,032,032,032,032,145
5547 :000,255,146,150,082,069,105
5553 :067,079,078,083,073,068,113
5559 :069,082,000,255,084,082,243
5565 :065,086,069,076,032,032,037
5571 :032,032,154,000,009,032,198
5577 :000,000,000,000,169,014,128
5583 :133,253,169,025,133,254,150
5589 :169,028,133,167,169,030,141
5595 :032,210,255,162,034,160,048
5601 :023,032,184,020,169,043,184
5607 :141,076,006,169,046,141,042
5613 :140,007,173,036,037,208,070
5619 :011,169,020,162,008,032,133
5625 :210,255,202,208,250,096,190
5631 :174,008,143,169,000,032,013
5637 :205,189,162,023,160,030,006
5643 :024,032,240,255,169,030,249
5649 :032,210,255,174,009,143,072
5655 :224,100,176,022,169,032,234
5661 :032,210,255,224,010,176,168
5667 :013,032,210,255,032,210,019
5673 :255,138,009,048,032,210,221
5679 :255,096,169,000,032,205,036
```

```
5685 :189,173,184,007,141,185,164
5691 :007,162,006,173,027,212,134
5697 :041,015,201,010,176,247,243
5703 :009,048,157,185,007,202,167
5709 :208,239,169,060,141,184,054
5715 :007,141,188,007,032,122,068
5721 :022,169,052,133,248,169,114
5727 :000,133,247,168,162,002,039
5733 :149,252,202,016,251,032,235
5739 :166,022,169,032,162,002,148
5745 :149,249,202,016,251,032,244
5751 :211,022,096,169,017,133,255
5757 :252,169,000,133,251,166,072
5763 :251,228,252,208,001,096,143
5769 :160,000,024,032,240,255,080
5775 :162,000,189,040,037,032,091
5781 :210,255,169,037,032,210,038
5787 :255,232,224,003,208,240,037
5793 :230,251,076,130,022,166,012
5799 :247,232,232,134,247,200,179
5805 :196,248,208,001,096,189,087
5811 :000,120,074,133,002,185,181
5817 :000,144,162,002,041,238,004
5823 :240,006,202,041,014,240,166
5829 :001,202,181,252,024,101,190
5835 :002,176,216,149,252,076,050
5841 :166,022,169,004,133,248,183
5847 :169,000,133,247,160,002,158
5853 :169,015,133,002,185,037,250
5859 :037,041,063,145,247,136,128
5865 :016,246,169,040,024,101,061
5871 :247,133,247,169,000,101,112
5877 :248,133,248,160,002,162,174
5883 :002,169,016,024,117,252,063
5889 :149,252,176,013,181,249,253
5895 :145,247,202,136,016,239,224
5901 :198,002,016,218,096,072,103
5907 :169,037,149,249,104,074,033
5913 :074,074,041,001,009,036,004
5919 :076,007,023,011,035,000,183
5925 :255,042,032,083,000,255,192
5931 :042,032,077,000,255,042,235
5937 :032,084,000,255,042,032,238
5943 :087,000,255,042,032,084,043
5949 :000,255,042,032,070,000,204
5955 :255,042,032,083,000,011,234
5961 :044,000,255,092,032,048,032
5967 :048,060,048,048,048,060,135
5973 :048,048,048,000,255,154,126
5979 :032,072,069,065,076,084,233
5985 :072,032,000,000,000,000,201
5991 :169,028,032,210,255,169,198
5997 :000,133,253,169,015,133,044
6003 :254,169,030,133,167,162,006
6009 :000,160,128,032,184,020,133
6015 :096,169,128,133,248,169,046
6021 :000,133,247,169,000,133,047
6027 :250,133,249,169,000,133,049
6033 :253,133,254,168,162,001,092
6039 :032,241,023,162,000,160,001
6045 :004,189,037,036,145,247,047
6051 :200,232,224,008,208,245,000
6057 :162,008,172,054,025,136,214
6063 :230,253,165,253,201,010,007
6069 :208,003,076,225,023,169,117
6075 :255,145,247,200,169,028,207
6081 :145,247,200,169,042,145,117
6087 :247,200,165,253,009,048,097
6093 :145,247,200,169,031,145,118
6099 :247,200,232,189,037,036,128
6105 :145,247,208,247,200,076,060
6111 :175,023,032,002,024,169,136
6117 :000,162,004,145,247,200,219
6123 :202,208,250,076,017,024,244
6129 :162,001,160,000,189,054,039
6135 :025,145,247,200,232,236,052
6141 :054,025,208,244,096,162,018
6147 :001,189,089,025,145,247,187
6153 :200,232,236,089,025,208,231
6159 :244,096,169,001,133,253,143
6165 :133,254,208,009,230,253,084
6171 :165,253,201,010,208,001,097
6177 :096,230,248,169,009,024,041
6183 :101,249,133,249,169,000,172
6189 :101,250,133,250,032,241,028
6195 :023,166,249,160,004,165,050
6201 :253,073,048,145,247,200,255
6207 :200,189,037,036,240,006,003
6213 :145,247,232,200,208,245,066
6219 :166,253,189,127,036,133,211
6225 :250,172,054,025,136,165,115
6231 :254,010,170,169,048,133,103
6237 :251,133,252,169,255,145,018
6243 :247,200,169,028,145,247,111
6249 :200,169,042,145,247,200,084
6255 :169,154,145,247,200,165,167
6261 :254,201,010,144,007,230,195
6267 :252,233,010,076,118,024,068
6273 :101,251,133,251,165,252,002
6279 :145,247,200,165,251,145,008
6285 :247,200,169,032,145,247,157
6291 :200,169,151,145,247,200,235
6297 :189,220,033,145,247,200,163
6303 :232,189,220,033,145,247,201
6309 :200,202,169,032,145,247,136
6315 :200,169,048,133,251,133,081
6321 :252,189,000,120,201,010,181
6327 :144,007,230,252,233,010,035
6333 :076,181,024,101,251,133,187
6339 :251,165,252,145,247,200,175
6345 :165,251,145,247,200,169,098
6351 :032,145,247,200,169,000,232
6357 :145,247,200,230,254,165,174
6363 :254,197,250,240,003,076,215
6369 :086,024,032,002,024,165,046
6375 :250,133,254,166,253,202,209
6381 :189,127,036,133,002,232,188
6387 :189,127,036,056,229,002,114
6393 :133,002,169,008,229,002,024
6399 :133,002,048,038,169,009,142
6405 :145,247,200,169,035,145,178
6411 :247,200,169,000,145,247,251
6417 :200,198,002,048,019,169,141
6423 :009,145,247,200,169,032,057
6429 :145,247,200,169,000,145,167
6435 :247,200,198,002,016,237,167
6441 :169,000,162,004,145,247,000
6447 :200,202,208,250,076,025,240
6453 :024,035,009,035,000,255,155
6459 :032,032,032,032,032,032,251
6465 :032,032,032,032,000,009,202
6471 :044,000,255,018,154,037,067
6477 :144,205,193,208,160,160,123
6483 :160,160,160,160,146,000,101
6489 :014,255,028,042,077,069,062
6495 :078,085,032,032,032,032,130
6501 :032,000,000,173,011,143,204
6507 :141,032,143,169,008,141,229
6513 :000,143,206,000,143,208,045
6519 :001,096,174,000,143,189,210
6525 :000,143,208,009,032,177,182
6531 :026,032,208,026,076,115,102
6537 :025,016,023,106,176,003,230
6543 :076,166,027,173,024,143,240
6549 :010,109,009,143,144,002,054
6555 :169,255,141,009,143,076,180
```

```
6561 :115,025,172,009,143,240,097
6567 :203,072,162,000,232,221,033
6573 :127,036,176,250,236,011,241
6579 :143,240,009,142,032,143,120
6585 :142,011,143,206,009,143,071
6591 :104,032,207,025,032,233,056
6597 :025,032,093,026,032,140,033
6603 :026,076,115,025,133,002,068
6609 :133,251,198,251,165,251,178
6615 :010,010,024,101,251,133,232
6621 :251,133,253,169,146,133,026
6627 :252,169,145,133,254,096,252
6633 :173,008,143,041,248,208,030
6639 :005,169,001,133,255,096,130
6645 :169,003,024,109,021,143,202
6651 :168,177,251,133,255,173,128
6657 :010,143,016,003,230,255,146
6663 :096,197,002,208,009,169,176
6669 :002,032,087,026,169,255,072
6675 :133,002,165,002,141,010,216
6681 :143,173,008,143,160,005,145
6687 :074,136,208,252,032,087,052
6693 :026,173,022,143,032,087,008
6699 :026,160,006,136,208,001,068
6705 :096,185,026,143,209,253,193
6711 :208,007,169,003,032,087,049
6717 :026,208,238,170,202,208,019
6723 :209,253,208,007,169,001,146
6729 :032,087,026,208,224,255,114
6735 :232,138,209,253,208,217,056
6741 :240,240,024,101,255,133,054
6747 :255,096,160,005,177,251,011
6753 :056,229,255,176,004,198,247
6759 :255,208,243,145,251,165,090
6765 :255,170,172,021,143,200,046
6771 :024,113,251,144,002,169,050
6777 :255,145,251,152,073,003,232
6783 :168,138,074,074,113,251,177
6789 :144,002,169,255,145,251,075
6795 :096,070,255,208,001,096,097
6801 :173,008,143,056,229,255,241
6807 :176,002,169,000,141,008,135
6813 :143,070,255,208,001,096,162
6819 :173,009,143,056,229,255,004
6825 :176,002,169,000,141,009,154
6831 :143,096,160,000,162,015,239
6837 :173,021,143,240,002,162,154
6843 :240,134,251,162,052,202,204
6849 :208,003,132,002,096,189,055
6855 :000,144,037,251,240,243,090
6861 :200,208,240,165,002,024,020
6867 :109,023,143,010,109,022,115
6873 :143,109,008,143,144,003,255
6879 :024,169,255,141,008,143,195
6885 :173,010,143,208,010,169,174
6891 :016,109,008,143,176,003,178
6897 :141,008,143,169,000,141,075
6903 :010,143,096,169,146,133,176
6909 :252,169,000,133,251,169,203
6915 :000,170,240,007,160,005,073
6921 :230,251,136,208,251,232,037
6927 :224,052,208,001,096,160,244
6933 :001,177,251,200,056,241,179
6939 :251,208,006,032,088,027,127
6945 :076,007,027,176,010,234,051
6951 :073,255,024,105,001,160,145
6957 :128,208,002,160,008,133,172
6963 :253,132,254,041,224,240,171
6969 :002,208,020,070,254,165,008
6975 :253,041,016,240,002,208,055
6981 :010,070,254,165,253,041,094
6987 :008,208,002,070,254,165,014
6993 :254,157,000,144,076,007,207
6999 :027,173,000,144,041,240,200
7005 :240,004,169,001,208,002,205
7011 :169,016,141,000,144,157,214
7017 :000,144,096,162,052,202,249
7023 :240,018,189,000,144,041,231
7029 :015,240,004,169,067,208,052
7035 :002,169,130,157,192,055,060
7041 :208,235,096,173,021,143,237
7047 :072,162,063,189,064,143,060
7053 :157,128,143,189,000,143,133
7059 :157,064,143,189,128,143,203
7065 :157,000,143,202,208,235,074
7071 :104,073,001,141,021,143,130
7077 :096,173,009,143,201,040,059
7083 :144,067,174,032,143,189,152
7089 :127,036,133,250,202,189,090
7095 :127,036,133,249,198,249,151
7101 :173,036,037,010,024,109,066
7107 :025,143,133,255,230,249,206
7113 :165,249,197,250,240,014,036
7119 :032,023,028,070,255,032,135
7125 :093,026,032,140,026,076,094
7131 :189,027,173,009,143,056,048
7137 :237,025,143,144,005,237,248
7143 :025,143,176,002,169,001,235
7149 :141,009,143,076,115,025,234
7155 :162,000,160,052,134,249,232
7161 :132,250,230,249,165,249,244
7167 :197,250,240,019,174,129,240
7173 :143,134,255,032,023,028,108
7179 :032,044,026,070,255,032,214
7185 :093,026,076,251,027,096,074
7191 :032,207,025,032,038,026,127
7197 :096,162,000,169,000,157,101
7203 :000,063,202,208,250,169,159
7209 :000,170,168,185,010,031,093
7215 :157,000,063,185,018,031,245
7221 :157,064,063,232,232,232,009
7227 :200,192,007,208,236,185,063
7233 :010,031,157,000,063,157,227
7239 :001,063,157,002,063,185,030
7245 :018,031,157,064,063,169,067
7251 :252,141,248,007,169,253,129
7257 :141,249,007,162,007,169,056
7263 :012,157,039,208,202,016,217
7269 :250,169,001,141,029,208,131
7275 :169,001,141,016,208,169,043
7281 :004,141,000,208,169,050,173
7287 :141,001,208,169,054,141,065
7293 :002,208,169,056,141,003,192
7299 :208,169,000,160,004,153,057
7305 :002,031,136,016,250,169,229
7311 :034,141,007,031,169,173,186
7317 :141,006,031,169,054,141,179
7323 :009,031,169,000,141,008,001
7329 :031,096,169,000,133,253,075
7335 :169,004,141,000,208,152,073
7341 :032,250,030,133,252,138,240
7347 :032,250,030,133,251,141,248
7353 :001,208,169,012,141,039,243
7359 :208,173,016,208,009,001,038
7365 :141,016,208,173,021,208,196
7371 :009,001,141,021,208,032,103
7377 :026,031,173,107,031,240,049
7383 :248,041,019,240,244,170,153
7389 :041,016,208,039,138,041,192
7395 :001,240,017,173,001,208,099
7401 :197,251,240,227,198,253,063
7407 :056,233,008,141,001,208,118
7413 :208,217,173,001,208,197,225
7419 :252,240,210,230,253,024,180
7425 :105,008,141,001,208,208,160
7431 :200,169,000,141,039,208,252
```

```
7437 :032,026,031,173,107,031,157
7443 :240,248,041,016,208,007,011
7449 :169,012,141,039,208,208,034
7455 :176,173,021,208,041,254,136
7461 :141,021,208,165,253,096,153
7467 :162,007,189,002,031,149,071
7473 :247,202,016,248,169,001,164
7479 :141,040,208,173,021,208,078
7485 :009,002,141,021,208,032,218
7491 :026,031,173,107,031,240,163
7497 :248,106,176,020,106,176,137
7503 :067,106,176,110,106,176,052
7509 :005,106,176,005,144,231,240
7515 :076,240,029,076,231,030,005
7521 :165,248,240,221,173,003,123
7527 :208,056,233,004,141,003,236
7533 :208,198,248,165,248,106,002
7539 :176,003,076,036,030,165,089
7545 :253,233,026,133,253,176,171
7551 :002,198,254,165,251,056,029
7557 :233,025,133,251,144,003,154
7563 :076,036,030,198,252,076,039
7569 :036,030,165,248,201,029,086
7575 :240,169,173,003,208,024,200
7581 :105,004,141,003,208,230,080
7587 :248,165,248,106,176,123,205
7593 :165,253,105,026,133,253,080
7599 :144,002,230,254,165,251,197
7605 :024,105,025,133,251,144,095
7611 :104,230,252,076,036,030,147
7617 :165,247,208,003,076,066,190
7623 :029,173,002,208,056,233,132
7629 :004,141,002,208,198,247,237
7635 :165,247,106,144,076,165,090
7641 :253,233,001,133,253,176,242
7647 :002,198,254,165,251,056,125
7653 :233,001,133,251,176,057,056
7659 :198,252,076,036,030,165,224
7665 :247,201,049,208,003,076,001
7671 :066,029,173,002,208,024,237
7677 :105,004,141,002,208,230,175
7683 :247,165,247,106,144,003,147
7689 :076,036,030,165,253,105,162
7695 :001,133,253,144,002,230,010
7701 :254,165,251,024,105,001,053
7707 :133,251,144,005,230,252,018
7713 :076,036,030,169,001,133,222
7719 :249,165,248,074,144,004,155
7725 :006,249,006,249,165,247,199
7731 :106,176,002,006,249,160,238
7737 :000,177,251,133,002,165,017
7743 :249,049,253,208,038,169,005
7749 :192,036,002,048,013,165,013
7755 :002,041,063,170,189,120,148
7761 :034,133,002,076,106,030,206
7767 :080,007,169,000,133,002,222
7773 :076,106,030,165,002,041,001
7779 :063,170,189,121,034,133,041
7785 :002,162,015,160,016,024,228
7791 :032,240,255,169,149,032,220
7797 :210,255,169,032,162,007,184
7803 :032,210,255,202,016,250,064
7809 :169,157,162,007,032,210,098
7815 :255,202,016,250,165,002,001
7821 :208,003,076,066,029,041,052
7827 :063,010,170,189,220,033,064
7833 :032,210,255,189,221,033,069
7839 :032,210,255,169,032,032,121
7845 :210,255,189,000,120,170,085
7851 :201,010,176,005,169,032,252
7857 :032,210,255,169,000,032,107
7863 :205,189,169,029,032,210,249
7869 :255,169,144,032,210,255,230
7875 :169,018,032,210,255,169,024
7881 :160,032,210,255,165,002,001
7887 :041,063,162,000,232,221,158
7893 :127,036,176,250,138,105,021
7899 :176,032,210,255,169,146,183
7905 :032,210,255,076,066,029,125
7911 :173,021,208,041,253,141,044
7917 :021,208,162,007,181,247,039
7923 :157,002,031,202,016,248,131
7929 :096,234,010,010,010,024,121
7935 :105,050,096,000,000,000,250
7941 :000,000,000,000,000,192,197
7947 :192,224,240,224,192,200,003
7953 :255,255,153,129,195,195,175
7959 :129,153,255,169,000,141,102
7965 :107,031,173,000,220,041,089
7971 :031,073,031,208,045,173,084
7977 :001,220,041,031,073,031,182
7983 :208,036,032,228,255,208,246
7989 :001,096,056,233,073,144,144
7995 :222,170,232,233,005,176,073
8001 :216,138,041,002,240,004,194
8007 :138,073,001,170,169,000,110
8013 :141,107,031,056,042,202,144
8019 :208,252,141,107,031,173,227
8025 :000,220,045,001,220,041,104
8031 :016,240,[illegible]6,169,006,101,105
8037 :162,197,[illegible]62,208,252,096,154
8043 :000,032,250,026,032,177,112
8049 :026,165,002,201,026,144,165
8055 :003,032,132,027,032,122,211
8061 :033,032,132,027,032,122,247
8067 :033,169,001,032,207,025,086
8073 :160,005,177,251,074,074,110
8079 :170,160,002,138,024,113,238
8085 :251,144,002,165,255,145,087
8091 :251,136,208,243,160,002,131
8097 :209,251,208,019,160,003,243
8103 :177,251,200,056,241,251,063
8109 :169,128,042,168,200,177,033
8115 :251,233,001,145,251,165,201
8121 :251,024,105,005,133,251,186
8127 :201,255,208,198,032,250,055
8133 :026,032,177,026,162,051,159
8139 :189,000,144,041,017,240,066
8145 :003,030,000,144,202,208,028
8151 :243,032,108,027,032,237,126
8157 :009,032,087,022,032,090,237
8163 :011,032,205,021,162,015,161
8169 :134,002,160,029,024,032,102
8175 :240,255,169,152,032,210,017
8181 :255,169,032,162,011,032,138
8187 :210,255,202,208,250,230,070
8193 :002,166,002,224,024,208,115
8199 :227,173,100,007,141,140,027
8205 :007,141,180,007,141,220,197
8211 :007,169,032,162,011,157,045
8217 :220,007,202,208,250,169,057
8223 :020,141,226,007,169,000,082
8229 :162,003,149,003,202,016,060
8235 :251,169,009,133,174,169,180
8241 :000,141,032,143,238,032,123
8247 :143,173,032,143,201,010,245
8253 :208,003,076,048,03[illegible],032,204
8259 :153,033,169,000,133,178,221
8265 :133,179,162,004,134,251,168
8271 :160,031,132,252,169,190,245
8277 :133,247,133,249,169,004,252
8283 :133,248,133,250,166,167,164
8289 :160,003,169,032,145,247,085
8295 :136,016,251,165,247,024,174
8301 :105,040,133,247,144,002,012
8307 :230,248,202,208,233,174,130
```

```
8313 :032,143,189,127,036,133,013
8319 :254,202,189,127,036,133,044
8325 :253,166,251,164,252,024,219
8331 :032,240,255,166,253,189,250
8337 :000,144,041,015,208,003,044
8343 :076,111,033,189,068,034,150
8349 :170,024,101,178,133,178,173
8355 :138,201,010,176,005,169,094
8361 :032,032,210,255,169,154,253
8367 :032,210,255,169,000,032,105
8373 :205,189,166,251,160,037,165
8379 :024,032,240,255,160,003,133
8385 :169,032,032,210,255,136,003
8391 :208,250,230,251,230,253,085
8397 :198,167,208,181,165,174,018
8403 :208,003,076,105,033,173,041
8409 :032,143,024,105,014,170,193
8415 :160,031,024,032,240,255,197
8421 :169,154,032,210,255,165,190
8427 :178,170,201,010,176,005,207
8433 :169,032,032,210,255,169,084
8439 :000,032,205,189,169,156,230
8445 :032,210,255,169,032,072,255
8451 :032,210,255,173,032,143,080
8457 :009,048,032,210,255,104,155
8463 :032,210,255,032,210,255,241
8469 :165,179,170,201,010,176,154
8475 :005,169,032,032,210,255,218
8481 :169,028,032,210,255,169,128
8487 :000,032,205,189,162,024,139
8493 :160,030,024,032,240,255,018
8499 :169,152,032,210,255,165,010
8505 :178,024,101,003,133,003,243
8511 :169,000,101,004,133,004,218
8517 :165,179,101,005,133,005,145
8523 :169,000,101,006,133,006,234
8529 :166,003,165,004,032,205,144
8535 :189,162,024,160,036,024,170
8541 :032,240,255,166,005,165,188
8547 :006,032,205,189,198,174,135
8553 :032,046,017,076,053,032,105
8559 :189,068,034,024,101,179,194
8565 :133,179,076,201,032,173,143
8571 :026,143,141,129,143,032,225
8577 :243,027,169,000,141,129,070
8583 :143,174,032,143,189,127,175
8589 :036,168,202,189,127,036,131
8595 :170,202,032,247,027,096,153
8601 :169,156,032,210,255,032,239
8607 :235,017,162,003,160,030,254
8613 :024,032,240,255,032,193,173
8619 :033,174,032,143,189,127,101
8625 :036,202,056,253,127,036,119
8631 :133,167,105,003,170,160,153
8637 :030,032,240,255,162,000,140
8643 :189,207,033,208,001,096,161
8649 :032,210,255,232,208,244,102
8655 :154,068,069,077,032,032,127
8661 :032,032,028,082,069,080,024
8667 :000,032,032,077,069,078,251
8673 :072,086,084,077,065,082,179
8679 :073,067,084,078,089,078,188
8685 :074,080,065,079,072,073,168
8691 :078,073,076,077,073,087,195
8697 :073,077,078,073,065,077,180
8703 :079,078,068,083,068,078,197
8709 :069,075,083,068,069,077,190
8715 :068,068,067,086,065,087,196
8721 :086,078,067,083,067,071,213
8727 :065,070,076,075,089,084,226
8733 :078,065,076,077,083,065,217
8739 :082,076,065,079,075,084,240
8745 :088,077,084,073,068,087,006
8751 :089,067,079,078,077,065,246
8757 :090,085,084,078,086,087,051
8763 :065,079,082,067,065,065,226
8769 :075,072,073,000,004,004,037
8775 :003,013,004,008,036,016,151
8781 :025,023,012,024,020,011,192
8787 :010,008,011,003,003,005,123
8793 :007,003,010,003,012,006,130
8799 :013,008,012,021,009,011,169
8805 :009,007,006,010,008,029,170
8811 :004,004,003,008,005,007,138
8817 :005,004,010,007,047,003,189
8823 :004,000,001,003,003,004,134
8829 :005,008,009,008,010,026,191
8835 :031,011,014,016,014,012,229
8841 :037,012,019,016,017,020,002
8847 :025,009,032,025,025,029,032
8853 :027,030,026,031,029,032,068
8859 :017,034,038,035,040,040,103
8865 :041,042,038,037,046,049,158
8871 :040,040,044,000,000,000,035
8877 :239,239,047,111,231,231,247
8883 :231,231,231,210,210,210,222
8889 :207,207,207,000,000,000,038
8895 :000,000,000,000,000,193,128
8901 :193,239,047,047,111,103,169
8907 :039,039,039,039,018,018,139
8913 :018,015,079,079,077,205,170
8919 :205,000,000,000,000,000,164
8925 :193,193,240,048,048,112,031
8931 :103,039,039,039,039,019,249
8937 :019,019,015,015,079,014,138
8943 :206,205,000,000,000,199,081
8949 :007,066,193,240,048,048,079
8955 :040,040,103,041,041,041,045
8961 :019,019,019,016,016,078,168
8967 :140,205,013,205,201,199,202
8973 :007,007,004,196,241,049,005
8979 :110,046,046,045,041,041,092
8985 :041,020,020,020,084,016,226
8991 :080,012,076,011,010,073,037
8997 :009,009,071,070,197,241,122
9003 :049,110,046,046,045,045,128
9009 :042,042,042,149,149,149,110
9015 :017,145,012,076,011,138,198
9021 :074,137,088,151,200,000,199
9027 :000,049,049,110,046,045,110
9033 :045,042,042,042,021,021,030
9039 :021,085,017,145,076,075,242
9045 :031,095,090,025,087,214,115
9051 :000,000,241,049,110,172,151
9057 :044,044,043,043,171,101,031
9063 :037,037,081,099,017,096,214
9069 :096,096,089,091,091,091,151
9075 :000,000,000,000,241,049,149
9081 :113,044,044,043,043,043,195
9087 :038,037,037,165,035,035,218
9093 :098,161,160,093,156,027,060
9099 :219,000,000,000,000,241,087
9105 :049,113,044,044,043,043,225
9111 :107,038,038,038,102,035,253
9117 :163,034,033,097,029,092,093
9123 :156,000,000,000,000,000,063
9129 :000,000,000,236,236,235,108
9135 :230,038,038,038,038,038,083
9141 :036,100,034,033,033,029,190
9147 :029,000,000,000,000,000,216
9153 :242,050,242,000,243,000,202
9159 :000,000,230,230,038,038,223
9165 :230,036,228,226,225,222,092
9171 :222,030,000,000,000,000,207
9177 :000,242,050,050,000,000,047
9183 :243,000,000,000,000,230,184
```

```
9189 :230,000,000,000,000,000,203
9195 :000,000,222,222,000,000,167
9201 :000,000,242,242,242,242,185
9207 :000,000,243,000,000,000,234
9213 :000,230,000,000,000,000,227
9219 :000,000,000,222,030,222,221
9225 :000,000,242,000,000,000,251
9231 :242,000,000,000,000,000,001
9237 :000,000,000,000,000,000,021
9243 :000,000,000,000,000,222,249
9249 :222,000,000,255,032,082,112
9255 :069,071,073,079,078,083,236
9261 :000,078,069,087,032,069,124
9267 :078,071,076,000,085,082,187
9273 :066,065,078,032,078,069,189
9279 :000,072,069,065,082,084,179
9285 :076,078,068,000,071,032,138
9291 :080,076,065,073,078,083,018
9297 :000,065,084,076,065,078,193
9303 :084,073,067,000,083,079,217
9309 :085,084,072,069,082,078,051
9315 :000,065,082,075,076,065,206
9321 :084,069,088,000,077,079,246
9327 :085,078,084,065,073,078,062
9333 :000,080,065,067,073,070,216
9339 :073,067,032,000,001,007,047
9345 :010,015,022,031,035,039,025
9351 :047,052,052,220,243,243,224
9357 :047,063,220,078,228,077,086
9363 :077,228,227,206,092,062,015
9369 :243,092,227,242,227,243,147
9375 :099,063,047,228,063,069,216
9381 :100,190,069,070,100,077,003
9387 :077,070,070,212,078,212,122
9393 :243,243,197,212,228,243,007
9399 :197,235,242,228,242,047,094
9405 :033,059,033,246,104,126,022
9411 :202,189,036,097,089,189,229
9417 :220,052,118,122,081,038,064
9423 :003,171,186,238,254,204,239
9429 :171,002,080,070,070,235,073
9435 :000,145,069,001,001,134,057
9441 :087,203,097,096,119,223,026
9447 :066,234,170,246,245,234,146
9453 :158,124,254,111,247,057,164
9459 :067,159,211,066,027,095,100
9465 :029,104,164,179,005,065,027
9471 :052,233,044,056,004,136,012
9477 :017,210,066,230,063,169,248
9483 :175,077,154,057,061,092,115
9489 :140,062,047,120,216,037,127
9495 :059,005,145,213,145,243,065
9501 :187,242,011,230,131,193,255
9507 :000,000,068,082,085,159,173
9513 :028,152,000,000,013,013,247
```

MLX

(Article on page 132.)

```
10 REM LINES CHANGED FROM MLX VERSION 2.0
   0 ARE 750,765,770 AND 860        :rem 50
100 PRINT"{CLR}{6}";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1         :rem 67
101 POKE 788,52:REM DISABLE RUN/STOP
                                    :rem 119
110 PRINT"{RVS}{39 SPACES}";        :rem 176
120 PRINT"{RVS}{14 SPACES}{RIGHT}{OFF}
    {*}£{RVS}{RIGHT} {RIGHT}{2 SPACES}
    {*}{OFF}{*}£{RVS}£{RVS}
    {14 SPACES}";                   :rem 250
130 PRINT"{RVS}{14 SPACES}{RIGHT} {G}
    {RIGHT} {2 RIGHT} {OFF}£{RVS}£{*}
    {OFF}{*}{RVS}{14 SPACES}";      :rem 35
140 PRINT"{RVS}{41 SPACES}"         :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} MACHINE LANG
    UAGE EDITOR VERSION 2.01{5 DOWN}"
                                    :rem 237
210 PRINT"{5}{2 UP}STARTING ADDRESS?
    {8 SPACES}{9 LEFT}";            :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F)
                                    :rem 166
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210           :rem 235
225 PRINT:PRINT:PRINT               :rem 180
230 PRINT"{5}{2 UP}ENDING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=
    CHR$(31+119*F)                  :rem 20
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230           :rem 183
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230
                                    :rem 176
260 PRINT:PRINT:PRINT               :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                    :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
    2),5);":";                      :rem 33
315 FORJ=ATO6                       :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                    :rem 228
390 IFN=-211THEN 710                :rem 62
400 IFN=-204THEN 790                :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                  :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                    :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                    :rem 238
420 IF N<>-196 THEN 480             :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                       :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0                               :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";      :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";       :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                    :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                       :rem 50
480 IFN<0 THEN PRINT:GOTO310        :rem 168
490 A(J)=N:NEXTJ                    :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT     :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
    );                              :rem 94
511 IFN=-1THENA=6:GOTO315           :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
                                    :rem 122
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
    NTER":PRINT:GOSUB1000:GOTO310   :rem 176
530 GOSUB2000                       :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0               :rem 227
550 AD=AD+6:IF AD<E THEN 310        :rem 212
560 GOTO 710                        :rem 108
570 N=0:Z=0                         :rem 88
580 PRINT"{£}";                     :rem 81
581 GETA$:IFA$=""THEN581            :rem 95
```

```
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*
    (A$="J")-5*(A$="K")-6*(A$="L"):rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
    ):IFA$="H"THENA$="Ø"          :rem 134
584 IFAV>ØTHENA$=CHR$(48+AV)      :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN67Ø                 :rem 229
59Ø IFA>128THENN=-A:RETURN        :rem 137
6ØØ IFA<>2Ø THEN 63Ø              :rem 1Ø
61Ø GOSUB69Ø:IFI=1ANDT=44THENN=-1:PRINT"
    {OFF}{LEFT} {LEFT}";:GOTO69Ø  :rem 62
62Ø GOTO57Ø                       :rem 1Ø9
63Ø IFA<48ORA>57THEN58Ø           :rem 1Ø5
64Ø PRINTA$;:N=N*1Ø+A-48          :rem 1Ø6
65Ø IFN>255 THEN A=2Ø:GOSUB1ØØØ:GOTO6ØØ
                                  :rem 229
66Ø Z=Z+1:IFZ<3THEN58Ø            :rem 71
67Ø IFZ=ØTHENGOSUB1ØØØ:GOTO57Ø    :rem 114
68Ø PRINT",";:RETURN              :rem 24Ø
69Ø S%=PEEK(2Ø9)+256*PEEK(21Ø)+PEEK(211)
                                  :rem 149
691 FORI=1TO3:T=PEEK(S%-I)        :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                  :rem 2Ø5
7ØØ PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                  :rem 7
71Ø PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                  :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
     ALONE TO CANCEL SAVE){DOWN}":rem 1Ø6
72Ø F$="":INPUT"{DOWN} FILENAME";F$:IFF$=
    ""THENPRINT:PRINT:GOTO31Ø     :rem 71
73Ø PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"       :rem 228
74Ø GETA$:IFA$<>"T"ANDA$<>"D"THEN74Ø
                                  :rem 36
75Ø DV=1-7*(A$="D"):IFDV=8THENF$="Ø:"+F$:
    OPEN15,8,15,"S"+F$:CLOSE15    :rem 212
76Ø T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256              :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE78Ø,LEN(
    T$):SYS65469                  :rem 1Ø9
763 POKE78Ø,1:POKE781,DV:POKE782,1:SYS654
    66                            :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
    *256:POKE78Ø,253              :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
    2)*256:SYS65496               :rem 235
77Ø IF(PEEK(783)AND1)OR(191ANDST)THEN78Ø
                                  :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":GOTO31Ø
                                  :rem 113
78Ø PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN72Ø      :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO72Ø          :rem 1Ø3
79Ø PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                  :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
     ALONE TO CANCEL LOAD)"       :rem 82
8ØØ F$="":INPUT"{2 DOWN} FILENAME";F$:IFF
    $=""THENPRINT:GOTO31Ø         :rem 144
81Ø PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"       :rem 227
82Ø GETA$:IFA$<>"T"ANDA$<>"D"THEN82Ø
                                  :rem 34
83Ø DV=1-7*(A$="D"):IFDV=8THENF$="Ø:"+F$
                                  :rem 157
84Ø T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256              :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE78Ø,LEN(
    T$):SYS65469                  :rem 1Ø7
845 POKE78Ø,1:POKE781,DV:POKE782,1:SYS654
    66                            :rem 7Ø
85Ø POKE78Ø,Ø:SYS65493            :rem 11
86Ø IF(PEEK(783)AND1)OR(191ANDST)THEN87Ø
                                  :rem 111
865 PRINT"{DOWN}DONE.":GOTO31Ø    :rem 96
87Ø PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN8ØØ
                                  :rem 172
88Ø OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO8ØØ          :rem 1Ø2
1ØØØ REM BUZZER                   :rem 135
1ØØ1 POKE54296,15:POKE54277,45:POKE54278,
    165                           :rem 2Ø7
1ØØ2 POKE54276,33:POKE 54273,6:POKE54272,
    5                             :rem 42
1ØØ3 FORT=1TO2ØØ:NEXT:POKE54276,32:POKE54
    273,Ø:POKE54272,Ø:RETURN      :rem 2Ø2
2ØØØ REM BELL SOUND               :rem 78
2ØØ1 POKE54296,15:POKE54277,Ø:POKE54278,2
    47                            :rem 152
2ØØ2 POKE 54276,17:POKE54273,4Ø:POKE54272
    ,Ø                            :rem 86
2ØØ3 FORT=1TO1ØØ:NEXT:POKE54276,16:RETURN
                                  :rem 57
3ØØØ PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
    GOTO1ØØØ                      :rem 89
```

Balloon Blitz

(Article on page 56.)

Program 1:

Balloon Blitz—64 Version

```
1 PRINT"{CLR}":GOSUB1ØØ           :rem 224
2 IF BO=ØTHEN 5Ø                  :rem 83
3 A=PEEK(563202):IFA=123THENX=X-S :rem 160
4 IFA=119THENX=X+S:IFA=123THENX=X-S
                                  :rem 239
5 IFX>255THENX=22                 :rem 3Ø
6 IFX<22THENX=255                 :rem 29
7 POKEV+12,X                      :rem 2Ø6
8 Z=Z-1.5*D:IFZ<24THENZ=255       :rem 127
9 IFZ>255THENZ=24                 :rem 4Ø
1Ø POKEV+6,Z                      :rem 2Ø5
11 CM=CM+H:POKEV+4,CM:POKEV+14,CM+25:IFCM
    =2Ø4THENH=H*-1                :rem 89
12 IFCM=24THENH=H*-1              :rem 169
13 GETA$:T=VAL(A$):IFT>ØANDT<1ØTHENPRINT"
    {HOME}{DOWN}";TAB(13)"SPEED ";T;"
    {HOME}":S=T/1.75              :rem 53
14 IFA>117THENGOTO3               :rem 167
15 FORI=MTOM+24:POKEI,Ø:NEXT:POKEM+24,15:
    POKEM+5,31:POKEM+6,Ø:POKEM+4,35
                                  :rem 195
16 BO=BO-1:PRINT"{HOME}{5 DOWN}":PRINTTAB
    (34)BO;"{LEFT}{2 SPACES}"     :rem 154
17 D=INT(RND(1)*DS+Ø):D=D-SUB:POKEV+3Ø,Ø:
    POKEV+3,9Ø:POKEV+2,X          :rem 184
18 FORI=1TO12                     :rem 13
19 A=PEEK(563202):IFA=119ORA=1Ø3THENX=X+S
                                  :rem 141
2Ø IFA=123ORA=1Ø7THENX=X-S         :rem 88
21 IFX>255THENX=22                :rem 76
22 IFX<22THENX=255                :rem 75
23 POKEV+12,X                     :rem 252
24 Z=Z-1.5*D:IFZ>255THENZ=24      :rem 175
25 IFZ<24THENZ=255                :rem 84
26 POKEV+6,Z                      :rem 212
27 CM=CM+H:POKEV+4,CM:POKEV+14,CM+25:IFCM
    =24THENH=H*-1                 :rem 48
```

```
28 IFCM=204THENH=H*-1                    :rem 224
29 POKEV,205:POKEV+3,90+I*10:COL=PEEK(V+3
   0):IFCOL=142ORCOL=206THENGOTO33
                                         :rem 176
30 POKEM,240-I*5:POKEM+1,240-I*5:NEXT
                                         :rem 152
31 POKEM,0:POKEM+1,0:POKEM+4,0:D=INT(RND(
   1)*DS+0):D=D-SUB                      :rem 170
32 POKEV+2,0:MI=MI+1:PRINTTAB(34)"
   {9 DOWN}";MI:GOTO2                    :rem 75
33 POKE2043,203:POKEV+2,0:POKEV+28,57:POK
   EM,255:POKEM+1,4:POKEM+2,0            :rem 226
34 POKEM+3,8:POKEM+5,63:POKEM+6,90:POKEM+
   4,129                                 :rem 217
35 FORI=1TO3:FORJ=1TO30:POKEV+42,J/10:NEX
   T:NEXT                                :rem 3
36 POKEV+28,49:POKEV+30,0:POKE2043,201:PO
   KEV+42,12:FORI=MTOM+24:POKEI,0:NEXT
                                         :rem 145
37 D=INT(RND(1)*DS+0):D=D-SUB            :rem 191
38 HI=HI+1:PRINTTAB(34)"{4 DOWN}";HI:GOTO
   2                                     :rem 117
50 AV=HI/20                              :rem 91
51 IFAV>=.95THEN AV$="**GENERAL**":GOTO57
                                         :rem 147
52 IFAV>=.85THENAV$="{2 SPACES}MAJOR
   {2 SPACES}":GOTO57                    :rem 102
53 IFAV>=.75THENAV$=" CAPTAIN ":GOTO57
                                         :rem 237
54 IFAV>=.50THENAV$=" SERGEANT ":GOTO57
                                         :rem 64
55 IFAV>=.25THENAV$=" CORPORAL ":GOTO57
                                         :rem 76
56 AV$=" PRIVATE "                       :rem 194
57 PRINT"{HOME}{23 DOWN}{10 SPACES}GAME O
   VER{14 SPACES}"                       :rem 45
58 FORI=1TO300:NEXTI                     :rem 3
59 FORJ=1TO20:PRINT"{UP}{5 SPACES}RANK:
   {RVS}";AV$:FORI=1TO50:NEXTI           :rem 190
60 PRINT"{UP}{5 SPACES}RANK: {OFF}";AV$:F
   ORI=1TO50:NEXTI:NEXTJ                 :rem 27
61 PRINT"{UP}{RVS}{6 SPACES}ANOTHER GAME(
   Y/N)?{8 SPACES}"                      :rem 108
62 GETAN$:IFAN$<>"Y"ANDAN$<>"N"THEN62
                                         :rem 183
63 IFAN$="N"THENPOKE254,0:SYS254         :rem 163
64 PRINT"{UP}{RVS}{2 SPACES}WHICH LEVEL O
   F PLAY 1-6                            :rem 6
65 GETLE$:IFLE$<"1"ORLE$>"6"THEN65
                                         :rem 215
66 LE=VAL(LE$):IF LE=1 THEN DS=2:SUB=-1
                                         :rem 223
67 IF LE=2 THEN DS=3:SUB=0               :rem 195
68 IF LE=3 THEN DS=3:SUB=1               :rem 198
69 IF LE=4 THEN DS=4:SUB=2               :rem 202
70 IF LE=5 THEN DS=6:SUB=3               :rem 198
71 IF LE=6 THEN DS=7:SUB=4               :rem 202
72 D=INT(RND(1)*DS+0):D=D-SUB            :rem 190
73 X=150:Y=130:S=1:Z=0:BO=20:HI=0:MI=0:T=
   1                                     :rem 208
74 FORI=MTOM+24:POKEI,0:NEXT             :rem 45
75 PRINT"{UP}{RVS}{GRN}{13 SPACES}LEVEL "
   ;LE;"{LEFT}{11 SPACES}{OFF}{WHT}"
                                         :rem 245
76 PRINT"{HOME}":PRINTTAB(13)"SPEED
   {2 SPACES}1"                          :rem 138
77 PRINT"{HOME}{5 DOWN}":PRINTTAB(35)"20"
                                         :rem 164
78 PRINTTAB(34)"{4 DOWN}{3 SPACES}"
                                         :rem 19
79 PRINTTAB(34)"{4 DOWN}{3 SPACES}"
                                         :rem 20
80 GOTO2                                 :rem 211
100 V=53248:M=54272:CM=24:CN=25:H=.5:X=15
    0:Y=130:S=1:Z=0:BO=20:HI=0:MI=0
                                         :rem 148
101 POKE2040,204:POKE2041,202:POKE2042,20
    5:POKE2043,201:POKE2044,204          :rem 19
102 POKE2045,204:POKE2046,200:POKE2047,20
    5:POKEV,205:POKEV+1,200              :rem 150
103 POKEV+4,24:POKEV+5,63:POKEV+6,0:POKEV
    +7,205:POKEV+8,150:POKEV+9,170
                                         :rem 233
104 POKEV+10,60:POKEV+11,183:POKEV+12,150
    :POKEV+13,80:POKEV+14,49:POKEV+15,57
                                         :rem 13
105 POKEV+23,149:POKEV+28,49:POKEV+29,133
    :POKEV+37,13:POKEV+38,5:POKEV+39,9
                                         :rem 207
106 POKEV+40,14:POKEV+41,15:POKEV+42,12:P
    OKEV+43,9:POKEV+44,9:POKEV+45,14
                                         :rem 77
107 POKEV+46,15:POKE53280,2              :rem 13
108 FORI=0TO24:POKEM+I,0:NEXT            :rem 62
109 POKEM+5,85:POKEM+6,85:POKEM+12,85:POK
    EM+13,85:POKE 53280,2                :rem 67
110 PRINT"{CLR}{DOWN}{WHT}";TAB(6)"
    {2 SPACES}{RVS}WELCOME TO BALLOON BLI
    TZ":PRINT"{2 DOWN}"                  :rem 99
111 POKEM+24,15:POKEM+4,33:POKEM+11,17
                                         :rem 176
112 FORFF=1TO6:READH1,L1,H2,L2:POKEM+1,H1
    :POKEM,L1:POKEM+8,H2:POKEM+7,L2
                                         :rem 173
113 PRINTTAB(9)"{RVS}{UP}{RIGHT} ** YOUR
    {2 SPACES}MISSION ** "               :rem 160
114 IFH1=50THENFORT=1TO200:NEXT          :rem 198
115 FORT=1TO100:NEXT                     :rem 237
116 PRINTTAB(9)"{OFF}{UP}{RIGHT} ** YOUR
    {2 SPACES}MISSION ** ":NEXTFF        :rem 40
117 FORI=MTOM+24:POKEI,0:NEXT            :rem 91
118 PRINT"{DOWN}{7}{2 SPACES}<PATROL FIEL
    DS IN A HOT AIR BALLOON>"            :rem 191
119 PRINT"{2 SPACES}<DESTROY ENEMY TANKS
    {SPACE}IN YOUR SECTOR>"              :rem 197
120 PRINT"{DOWN}{8 SPACES}PUT JOYSTICK IN
     PORT 2"                             :rem 236
121 PRINT"{8 SPACES}PUSH FIRE BUTTON TO D
    ROP BOMBS"                           :rem 242
122 PRINT"{DOWN} CONTROL YOUR SPEED WITH
    {SPACE}THE NUMBER KEYS":PRINT TAB(16)
    "1=SLOWEST"                          :rem 179
123 PRINTTAB(16)"9=FASTEST":PRINTTAB(12)"
    {3 DOWN}ONE MOMENT PLEASE"           :rem 187
124 FOR I=0 TO 5:FOR N=0 TO 62:READ Q:POK
    E 12800+(I*64)+N,Q:NEXT:NEXT         :rem 63
125 FORI=1TO10:GETA$:NEXT                :rem 50
126 PRINT"{UP}{5 SPACES}ENTER LEVEL OF PL
    AY 1 THROUGH 6 "                     :rem 68
127 PRINTTAB(16)"{DOWN}1=EASIEST":PRINTTA
    B(16)"6=HARDEST"                     :rem 159
128 GET LE$:IF LE$<"1"OR LE$>"6"THEN128
                                         :rem 55
129 A$="1":LE=VAL(LE$)                   :rem 164
130 IF LE=1 THEN DS=2:SUB=-1             :rem 22
131 IF LE=2 THEN DS=3:SUB=0              :rem 235
132 IF LE=3 THEN DS=3:SUB=1              :rem 238
133 IF LE=4 THEN DS=4:SUB=2              :rem 242
134 IF LE=5 THEN DS=6:SUB=3              :rem 247
135 IF LE=6 THEN DS=7:SUB=4              :rem 251
136 D=INT(RND(1)*DS+0):D=D-SUB           :rem 239
137 PRINT"{CLR}": POKE 53280,0:POKE 53281
    ,0                                   :rem 142
138 FOR I=1 TO 17                        :rem 69
139 PRINT"{RVS}{BLU}{32 SPACES}{OFF}"
                                         :rem 49
```

```
140 NEXT                                :rem 212
141 PRINTTAB(21)"{UP}{5 RIGHT}£§*³"
                                        :rem 156
142 PRINT"{RVS}{BLU}{32 SPACES}{OFF}"
                                        :rem 43
143 PRINTTAB(21)"{UP}{2 RIGHT}£§*³£
    {2 SPACES}§*³£§*³"                  :rem 87
144 PRINT"{RVS}{BLU}{32 SPACES}{OFF}"
                                        :rem 45
145 PRINTTAB(20)"{UP}£§*³£{2 SPACES}M
    {2 SPACES}N{2 SPACES}§*³"           :rem 49
146 PRINT"{RVS}{BLU}{32 SPACES}{OFF}"
                                        :rem 47
147 PRINTTAB(19)"{UP}£ N{4 SPACES}MN
    {4 SPACES}"                         :rem 162
148 PRINT"{RVS}§6³{32 SPACES}{OFF}"
                                        :rem 171
149 PRINTTAB(19)"{UP}{RIGHT}£{6 SPACES}
    §*³"                                :rem 55
150 PRINT"{RVS}§6³{32 SPACES}{OFF}"
                                        :rem 164
151 PRINT"{RVS}§6³{32 SPACES}{OFF}"
                                        :rem 165
152 PRINT"{RVS}{GRN}§32 U³{OFF}"        :rem 43
153 PRINT"{RVS}{GRN}{12 SPACES}LEVEL:";LE
    ;"{LEFT}{12 SPACES}{OFF}"           :rem 198
154 FOR I=0 TO 31                       :rem 62
155 POKE 1984+I,160                     :rem 215
156 POKE 56256+I,5                      :rem 168
157 NEXT                                :rem 220
158 PRINT "{WHT}{HOME}{3 DOWN}";TAB(33)"B
    OMBS"                               :rem 58
159 PRINT TAB(33)"LEFT"                 :rem 41
160 PRINT TAB(33)"{4 DOWN}HITS"         :rem 114
161 PRINT TAB(33)"{4 DOWN}MISS{HOME}"
                                        :rem 138
162 PRINT TAB(13)"SPEED{2 SPACES}";A$;"
    {HOME}"                             :rem 153
163 PRINT"{HOME}{6 DOWN}";TAB(35)"20"
                                        :rem 85
164 POKEV+21,255                        :rem 118
165 RETURN                              :rem 124
200 DATA25,30,18,209,33,135,25,30,42,62,3
    1,165,50,60,37,162,42,62,31,165,50,60
                                        :rem 20
201 DATA37,162                          :rem 220
202 DATA 0,127,0,1,255,192,3,255,224,3,25
    5,224                               :rem 195
203 DATA 7,255,240,7,255,240,7,255,240,3,
    255,224                             :rem 53
204 DATA 3,255,224,3,255,224,2,255,160,1,
    127,64                              :rem 252
205 DATA 1,62,64,0,156,128,0,156,128,0,73
    ,0,0,73,0                           :rem 126
206 DATA 0,62,0,0,62,0,0,62,0,0,28,0
                                        :rem 184
207 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:rem 171
208 DATA 127,224,63,255,224,63,255,224,0,
    127,224,0,63,192,63,255,252         :rem 21
209 DATA 127,255,254,255,255,255,255,255,
    255,127,255,254,63,255,252          :rem 5
210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0     :rem 73
211 DATA 0,0,0,0,0,0,0,0,0,0,0,102,0,0,126,
    0,0,60,0,0,60,0,0,60,0,0,60,0,0,60
                                        :rem 200
212 DATA 0,0,24,0                       :rem 41
213 DATA 10,170,160,42,170,168,170,170,17
    0,170,170,170,170,170,170,170,170,170
                                        :rem 233
214 DATA170,170,170,170,170,170,170,170,1
    70,42,255,42,29,170,56,0,170,0
                                        :rem 195
215 DATA 0,170,0,0,170,0,0,170,0,0,170,0,
    0,170,0,0,170,0,97,170,134,170,170
                                        :rem 26
216 DATA 170,42,170,168                 :rem 108
217 DATA 0,60,0,0,255,0,3,255,192,15,253,
    240,15,63,240,63,255,252,63,255,252
                                        :rem 136
218 DATA 63,247,220,63,255,252,63,247,220
    ,61,255,252,63,255,252,15,255,240
                                        :rem 69
219 DATA 15,247,240,3,255,192,0,40,0,0,40
    ,0,0,40,0,0,40,0,0,40,0,0,40,0:rem 70
220 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,255
                                        :rem 117
221 DATA 224,31,255,248,63,255,252,127,25
    5,254,255,255,255,255,255,255,255
                                        :rem 88
222 DATA 255,255,127,255,254,63,255,252,3
    1,255,248,7,255,224                 :rem 156
```

Program 2:

Balloon Blitz—VIC Version

Note: See instructions in article before typing in.

```
80 POKE36879,27                         :rem 61
90 POKE945,0                            :rem 150
100 GOTO 10000                          :rem 187
103 DD=37154:PA=37137:PB=37152:BO=20:HI=0
    :V=36878:S4=36877:S1=36876          :rem 10
104 CL=5:S$=" 1":FL=0:T=1:S=9           :rem 49
105 XX=PEEK(945):IFPEEK(8098+XX)<47THENPO
    KE8098+CO+XX,7:POKE8098+XX,37       :rem 26
106 IFPEEK(8099+XX)<47THENPOKE8099+CO+XX,
    7:POKE8099+XX,37                    :rem 223
107 FORI=7724TO7767:POKEI,32:NEXTI:XX=PEE
    K(945)                              :rem 155
108 POKE944,0:POKE945,15:POKE946,15:rem 0
109 DO$="{CYN}{HOME}{22 DOWN}{RVS}{GRN}"
                                        :rem 42
110 POKE4176,15:POKE4177,15:POKE4304,1:PO
    KE4305,5                            :rem 26
120 IFBO=0ANDFL=0THEN800                :rem 185
130 SYS828:SYS4190                      :rem 60
140 CL=CL+1:IFCL=6THENCL=1:SYS4096:rem 79
150 POKE37139,0:POKEDD,127              :rem 223
160 RI=-((PEEK(PB)AND128)=0):POKEDD,255
                                        :rem 218
170 P=PEEK(PA):LE=((PAND16)=0)          :rem 205
175 IF-((PAND32)=0)=1ANDFL=0THENFL=1:GOSU
    B500                                :rem 174
180 X=1+LE+RI:POKE4304,X                :rem 153
190 IFINT(RND(1)*60)<LI*2THENPOKE944,INT(
    RND(1)*3)                           :rem 178
195 GETA$                               :rem 228
200 T=VAL(A$):IFT>0ANDT<10THENS$=STR$(T):
    S=(10-T)                            :rem 173
205 PRINT"{HOME}{RVS}{YEL}{2 SPACES}SPEED
     ="S$" LEVEL= "LI$"{2 SPACES}";
                                        :rem 129
206 BO$=STR$(BO):HI$=STR$(HI)+" "       :rem 35
207 PRINTDO$"{3 SPACES}BOMBS";BO$;" HITS"
    ;HI$;                               :rem 102
290 IFFL=1THENGOSUB520                  :rem 112
295 IFIN=1THENIN=0:GOTO107              :rem 114
300 FORI=0TO(S-1)*15:NEXTI:GOTO120:rem 45
500 SYS4336:BO=BO-1                     :rem 91
502 REM IFINT(RND(1)*20)<LI*2THENPOKE944,
    INT(RND(1)*3)                       :rem 143
506 POKEV,PEEK(V)OR10:POKES1,240        :rem 201
510 RETURN                              :rem 118
520 SYS4384:POKES1,240-Q                :rem 150
530 Q=Q+1:IFQ<>14THENRETURN             :rem 219
535 XX=PEEK(4608):Q=0:FL=0              :rem 3
```

```
540 IFPEEK(8098+XX)=33ORPEEK(8098+XX)=34T
    HEN600                          :rem 41
542 POKES1,0:POKEV,PEEK(V)OR12:POKES4,130
                                    :rem 21
545 IFPEEK(8098+XX)>45ANDPEEK(8098+XX)<50
    THENFORI=1TO100:NEXTI:GOSUB970:POKES4
    ,0:RETURN                       :rem 31
547 POKE8076+CO+XX,7:POKE8076+XX,37
                                    :rem 78
550 POKE8098+XX+CO,10:POKE8098+XX,62
                                    :rem 120
560 FORI=1TO100:NEXTI               :rem 47
570 POKE8098+CO+XX,7:POKE8098+XX,37
                                    :rem 82
580 GOSUB970                        :rem 189
590 POKES4,0:RETURN                 :rem 202
600 XX=PEEK(945):POKES1,0:POKEV,PEEK(V)OR
    15:POKES4,140                   :rem 83
601 IFPEEK(8098+XX)<>48THENPOKE8076+CO+XX
    ,9:POKE8098+CO+XX,9             :rem 165
603 IFPEEK(8099+XX)<>47THENPOKE8077+CO+XX
    ,9:POKE8099+CO+XX,9             :rem 169
610 IFPEEK(8098+XX)<>48THENPOKE8076+XX,53
    :POKE8098+XX,60                 :rem 135
620 IFPEEK(8099+XX)<>47THENPOKE8077+XX,54
    :POKE8099+XX,61                 :rem 140
630 FORI=1TO250:NEXTI               :rem 51
640 HIT=HIT+1:IFPEEK(8098+XX)<>48THENPOKE
    8076+XX+CO,7:POKE8098+CO+XX,7   :rem 65
650 IFPEEK(8099+XX)<>47THENPOKE8077+XX+CO
    ,7:POKE8099+XX+CO,7             :rem 167
660 IFPEEK(8098+XX)<>48THENPOKE8076+XX,37
    :POKE8098+XX,37                 :rem 146
665 IFPEEK(8099+XX)<>47THENPOKE8077+XX,37
    :POKE8099+XX,37                 :rem 153
690 FORX=14TO0STEP-2:FORI=1TO10:POKEV,PEE
    K(V)AND240ORX:NEXTI:NEXTX       :rem 118
695 POKES4,0                        :rem 182
700 IN=1:RETURN                     :rem 182
800 AV=HI/20                        :rem 142
810 IFAV>.89THENAV$="GENERAL":GOTO850
                                    :rem 21
820 IFAV>.74THENAV$="MAJOR":GOTO850
                                    :rem 139
830 IFAV>.50THENAV$="SERGEANT":GOTO850
                                    :rem 102
840 IFAV>.30THENAV$="CORPORAL":GOTO850
                                    :rem 110
845 AV$="PRIVATE"                   :rem 248
850 GOSUB950:PRINTDO$"{5 SPACES}GAME OVER
    ";                              :rem 14
855 FORI=1TO1500:NEXTI              :rem 107
860 GOSUB950:PRINTDO$" YOUR RANK IS ";AV$
    ;                               :rem 198
870 FORI=1TO2000:NEXTI              :rem 100
880 GOSUB950:PRINTDO$" PLAY AGAIN YNN";
                                    :rem 198
885 GETA$:IFA$="N"THENPOKE950,0:SYS950
                                    :rem 22
890 IF A$<>"Y"THEN885               :rem 128
900 GOSUB 950:PRINTDO$" WHICH LEVEL 1C6";
                                    :rem 201
905 GETA$:LI=VAL(A$):IFLI<1ORLI>6THEN905
                                    :rem 202
910 LI$=A$:GOTO103                  :rem 252
950 PRINTDO$"{21 SPACES}";:RETURN:rem 123
970 FORX=14TO0STEP-2:FORI=1TO6:POKEV,PEEK
    (V)AND240ORX:NEXTI:NEXTX:RETURN
                                    :rem 102
8052 POKE8064+CO,8:POKE8064,63:POKE8065+C
     O,8:POKE8065,64:POKE8066+CO,8:POKE80
     66+CO,65                       :rem 223
10000 PRINT"{CLR}{BLU}{DOWN}{RVS}
    {4 RIGHT}BALLOON BLITZ"         :rem 151
10010 PRINT"{DOWN}{2 SPACES}** YOUR MISSI
    ON **"                          :rem 237
10020 S1=36876:S2=36875:V=36878:POKEV,15
                                    :rem 249
10030 FORX=1TO6:READP1,P2:POKES1,P1:POKES
    2,P2                            :rem 53
10040 FORT=1TO150:NEXT T            :rem 164
10050 IF P1=235THENFORT=1TO200:NEXT T
                                    :rem 183
10060 NEXTX:POKES1,0:POKES2,0:POKEV,0
                                    :rem 60
10070 POKEV,0                       :rem 217
10080 PRINT"{DOWN} PATROL FIELDS IN A
    {4 SPACES}HOT AIR BALLOON"      :rem 10
10090 PRINT"{DOWN} DESTROY ENEMY TANKS
    {3 SPACES}IN YOUR SECTOR"       :rem 187
10100 PRINT"{DOWN} USE JOYSTICK TO
    {7 SPACES}CONTROL BALLOON{DOWN}"
                                    :rem 13
10110 PRINT" PUSH THE FIREBUTTON
    {3 SPACES}TO DROP A BOMB"       :rem 32
10120 PRINT"{DOWN} CONTROL YOUR SPEED
    {4 SPACES}WITH THE NUMBER KEYS"
                                    :rem 217
10130 PRINT" 1=SLOWEST, 9=FASTEST":rem 33
10140 PRINT"{DOWN}{2 SPACES}ONE MOMENT PL
    EASE";                          :rem 127
10150 PRINTCHR$(142)                :rem 109
10160 CS=5120:FORI=CSTOCS+2047:POKEI,PEEK
    (I+32768-CS):NEXT               :rem 15
10170 FORI=CS+264TOCS+527:READJ:POKEI,J:N
    EXTI                            :rem 79
10175 GOSUB11000:GOSUB12000:GOSUB13000:GO
    SUB14000                        :rem 122
10180 IFPEEK(13983)=102THEN10200    :rem 249
10190 CLR:POKE36869,253             :rem 27
10200 PRINT"{CLR}{2 DOWN}{2 SPACES}ENTER
    {SPACE}LEVEL OF PLAY"           :rem 58
10201 PRINT"{DOWN}{RVS} 1=EASIEST
    {2 SPACES}6=HARDEST "           :rem 226
10210 GET LI$::IFLI$<"1"ORLI$>"6"THEN1021
    0                               :rem 47
10215 LI=VAL(LI$)                   :rem 184
10230 PRINT"{CLR}":CO=30720:POKE36879,111
                                    :rem 152
10235 FORI=8054TO8119:POKEI,37:POKEI+CO,7
    :NEXT                           :rem 66
10240 FORI=8120TO8141:POKEI+CO,3:POKEI,16
    0:NEXTI                         :rem 165
10250 FORI=8142TO8185:POKEI,160:POKEI+CO,
    5:NEXTI                         :rem 180
10260 FORI=38673TO38773:POKEI,0:NEXTI
                                    :rem 240
10270 POKE7954,233:POKE7955,223:POKE7975,
    233:POKE7976,160                :rem 222
10280 POKE7977,160:POKE7978,223     :rem 214
10290 POKE7981,233:POKE7982,223:POKE7987,
    233                             :rem 218
10300 POKE7975,233:POKE7976,160:POKE7977,
    160:POKE7978,223:POKE7981,233
                                    :rem 231
10310 POKE7982,223:POKE7987,233:POKE7996,
    233:POKE7997,160:POKE7998,160
                                    :rem 242
10320 POKE 7999,160:POKE8000,160:POKE8001
    ,223:POKE8002,233:POKE8003,160:POKE
    8004,160                        :rem 153
10330 POKE8005,223:POKE8006,233:POKE8007,
    223:POKE8008,233:POKE8009,160
                                    :rem 166
10340 POKE8017,233:FORI=8018TO8023:POKEI,
    160:NEXT:POKE8024,205           :rem 139
```

```
10350 POKE8025,160:POKE8026,160:POKE8027,
      206:POKE8028,160:POKE8029,160
                                    :rem 177
10360 POKE8030,205:POKE8031,160     :rem 175
10370 POKE8038,233:FORI=8039TO8046:POKEI,
      160:NEXTI:POKE8047,206:POKE8048,205
                                    :rem 232
10380 POKE8049,160:POKE8050,160:POKE8051,
      160:POKE8052,160:POKE8053,205
                                    :rem 173
10382 POKE8068+CO,8:POKE8068,64:POKE8069+
      CO,0:POKE8069,160:POKE8070+CO,0:POK
      E8070,160                     :rem 161
10385 POKE8071+CO,8:POKE8071,65     :rem 241
10390 POKE8012+CO,5:POKE8012,35:POKE8034+
      CO,0:POKE8034,44              :rem 250
10400 POKE8015+CO,5:POKE8015,36:POKE8037+
      CO,5:POKE8037,45:POKE8059+CO,8:POKE
      8059,56                       :rem 59
10410 POKE8011+CO,5:POKE8011,36:POKE8033+
      CO,5:POKE8033,45:POKE8055+CO,8:POKE
      8055,56                       :rem 36
10420 POKE8078+CO,13:POKE8078,38:POKE8079
      +CO,13:POKE8079,39:POKE8100+CO,13:P
      OKE8100,47                    :rem 208
10430 POKE36878,48                  :rem 207
10440 POKE8101+CO,13:POKE8101,48:POKE8122
      +CO,8:POKE8122,57:POKE8123+CO,8:POK
      E8123,58                      :rem 87
10450 POKE8082+CO,13:POKE8082,38:POKE8083
      +CO,13:POKE8083,39:POKE8104+CO,13:P
      OKE8104,47                    :rem 199
10460 POKE8105+CO,13:POKE8105,48:POKE8126
      +CO,8:POKE8126,57:POKE8127+CO,8:POK
      E8127,58                      :rem 113
10470 GOTO103                       :rem 201
10480 DATA 215,215,225,225,231,231,235,23
      5,231,231,235,235             :rem 29
10490 DATA86,090,170,090,086,090,106,090
                                    :rem 235
10500 DATA149,165,169,169,165,169,170,169
                                    :rem 50
10510 DATA24,060,094,251,255,239,126,060
                                    :rem 223
10520 DATA0,000,000,000,000,048,048,120
                                    :rem 129
10530 DATA255,255,255,255,255,255,255,255
                                    :rem 39
10540 DATA85,086,090,090,106,106,170,154
                                    :rem 229
10550 DATA85,149,165,101,169,169,170,170
                                    :rem 242
10560 DATA85,85,85,85,85,85,85,85:rem 178
10570 DATA24,060,126,255,255,255,255,189
                                    :rem 242
10580 DATA24,061,127,255,255,255,255,127
                                    :rem 237
10590 DATA128,192,238,255,255,255,254,252
                                    :rem 41
10600 DATA24,024,024,024,024,024,060,126
                                    :rem 200
10610 DATA120,252,252,252,252,120,120,48
                                    :rem 207
10620 DATA0,000,000,000,000,000,000,000
                                    :rem 103
10630 DATA170,170,170,106,102,090,090,090
                                     :rem 5
10640 DATA170,170,154,169,169,165,165,165
                                    :rem 39
10650 DATA0,000,000,000,000,000,000,000
                                    :rem 106
10660 DATA153,090,060,060,060,024,000,000
                                    :rem 245
10670 DATA63,127,255,255,255,119,035,001
                                    :rem 231
10680 DATA254,255,255,255,254,252,248,224
                                    :rem 38
10690 DATA86,090,106,106,170,170,170,170
                                    :rem 227
10700 DATA149,165,169,169,170,170,170,170
                                    :rem 32
10710 DATA254,056,124,254,254,124,056,016
                                    :rem 19
10720 DATA101,101,101,101,101,101,101,101
                                    :rem 216
10730 DATA250,250,250,250,250,250,250,250
                                     :rem 1
10740 DATA191,191,191,191,191,191,191,191
                                    :rem 34
10750 DATA0,000,000,192,252,204,051,051
                                    :rem 146
10760 DATA106,090,086,086,090,090,090,090
                                    :rem 28
10770 DATA165,165,149,149,165,165,165,165
                                    :rem 49
10780 DATA150,085,105,085,150,170,170,170
                                    :rem 18
10790 DATA183,147,153,153,153,147,135,255
                                    :rem 35
10800 DATA 86,86,90,90,106,106,170,170
                                    :rem 83
10810 DATA 149,149,165,165,169,169,170,17
      0                             :rem 44
11000 I=828                         :rem 26
11002 READ A:IF A=256 THEN RETURN   :rem 68
11004 POKE I,A:I=I+1:GOTO 11002     :rem 167
11006 DATA 174,177,3,138,168,173,176
                                    :rem 252
11008 DATA 3,201,1,240,19,48,2      :rem 177
11010 DATA 16,2,202,202,232,224,21
                                    :rem 104
11012 DATA 208,2,162,0,224,255,208
                                    :rem 122
11014 DATA 2,162,20,185,162,151,41
                                    :rem 121
11016 DATA 15,201,8,208,10,169,7    :rem 30
11018 DATA 153,162,151,169,37,153,162
                                    :rem 34
11020 DATA 31,185,163,151,41,15,201
                                    :rem 167
11022 DATA 8,208,10,169,7,153,163   :rem 85
11024 DATA 151,169,37,153,163,31,189
                                    :rem 244
11026 DATA 162,151,41,15,201,7,208
                                    :rem 123
11028 DATA 10,169,8,157,162,151,169
                                    :rem 196
11030 DATA 33,157,162,31,189,163,151
                                    :rem 234
11032 DATA 41,15,201,7,208,10,169   :rem 73
11034 DATA 8,157,163,151,169,34,157
                                    :rem 197
11036 DATA 163,31,142,177,3,96,256:rem 144
12000 I=4096                        :rem 76
12002 READ A:IF A=256 THEN RETURN   :rem 69
12004 POKE I,A:I=I+1:GOTO 12002     :rem 169
12006 DATA 174,80,16,172,81,16,232
                                    :rem 135
12008 DATA 224,21,208,2,162,0,224   :rem 69
12010 DATA 255,208,2,162,20,169,32
                                    :rem 126
12012 DATA 153,44,30,153,45,30,153
                                    :rem 122
12014 DATA 66,30,153,67,30,169,1    :rem 35
12016 DATA 157,44,150,157,45,150,157
                                    :rem 240
```

```
12018 DATA 66,150,157,67,150,169,42
                                   :rem 198
12020 DATA 157,44,30,169,43,157,45
                                   :rem 140
12022 DATA 30,169,51,157,66,30,169
                                   :rem 142
12024 DATA 52,157,67,30,142,80,16 :rem 85
12026 DATA 142,81,16,96,256        :rem 56
13000 I=4190                       :rem 72
13002 READ A:IF A=256 THEN RETURN :rem 70
13004 POKE I,A:I=I+1:GOTO 13002  :rem 171
13006 DATA 174,209,16,138,168,173,208
                                   :rem 42
13008 DATA 16,201,1,240,8,48,3    :rem 182
13010 DATA 16,3,96,202,202,232,224
                                   :rem 119
13012 DATA 22,208,2,162,0,224,255 :rem 70
13014 DATA 208,2,162,21,169,32,185
                                   :rem 134
13016 DATA 44,30,201,42,234,240,9 :rem 73
13018 DATA 201,43,240,5,169,32,153:rem 130
13020 DATA 44,30,185,66,30,201,51 :rem 73
13022 DATA 240,9,201,52,240,5,169 :rem 78
13024 DATA 32,153,66,30,189,44,30 :rem 86
13026 DATA 201,42,240,14,201,43,240
                                   :rem 162
13028 DATA 10,169,2,157,44,150,169
                                   :rem 142
13030 DATA 41,157,44,30,189,66,30 :rem 87
13032 DATA 201,51,240,14,201,52,240
                                   :rem 159
13034 DATA 10,169,4,157,66,150,169
                                   :rem 145
13036 DATA 50,157,66,30,142,209,16
                                   :rem 137
13038 DATA 96,256                  :rem 81
14000 I=4336                       :rem 75
14002 READ A:IF A=256 THEN RETURN :rem 71
14004 POKE I,A:I=I+1:GOTO 14002  :rem 173
14006 DATA 169,88,133,251,133,253,169
                                   :rem 45
14008 DATA 30,133,252,169,150,133,254
                                   :rem 26
14010 DATA 173,209,16,141,0,18,169
                                   :rem 132
14012 DATA 32,141,1,18,169,6,141  :rem 27
14014 DATA 2,18,172,0,18,145,253  :rem 30
14016 DATA 169,55,145,251,96,234,234
                                   :rem 251
14018 DATA 234,234,234,234,234,234,172
                                   :rem 80
14020 DATA 0,18,173,2,18,145,253  :rem 28
14022 DATA 173,1,18,145,251,165,251
                                   :rem 182
14024 DATA 24,105,22,133,251,165,252
                                   :rem 225
14026 DATA 105,0,133,252,165,253,24
                                   :rem 177
14028 DATA 105,22,133,253,165,254,105
                                   :rem 25
14030 DATA 0,133,254,177,251,201,38
                                   :rem 177
14032 DATA 240,31,201,39,240,27,201
                                   :rem 167
14034 DATA 47,240,23,201,48,240,19
                                   :rem 131
14036 DATA 177,253,141,2,18,177,251
                                   :rem 193
14038 DATA 141,1,18,169,0,145,253 :rem 86
14040 DATA 169,55,145,251,96,177,253
                                   :rem 255
14042 DATA 141,2,18,177,251,141,1 :rem 76
14044 DATA 18,96,256               :rem 228
```

Error Trapping

(Article on page 113.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

Program 1:

Error Trapping—VIC Version

```
200 REM ERROR TRAP ROUTINE         :rem 93
210 PRINT:PRINT"ERROR DETECTED"   :rem 247
220 REM GET LINE AND MESSAGE       :rem 88
230 GOSUB4000                     :rem 217
240 PRINTER$:PRINT"ERROR AT LINE"LN
                                  :rem 202
250 END                           :rem 110
3000 DATA 169,58,141,0,3,169,196,141,1,3,
     165,58,141,123               :rem 142
3010 DATA 3,201,255,240,39,165,57,141,122
     ,3,142,121,3,160             :rem 217
3020 DATA 0,185,124,3,153,0,2,240,3,200,2
     08,245,162,255               :rem 115
3030 DATA 160,1,76,134,196,169,60,141,0,3
     ,169,3,141,1                  :rem 34
3040 DATA 3,96,108,0,3             :rem 47
3050 RESTORE:FORAD=828TO888:READVA:POKEAD
     ,VA:NEXTAD                    :rem 88
3060 T$="GOTO"+STR$(ET)+CHR$(0)   :rem 218
3070 FORAD=1TOLEN(T$):POKE891+AD,ASC(MID$
     (T$,AD,1)):NEXTAD             :rem 209
3080 SYS875:RETURN                 :rem 136
4000 EN=PEEK(889):EA=PEEK(49958+2*EN)+256
     *PEEK(49959+2*EN):ER$=""      :rem 95
4010 ER$=ER$+CHR$(PEEK(EA)AND127):IFPEEK(
     EA)<128THENEA=EA+1:GOTO4010    :rem 6
4020 LN=PEEK(890)+256*PEEK(891):RETURN
                                  :rem 216
```

Program 2:

Error Trapping—64 Version

```
200 REM ERROR TRAP ROUTINE         :rem 93
210 REM GET LINE AND MESSAGE       :rem 87
220 GOSUB4000                     :rem 216
230 PRINT:PRINT"ERROR DETECTED"   :rem 249
240 PRINTER$:PRINT"ERROR AT LINE"LN
                                  :rem 202
250 END                           :rem 110
3000 DATA 169,139,141,0,3,169,227,141,1,3
     ,165,58,141,123              :rem 185
3010 DATA 3,201,255,240,39,165,57,141,122
     ,3,142,121,3,160             :rem 217
3020 DATA 0,185,124,3,153,0,2,240,3,200,2
     08,245,162,255               :rem 115
3030 DATA 160,1,76,134,164,169,60,141,0,3
     ,169,3,141,1                  :rem 29
3040 DATA 3,96,108,0,3             :rem 47
3050 RESTORE:FORAD=828TO888:READVA:POKEAD
     ,VA:NEXTAD                    :rem 88
3060 T$="GOTO"+STR$(ET)+CHR$(0)   :rem 218
3070 FORAD=1TOLEN(T$):POKE891+AD,ASC(MID$
     (T$,AD,1)):NEXTAD             :rem 209
3080 SYS875:RETURN                 :rem 136
4000 EN=PEEK(889):IFEN>127THENEND:rem 237
```

```
4005 EA=PEEK(41766+2*EN)+256*PEEK(41767+2
     *EN):ER$=""                          :rem 37
4010 ER$=ER$+CHR$(PEEK(EA)AND127):IFPEEK(
     EA)<128THENEA=EA+1:GOTO4010          :rem 6
4020 LN=PEEK(890)+256*PEEK(891):RETURN
                                          :rem 216
```

Program 3:
Error Trapping—Demonstration Program

```
10 REM ON ERROR GOTO 200                  :rem 55
20 ET=200:GOSUB3050                       :rem 76
30 REM GENERATE AN ERROR                  :rem 171
40 FORI=10TO0STEP-1:PRINT"1/"I"="1/I:NEXT
   I                                      :rem 63
50 END                                    :rem 60
```

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

COMPUTE!'s Gazette Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE!'s Gazette
P.O. Box 961
Farmingdale, NY 11737

or call the Toll Free number listed below.

Change of Address. Please allow us 6-8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your Gazette subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one-year (12-month) U.S. subscription to *COMPUTE!'s Gazette* is $24 (2 years, $45; 3 years, $65. For subscription rates outside the U.S., see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of *COMPUTE!'s Gazette*, if you experience late delivery, or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!'s Gazette
800-334-0868
In NC 919-275-9809

PRODUCT MART

HOME COMPUTER DESK PLANS
PROTECT YOUR INVESTMENT!

Designed by home computer user. All the room you need for computer monitor, printer, peripherals, etc. Shelves for software, everything at your fingertips. Fits COMMODORE, ATARI, APPLE I & II, IBM-PC, TRS 80. Bottom shelf slotted for printer paper plus storage. 28" deep x 51¾" high x 71¾" length. **Quality Plans, Instructions.**

PLANS - $10.00

CARPENTER'S CREATIVE DESIGNS

P.O.Box 122 / Desert Center, CA 92239

PROFESSIONAL FOOTBALL
A Strategy Game For
Vic +16K & Commodore 64

Challenge the Commodore to a game of real football! *Over* 130 *play combinations, full feature scoreboard and total statistical summary.* Computer selects its plays based on time, score, down & distance, but watch out for surprises! Try to stop the explosive 2 minute offense, or try winning a cliff-hanger with time running out. Block punts, fieldgoals, and force turnovers, or be victimized by the aggressive defense. No two games are ever alike!

$16.95 for cassette and playbook.
$19.95 for disk version.

CMS SOFTWARE
Box 4876
Topeka, KS 66604
(913)-267-5864

VISA MasterCard

Visa, MC include Card #
Exp. Date and Signature

CONVERSE WITH YOUR COMPUTER

AT LAST! A FULL IMPLEMENTATION of the original ELIZA program is now available to run on your Commodore 64!

Created at MIT in 1966, ELIZA has become the world's most celebrated artificial intelligence demonstration program. ELIZA is a non-directive psychotherapist who analyzes each statement as you type it in and then responds with her own comment or question – and her remarks are often amazingly appropriate!

Designed to run on a large mainframe, ELIZA has never before been available to personal computer users except in greatly stripped down versions lacking the sophistication which made the original program so fascinating.

Now, our new Commodore 64 version possessing the **FULL** power and range of expression of the original is being offered at the introductory price of only $25. And if you want to find out how she does it (or teach her to do more) we will include the complete **SOURCE PROGRAM** for only $20 additional.

Order your copy of ELIZA today and you'll never again wonder how to respond when you hear someone say, "Okay, let's see what this computer of yours can actually do!"

ELIZA IS AVAILABLE IN THE FOLLOWING FORMATS:
(Please specify Disk or Cassette)

1. Protected Version $25
(Protected Version can be run but not listed or modified)
2. Un-protected Commodore 64 BASIC Source Version $45
(Source Version can be listed and modified as well as run)
Both versions include a six page user manual.

Please add $2.00 shipping and handling to all orders
(California residents please add 6% sales tax)

ARTIFICIAL INTELLIGENCE RESEARCH GROUP
921 North La Jolla Avenue, Dept. G
Los Angeles, CA 90046
(213) 656-7368 (213) 654-2214
MC, VISA and checks accepted

VISA MasterCard

Numeric key pad for Commodore VIC-20 and 64

With full cursor control and special function keys. No software interaction.

$59.95 Retail

Bear Technologies INC.
3417 Roger Chaffee Blvd.
Grand Rapids, MI 49508
(616) 245-5061
VISA and M/C accepted
Dealer inquiries invited

FLIGHT SIMULATOR GAMES

NEW COCKPIT 64
For the Commodore 64
- 100% Machine Language
- Windshield View
- 7 Airports

Tape **$29.95** Disk **$31.95**

	Tape	Disk
Runway (C64•16K VIC 20•Adam)	$24.95	$26.95
Sky Pilot (8K VIC 20•Atari)	$17.95	19.95
Flight Sim. II (C64•Apple•IBM)	N/A	$47.95
IFR (Commodore 64 only)	$29.95	$29.95
Air Traffic Controller (Apple)	N/A	$19.95

COD ORDER PHONE
WE SHIP WITHIN 48 HOURS
(312) 577-5154

SUSIE SOFTWARE
874A E. N.W. Highway
Mt. Prospect, IL 60056

KEEP THE DUST OFF & PUT THE ELEGANCE ON WITH GENUINE LEATHER DUST COVERS

Enjoy the look of soft elegance, along with durability that only real leather can offer. Don't settle for less than the best. Order singly or as a matched set, custom fitted to your Commodore computers.

ORDER TODAY

QTY.		AMT.
_____	Computer Cover/14.95	_____
_____	1541 Disk Cover/13.95	_____
_____	Dataset Cover/9.95	_____
	TOTAL $	_____

□ Check or Money Order enclosed.
□ Visa; □ Mastercard Exp. Date _____
Card No. _____
Signature _____
SHIP TO:
Name _____
Address _____
City _____ St./Zip _____

S & S ENTERPRISES
P. O. BOX 111
HOT SPRINGS, S.D. 57747
Dealer Inquiries Invited

SIGN OF THE SPHINX

Something new from **Werewolf Software** — A world of enigmatic artifacts, disturbing visions, peculiar people. From an abandoned subway station to the Eternal Black Mass and beyond, you discover traces of forgotten knowledge which may lead you to the Crimson Altar and its final secret.

A morbid, suprising adventure. Sign of the Sphinx is disk-based to use memory more efficiently. If you appreciate detail, if you have a taste for the bizarre, then you should investigate the activity at the Sign of the Sphinx.

Text adventure on disk for the Commodore 64. $15 plus $1 shipping. Calif. residents add tax.

WEREWOLF SOFTWARE
109 Minna Street
Suite 353
San Francisco, CA 94105

VISIT LAS VEGAS

on your COMMODORE 64

ADVANCED MICROWARE Introduces:

-------- 64 CASINO PAC ----------
Includes:
* SLOT MACHINE * POKER
* BLACKJACK * KENO

All for only $39
Each program is a graphic simulation of their respective Vegas video gaming machine. These are colorful and exciting games you will enjoy over and over. Practice your "system" or just play for fun.

Also available: 64TOUR
This is a Tour and demo of the many features of your 64 including a version of BASIC with new graphics commands. Only $15

Send Check or Money Order to:
ADVANCED MICROWARE
P.O. BOX 6143 Dept. CG-4G
SANTA ANA, CA. 92706
Specify DISK or TAPE, CA. orders-add 6%
Dealer Inquiries Invited (714)554-6470

For your **VIC-20:**

- **GORTEK** and the **MICROCHIPS** (tape)
- **DEMON ATTACK** (cart.)

$9 ea. or $17 for both

MC/VISA orders call toll-free

1-800-282-0333

or send certified check or M.O. for same day shipping. (Personal check 2 wks.) Add $2 per order postage.

ABBY'S
611 Cypress Dr.
Fairborn, OH 45324

We carry full Commodore line. Write for our catalog.

PROTECT YOUR EXPENSIVE EQUIPMENT FROM DUST, LIQUIDS WITH A CROWN PROTECTIVE

COVER

- CUSTOM MADE TO FIT
- HEAVY 32 oz VINYL
- ANTI-STATIC
- SOFT LINED
- CHOICE of COLOR, TAN or BROWN

Covers for:

VIC20/C-64	7.00
C-1541 D/DRIVE	8.00
C-1525 PRINTER	10.00
DATASETTE (New)	5.00
DATASETTE (Old)	5.00
GEMINI 10/10X PRINTER	13.00
GEMINI 15/15X PRINTER	16.00
EPSON MX80 PRINTER	11.00
EPSON MX100 PRINTER	14.00
APPLE IIe KEYBOARD	7.00

Order by stating name and model of equipment for cover desired. Choice of color: TAN or BROWN. Enclose check or M.O. + 1.50 shipping. Calif. Res. include 6.5% State Tax.

COVERS NOT NAMED ABOVE WILL BE FABRICATED TO YOUR SPECS. SEND YOUR REQUIREMENTS FOR LOW PRICE QUOTES.

CROWN CUSTOM COVERS
9606 SHELLYFIELD RD.,
DOWNEY, CA 90240

VIC 20/C64

"THE REUNION"©

(Brings the Commodore family together again)

"THE REUNION" simultaneously interfaces your *VIC 20 and *C/64 (including *Datasette, modem, etc) to your Commodore disk drive and/or printer **providing 2 computer systems.** Use either instantly.

*Simply select "VIC 20" or "C-64" on "THE REUNION", and your disk drive and/or printer is instantly connected to the Selected Computer.

SAVE and LOAD VIC 20 and C/64 programs on same disk. Ends switching disks, cables, and wear. Plug-in installation, 1 year warranty.

Send $29.95 (U.S. $, Check or M.O. plus $2.00 shipping. Canadian: $4.00). FL res. add 5% tax to:
HyTech
P.O. Box 466
Bay Pines, FL 33504
*Reg. T.M. of Commodore Bus. Mach. ©HyTech

ATTENTION C-64 DISK USERS

ORGANIZE NOW!

WITH THE

MASTER-DIRECTORY SUPPORT SYSTEM

With MDSS you can organize your disk files onto 1 master disk. Maintain sorted master-directories of your files categorized by business, education, recreation or any other category you choose. Print single or multiple copies of master-directory listings, disk jacket indexes or individual disk labels. MDSS can locate your "lost" disk files too!

MDSS is fully menu driven and very user-friendly. Includes an easy to follow instruction manual.

Requires C-64 and 1540 or 1541 disk drive. Printer optional.

Send check or money order for $16.95 to:

SUNSHINE SOFTWARE
P.O. BOX 831
DEARBORN, MICHIGAN 48120
MICHIGAN RESIDENTS ADD 4% SALES TAX

---

ditto

COPY DISKS AUTOMATICALLY

$39.95

- Copies 99% of currently available Commodore 64 disks.
- Supports 1 or 2 1541 drives.
- Time required 25 minutes.
- Easy to use menu driven.
- Available now—updates included.
- Ditto provided on unprotected disk.

ORDERS
800-762-5645

CARDINAL SOFTWARE
13646 Jefferson Davis Highway
Woodbridge, VA 22191

FAMILY TREE

A NEW COMPUTER SOFTWARE PACKAGE TO HELP THE AMATEUR AND PROFESSIONAL GENEALOGIST USE THE COMMODORE 64 OR VIC AS A DYNAMIC SYSTEM TO CONTROL DATA ON THE FAMILY TREE.

FEATURES:

- 664 NAMES PER DATA DISK
- FULLY INDEXED
- EASY EDITING AND UPDATING
- SEARCH FUNCTIONS
- PRODUCES FAMILY GROUP SHEETS
- PRODUCES PEDIGREE CHARTS
- OUTPUT TO SCREEN OR PRINTER
- COMPLETE MANUAL

BY HELPING YOU TO ORGANIZE YOUR FAMILY TREE IT WILL AID YOU IN DETERMINING THE AREAS OF YOUR RESEARCH

PRICE $39.95 US — $49.95 CANADIAN (MICHIGAN & ONTARIO RESIDENCE ADD TAX)

GENEALOGY SOFTWARE
PHONE 519-344-3990

P.O. BOX 1151
PORT HURON, MICHIGAN 48061

1046 PARKWOOD AVE
SARNIA, ONTARIO N7V 3T9

C-64™ & VIC-20™

SUPER TYPEWRITER
The mini word processor you've wanted . . .

FEATURES:

- ★ Changeable line width up to 80 characters
- ★ Automatic margin setting
- ★ Automatically centers each additional copy
- ★ Upper and Lower Letters
- ★ No more broken words with use of automatic carriage return
- ★ Edit Text

All Poorhaus Programs user accessible for learning or adding personal touch. Simple to use. Load and follow instructions within programs.

Super Typewriter	**$24.95**
Home Inventory	**12.95**
Check Register	**19.95**
Black Jack	**9.95**
Loan Analyzer	**9.95**

Some VIC-20 Programs may need memory expansion.

POORHAUS SOFTWARE
P.O. Box 10782, Yakima, WA 98909
(509) 966-8461
SPECIFY TAPE OR DISK
MC, VISA, AND CHECKS ACCEPTED

COMMODORE
-USER WRITTEN SOFTWARE-
Supporting all COMMODORE computers
Written by users, for users
★ GAMES ★ UTILITIES ★ EDUCATIONAL ★

VIC 20™
Vic 20 collections #1, 2, 3, 4, 5, 6.
over 70 programs per collection - Tape/Disk - $10.00
Vic 20 collections #7, 8, 9
over 50 programs per collection - Tape/Disk - $10.00

COMMODORE 64™
64 collections #1, 2, 3, 4, 5, 6, 7, 8
over 25 programs per collection - Tape/Disk - $10.00

PET® / CBM®
22 collections - Tape/Disk - $10.00

DINSET™: Reset Switch
Works on Vic 20 or Commodore 64 - $5.00

SERIAL CABLES
10Ft.—$10.00 15Ft.—$15.00

LOC-LITE™
Operation Status Indicator Assembled & Tested
$20.00

All prices include shipping and handling
CHECK, MONEY ORDERS.
VISA and MASTERCARD accepted.
For A Free Catalog Write:

Public Domain, Inc.
5025 S. Rangeline Rd., W. Milton, OH 45383
10:00 a.m. - 5:00 p.m. EST – Mon. thru Fri.
(513) 698-5638 or (513) 339-1725

ELECTRONICS

Circuit Design and Analysis

TEST CIRCUITS BEFORE YOU BUILD THEM!

ANALYSIS PACKAGE INCLUDES:
Two powerful programs to analyze the frequency and phase response of almost any circuit configuration of Resistors, Capacitors, Inductors, Op-Amps, FET's and NPN Transistors.

CIRCUIT ANALYSIS PACKAGE...$29.95

DESIGN PACKAGE INCLUDES:
Programs to design Active Filters, Passive Filters and Attenuators. plus...Ohms-Law, Resonance, Wire Gauge, Standard Resistor Value Solutions & more...

CIRCUIT DESIGN PACKAGE...$19.95

EACH ON DISK FOR THE COMMODORE 64 FROM:

Nth DIGIT SOLUTIONS
3243 Arlington Avenue, No.195
Riverside, CA 92506

NAME
ADDRESS
CITY
STATE ZIP

• CALIFORNIA RESIDENTS ADD 6 PER CENT SALES TAX

ADVERTISERS INDEX

Reader Service Number/Advertiser	Page
102 Abby's	159
103 Academy Software	16
104 Access Software Incorporated	71
Advanced Microware	158
Altcom, Inc.	83
105 Artificial Intelligence Research Group	158
106 Avalon Hill Game Company	7
107 B & B Microlabs	97
108 Batteries Included	41
109 Batteries Included	63
110 Bear Technologies	158
111 Bible Research Systems	97
112 Big Bytes	112
Brantford Educational Services	109
113 Broadway Computer Corporation	127
114 Bytes & Pieces, Inc.	117
115 Cardco, Inc.	IBC
Cardinal Software	97
Cardinal Software	159
Carpenter's Creative Designs	158
Century Micro Products	114
Cheatsheet Products	157
The CHF Company	110
CMS Software	158
Commodore Computers	BC
116 CompuServe	IFC
ComputAbility	103
Computer Mail Order	75
Computer Place	125
117 Continental Software	33
118 Covox Inc.	157
Creative Software	4
Creative Software	49
Crown Custom Covers	159
119 C.S.M. Software	125
Datasoft, Inc.	2,3
Dazco	114
Dennison	57
120 Diversified Manufacturing	112
121 Dow Jones News/Retrieval	95
122 Eastern House	40
123 Eastern House	110
Educomp	105
Electronic Arts	13
Epyx	47
French Silk	123
Genealogy Software	159
Genesis Computer Corporation	121
124 GOSUB of Slidell, Inc.	115
Handic Software Inc.	61
HyTech	159
125 Indus-Tool	79
126 Innovative Organizers	81
127 Jameco Electronics	73
Jason-Ranheim	125
John Henry Software	123
Kiwisoft Programs	112
128 Limbic Systems Inc.	121
129 Lynn Computer Services	77
130 MFJ Enterprises Incorporated	44
131 Microlog Corporation	101
132 Micro Ware	105
133 Mirage Concepts, Inc.	15
134 MSD Systems, Inc.	26,27
135 M-W Dist. Inc.	123
Nth Digit Solutions	159
Official Olympic Guide to Los Angeles	39
136 Orange Micro Inc.	25
137 Orbyte Software	35
Parallel Systems	78
138 Parker Brothers	42
Parker Brothers	42
Parker Brothers	43
139 Parsec Research	79
140 PB Systems	112
141 PC Gallery	77
Penguin Products	109
Poorhaus Software	159
Practicorp International, Inc.	67
142 Precision Software, Inc.	1
Prentice-Hall	23
143 Professional Software, Inc.	9
Pro-Line Software	45
144 Protecto Enterprizes	85
Protecto Enterprizes	86,87
Protecto Enterprizes	88,89
145 Public Domain, Inc.	159
Quicksilva Inc.	59
146 Reader's Digest Services, Inc.	29
147 Reader's Digest Services, Inc.	31
Scarborough Systems	11
Scholastic Wizware	53
SM Software Inc.	111
SM Software Inc.	111
148 Smart Software Ltd.	79
Softlaw	99
Software Discounters of America	128
Software Plus	119
S & S Enterprises	158
Spinnaker	19
Spinnaker	21
149 Starpoint Software	22
150 subLOGIC Corporation	51
151 Such A Deal	128
Sunshine Software	159
Susie Software	158
152 Synapse	125
Syntonic Corp.	114
153 Timeworks, Inc.	37
3G Company, Inc.	119
154 Totl Software, Inc.	101
155 Tussey Mt. Software	97
Ultrabyte	83
Ultrabyte	123
Werewolf Software	158
York 10	93

COMPUTE!'s GAZETTE DISK	17
COMPUTE!'s GAZETTE Subscriber Services	157
COMPUTE!'s GAZETTE Subscription	65

"The Complete CARDCO Line"

. . . and still growing!

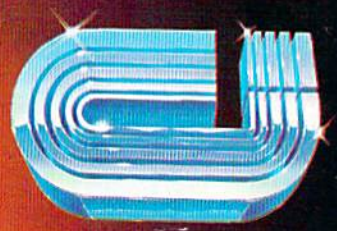
CARDCO provides "Commodore-ready" computer accessories that will enhance your utilization of Commodore-64 and VIC-20 Computers, increase their capability, and add to your enjoyment and skill. AND, they're available for use with other personal computers, too.

Designed with the user in mind, CARDCO offers fine accessories including Printer Interfaces with and without graphics, Expansion Interfaces, Memory Expansions, Cassette Interfaces, Numeric Keypads PLUS "NOW" Software for your VIC-20 and C-64. These programs include the "WRITE NOW" Word Processor, "MAIL NOW" Mailing List, PRINTER UTILITY PROGRAMS on Tape and on Disk, "SPELL NOW" Spell Checker, "GRAPH NOW" including "PAINT NOW", and "FILE NOW".

CARDCO has three new Letter Quality PRINTERS with your choice of drumhead design (8 1/2" carriage), Daisy Wheel Design (13 inch carriage) and Daisy Wheel Design (11 inch carriage). "Commodore-ready" . . . plus; with compatible input for PC, PC jr., TRS-80 and many more personal computers. CARDCO's NEW "DATA CASSETTE RECORDER/PLAYER" is also "Commodore-ready" and ready for instant shipment at prices that will amaze you.

CARDCO will constantly increase its line with unique and new products to enhance the enjoyment of computer owners.

Write for illustrated literature and prices or see CARDCO Computer Accessories and Software wherever Computers are sold.



cardco, inc.

300 S. Topeka Wichita, Kansas 67202 (316) 267-6525

"The world's largest manufacturer of Commodore accessories"

Commodore Software– The Best Game in Town.

...Take on the world, toughen up your trigger finger and fire away...

Commodore is the best computer value in town...at home, at school and at work... with our exciting, easy to use, inexpensive VIC 20 and C64 computers.

We're fast becoming the best game in town when it comes to entertainment for the whole family...and at affordable prices.

THE BEST ARCADE IN TOWN can be in your own home with our exciting, faithful reproductions of the best of Bally Midway arcade games. Our **Kickman,** (which just received a coveted "Electronic Games" award for an arcade translation) lets you steer the unicycle to catch the falling objects, as they fall quicker and quicker!!

Gorf, Lazarian, and Omega Race give you the best in classic space action against the one-eyed leviathon, the droids or the evil Empire.

In **The Wizard of Wor** you attempt to defeat the Wizard and the Warriors, fighting your way through to the end. With the new Commodore "MAGIC VOICE"... It talks back to you too!!

You commandeer the fleet at sea with our version of **Seawolf,** and become the master tactician as you battle "it out" with enemy fleet.

Clowns and **Blueprint** round out our arcade entertainment package to keep your fingers nimble and your mind in gear.

commodore COMPUTERS

First In Quality Software

See your local dealer now... He's got the

www.commodore.ca