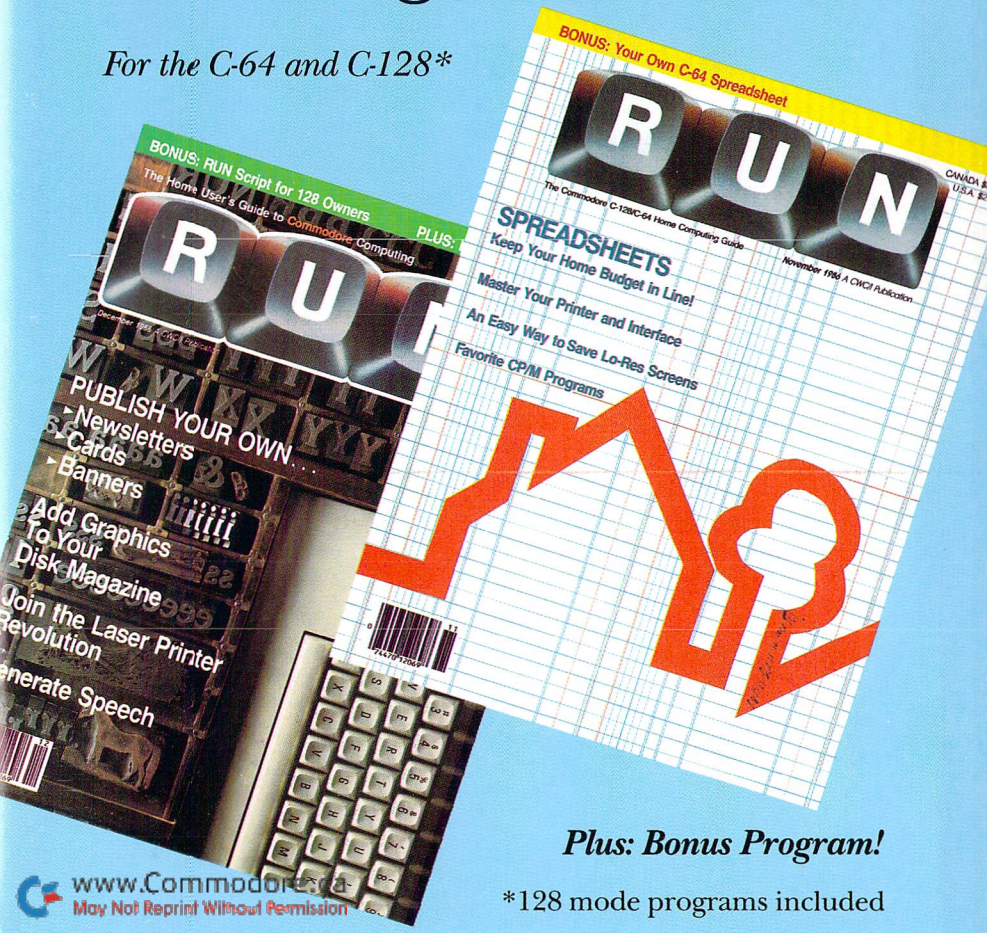
*November/December 1986 Edition*

# RUN

## *RUN* Programs on Disk

*For the C-64 and C-128**




***Plus: Bonus Program!***

*128 mode programs included

www.Commodore.ca
May Not Reprint Without Permission

# Introduction

## *November/December '86 ReRUN*

Here is our last ReRUN disk for 1986. And, what a year it has been! We've put together six regular ReRUN disks, which hold all of the programs from *RUN* magazine in 1986, and we're still recovering from our extra efforts on the Productivity Pak II disk.

I'd like to offer special thanks to our entire ReRUN staff (who, by the way, also put out 13 issues of *RUN* magazine this year) for an excellent job!

Now, on with the show. The November/December ReRUN disk holds 16 programs covering a wide range of applications and utilities.

At the top of the list is CalcAid 64, an electronic spreadsheet program that can help you in your home, at school or with your small business.

C-128 users will appreciate the newest version of our fine word processor, RUN Script. RUN Script 128 has many new features that were not present in the first release of RUN Script, for the C-64, back in March.

We've also included ML Perfect Typist 2.0, an update of our program for checking the accuracy of machine language code you type in from *RUN*.

Our disk-based newsletter program from the July issue has been applauded by user's groups and schools. Now there's an update to the reader module of that program, called Extra! Newsletter Graphics, that will let you incorporate multiple picture screens in the body of your newsletter.

Most of us need a screen print utility at one time or another. Put It on Paper will let you print out any text or keyboard graphics design you have on your C-64's screen.

We have three educational programs for you on this disk. Dashing Off the Dots is a Morse Code tutorial that provides training and practice in sending and receiving messages in that code. It comes in both C-128 and C-64 versions.

Math Square-Off is a challeng-



www.Commodore.ca
May Not Reprint Without Permission

ing mathematics puzzle for secondary school students and adults. It will really stretch those brain muscles as you try to fill in the missing links in six equations. Sum Fun is another mathematics game that provides some fun with addition for all ages.

Speaking of fun, those of you who like to draw with your computer will enjoy Micro Artist for the C-128 and the Plus/4. This is an easy-to-use, colorful graphics program.

Programmers who follow our Basically Speaking column will want to check out COMM AND-OR 64 and Error Channel for help in their programming efforts.

The Mega-Magic column premiered with our October issue. Each month it presents a one- or two-page hint or tip to make your computing just a bit easier. C-128 owners will want to use November's Sermaker utility for loading C-64 programs from the C-128 mode using the faster speed of the 1571 drive. C-64 programmers will benefit from December's Compare Basic, a utility for comparing two versions of the same Basic program to find the code differences.

That just about wraps it up—except for our bonus program. Peg Solitaire 64 is a board game package that will challenge you to solve 14 different games.

Well, have fun and a happy holiday season!

*Margaret Morabito*
*Technical Manager*
RUN *magazine*



www.Commodore.ca
May Not Reprint Without Permission

# How To Load

### *Loading from Menu*

To get started, C-64 users should type LOAD "MENU 64",8 and press the return key. When you get the Ready prompt, the menu is loaded and you should type RUN to see a list of the programs on your disk.

### *Loading from Keyboard*

If you do not wish to use the menu program, follow these instructions.

**C-64:**

To load a C-64 program written in Basic, type:

LOAD "DISK FILENAME",8

and then press the return key. The drive will whir while the screen prints LOADING and then READY, with a flashing cursor beneath. Type RUN and press the return key. The program will then start running.

To load a C-64 program written in machine language (ML), type:

LOAD "DISK FILENAME",8,1

**C-128:**

All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode.

All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs.

To load a C-128-mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type RUN.

### *Making Copies of ReRUN Disks*

Many of the programs on your ReRUN disk have routines that require you to have a separate disk onto which the program writes or saves subfiles. In order for you to use these programs, you will first have to make a copy of the original program onto another disk that has enough free space on it to hold these newly written subfiles.

If the program is written in Basic, it is simple to make a copy of the program. Just load the program into your computer following the procedures outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

If the program is written in ML, copying is not so simple. You cannot simply load and save an ML program. In this case, you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.



www.Commodore.ca
May Not Reprint Without Permission

**ReRUN Staff**

Technical Manager/Editor: *Margaret Morabito*
Technical Editor: *Tim Walsh*
Managing Editor/Production: *Swain Pratt*
Copy Editor: *Peg LePage*
Proofreader: *Harold Bjornsen*
Design and Layout: *Karla M. Whitney*
Typesetting: *Doreen Means, Beth Krommes, Ken Sutcliffe*
Special Products Director: *Jeff DeTray*
Special Products Manager: *Craig Pierce*
Special Products Assistants: *Debbie Bourgault, Robyn Johnson*
Direct Marketing Coordinator: *Cathy Carabello*



www.Commodore.ca
May Not Reprint Without Permission

# Directory

| Page | Article | Disk Filename | File | Type |
|---|---|---|---|---|
| | | MENU 128 | C128 | BASIC |
| | | MENU 64 | C64 | BASIC |
| 1 | CalcAid 64 | CALCAID 64 | C64 | BASIC |
| 6 | Put It on Paper | PUT IT ON PAPER | C64 | BASIC |
| 8 | Dashing Off the Dots | MORSE CODE 64 | C64 | BASIC |
| | | MORSE CODE 128 | C128 | BASIC |
| 11 | Math Square-Off | MATH SQUARE-OFF | C64 | BASIC |
| 14 | Micro Artist | MICRO ARTIST 128 | C128 | BASIC |
| | | MICRO ARTIST +4 | PLUS/4 | BASIC |
| 16 | Mega-Magic—November | SERMAKER PRG1 | C128 | BASIC |
| | | SERMAKER PRG2 | C128 | BASIC |
| | | SERMAKER PRG3 | C128 | BASIC |
| | | SERMAKER PRG4 | C128 | BASIC |
| 18 | Basically Speaking—November | COMM AND-OR 64 | C64 | BASIC |
| 20 | Extra! Newsletter Graphics | MAG READER V7.2 | C64 | BASIC |
| 23 | RUN Script 128 | RS128 | C128 | BASIC |
| | | MAKE C128 CHAR | C128 | BASIC |
| | | C128 CHAR SET | C128 | M/L |
| | | CREATE ML | C128 | BASIC |
| | | OB.RS NMI | C128 | M/L |
| | | OB.RS128 2.40 | C128 | M/L |
| 36 | ML Perfect Typist 2.0 | ML TYPIST 2.0 | C128 | BASIC |
| 39 | Sum Fun | SUM FUN 64 | C64 | BASIC |
| 42 | Mega-Magic—December | ERROR CHANNEL | C64 | BASIC |
| 43 | Basically Speaking—December | COMPARE BASIC | C64 | BASIC |
| £46 | Peg Solitaire 64 | PEG SOLITAIRE | C64 | BASIC |

NOTE: Do not load indented files as stand-alone programs!

£ Bonus program!



www.Commodore.ca
May Not Reprint Without Permission

# CalcAid 64

**By Trent Busch**

## RUN It Right

*Commodore 64; tape or disk; Printer optional*

This article will take you step by step through the features of CalcAid 64 and give you a sample spreadsheet to try for yourself.

Run the program. The flashing cursor at the top left of the screen represents the data entry line. Below that is a solid line running across the screen. This is a comment line that CalcAid 64 uses to display important messages and information. The numbers 0, 1 and 2 represent columns. The letters A through T are the rows.

If you don't like the screen colors, you can change them at any time. The F7 key changes the background color, and the F8 key steps through the border colors. To change the text color, simultaneously press the CTRL key with any number key from 1 to 8. Upon your next operation, the entire text will change color.

CalcAid 64 has 30 columns and 26 rows. Each column can display up to nine characters. Notice that only three columns are displayed on the screen. All 30 are there; you just cannot see them all at once. Imagine that you are looking through a window and can only see a portion of the overall picture.

The cursor keys allow you to move this window around the spreadsheet. Press the cursor-down key, and the spreadsheet will be quickly redrawn with rows B through U. Notice that the text is now the color that you chose. Experiment with the cursor keys until you can place the viewing window over all the columns and rows. Pressing the home key will return the window to A0.

### ENTERING INFORMATION

The intersection of a column and a row is called a cell. There are 780 cells that you can use, A0–Z29. There are three types of information that you can enter into a cell: text, numeric or formula.

In order to enter information into a cell, you need to follow a specific procedure. Type in the cell location, row first and column second, without putting in any



www.Commodore.ca
May Not Reprint Without Permission

spaces. Next, type a colon. This separates the cell location from the data. Now you can type in text or numeric data up to nine characters:

```
A0:BUDGET 84
C12:250
```

Text information can contain almost any character on the keyboard, but must not begin with a number or a plus or minus sign. Numeric information, however, *must* start with a number or a plus or minus sign.

After typing in your information, press the return key. If everything was typed in correctly, you should see the data in the proper cell. If you didn't enter your information properly, CalcAid 64 will display a Format error message on the display line. To rectify this, simply retype the entire line correctly. Text data will be left-justified, while numeric data will be right-justified.

To replace data, just retype the cell coordinates, a colon and the new data. To clear text or numeric data from a cell, simply type the cell coordinates followed by a colon and then press the return key. This procedure will not clear a formula, however.

Pressing SHFT CLR will clear the entire spreadsheet. For safety reasons, this is a two-step process. First, press SHFT CLR and then answer the question on the comment line. Press Y to clear the spreadsheet. Press N to exit the Clear mode.

## CALCULATIONS

While you now know how to create neat columns and rows, the real power of CalcAid 64 lies in its ability to do mathematical computations using the data in each cell. For example, you can add cell A0 to cell A1 and put the answer in cell A2. This is accomplished by putting the formula A0 + A1 into cell A2. Here is the proper format:

```
A2:{F1}A0+A1
```

The F1 key will result in a reverse-character F on the screen. This key is used to access the special features of CalcAid 64. If you forget to press F1 when entering a formula, the formula will be entered as text and displayed in the cell. Only the result of the computation, not the formula itself, should be displayed in a cell.

A special command allows you to view the formula in a particular cell:

```
A2:{F1}V
```

If a formula resides in cell A2, it will be printed on the comment line. The full value of the numeric data in cell A2 also will be printed. This is important, because each column is limited to nine characters. CalcAid 64 will fill the cell with asterisks if the numeric data is longer than nine characters. You will then need to



www.Commodore.ca
May Not Reprint Without Permission

use the View command to examine that cell.

Here are the formulas that this spreadsheet can utilize for computation:

addition: cell + cell or cell + constant
subtraction: cell – cell or cell – constant
multiplication: cell*cell or cell* constant
division: cell/cell or cell/constant
exponentiation: cell↑cell or cell↑ constant

CalcAid 64 cannot handle complex formulas. A more involved computation can be done by storing the intermediate answer in a spare cell. Extra characters after the second cell or constant will be ignored or show up as a Format error. When typing in a formula, leave out all spaces and be sure to enter the cell first and the constant second.

After you enter a formula and press the return key, you must press the left-arrow key. Wait for the calculation to be performed. During calculations, there will be a working message on the comment line. Calculations are done column by column from top to bottom. Column 1 will be completely done before column 2. This is an important point.

For example, let cell A0 = F9* G6. If cell F9 has a formula in it, the resulting answer will be figured after cell A0 is computed. To overcome this, you should press the left-arrow key twice. After all computations are complete, the spreadsheet will be redrawn with the results displayed in the proper cells. Attempts to divide by 0 will be noted in that cell, as will an overflow note if an exponentiation calculation is too large.

## COMMANDS AND SPECIAL FEATURES

CalcAid 64 has several other commands that are very useful. The following examples show the proper format for the commands. You may use any cells that you wish. The range must be in a straight row or column, with the first coordinate smaller than the second.

A1:{F1}SUMA2 – Z2

This command puts the sum of cells A2–Z2 into cell A1. Text data is ignored.

Z29:{F1}AVGB3 – B12

This command calculates the average of cells B3–B12 and puts the answer into cell Z29. Any text data is ignored.

C12:{F1}MIND0 – G0

This command looks for the minimum figure over a range of cells and puts the answer in cell C12. Text data is ignored.

F5:{F1}MAXZ0 – Z29



www.Commodore.ca
May Not Reprint Without Permission

This is similar to the MIN command except it returns the maximum value in a range. Again, text data is ignored.

Remember, you can use any cells that you wish, but they must be in a straight column or row.

Z29:{F1}SUMA0 – D29

This formula will not work because cells A0–D29 are in a diagonal.

Here are the rest of CalcAid 64's special features:

A0:{F1}T

This command makes row A and column 0 titles that are always displayed on the screen. This is helpful in remembering what each cell is supposed to be. As you move your display window around, you can always have a reference to numeric displays. You must always use cell A0 in this command.

A0:{F1}O

This command turns off the Title mode. The cell must always be A0.

C15:{F1}C

This command will clear an individual cell, including the formula, text and numeric data.

F25:{F1}J

This command jumps the display to a particular area of the spreadsheet. Sometimes this is faster than using the cursor keys to move the display window.

D3:{F1}COPD4 – D29

This command is used when you are entering lots of identical information. In this example, the contents of cell D3 will be copied into cells D4–D29. Only text or numeric information can be copied. Formulas must be typed individually. This works with rows or columns.

Press F2 and you will see a maximum precision display on the comment line. This command does not affect the accuracy of the calculations. It rounds the number only for display purposes. Use the View command to see the full value. Press a number from 0 to 6. Zero means integers and 6 means six decimal places. CalcAid 64 is automatically set up for two decimal places. This command is only for numbers that are computed by a formula. If you want two-place decimals on all the numbers, you must type them that way.

Press F4 and follow the screen directions to save the spreadsheet to tape or disk. Pick out a logical filename under which you can save the spreadsheet.

Press F3 and follow the screen directions to load the spreadsheet from tape or disk.

To print the spreadsheet on pa-



www.Commodore.ca
May Not Reprint Without Permission

| BUDGET 86 | RENT | CAR LOAN | GAS (+/-) |
|---|---|---|---|
| JAN | 585.99 | 393.75 | 50.00 |
| FEB | 585.99 | 393.75 | 50.00 |
| MAR | 585.99 | 393.75 | 50.00 |
| APR | 585.99 | 393.75 | 50.00 |
| MAY | 585.99 | 393.75 | 50.00 |
| JUN | 585.99 | 393.75 | 50.00 |
| JUL | 585.99 | 393.75 | 50.00 |
| AUG | 585.99 | 393.75 | 50.00 |
| SEP | 585.99 | 393.75 | 50.00 |
| OCT | 585.99 | 393.75 | 50.00 |
| NOV | 585.99 | 393.75 | 50.00 |
| DEC | 585.98 | 393.75 | 50.00 |
| --------- | --------- | --------- | --------- |
| TOTALS | 7031.87 | 4725 | 600 |
| MINIMUM | 585.98 | 393.75 | 50 |
| MAXIMUM | 585.99 | 393.75 | 50 |
| ********* | ********* | ********* | ********* |
| BUDGET | 585.99 | 393.75 | 50 |

per, press F5 and follow the screen directions. You can print the whole spreadsheet or any portion of it. You will need to know the top-left cell coordinates and the bottom-right cell coordinates of the area that you want printed out. If you specify more than seven columns, CalcAid 64 will automatically break the printout into sections for you.

To print the formulas on paper, press the F6 key and follow the screen directions.

The more you use this program, the more applications you will find for it.■



www.Commodore.ca
May Not Reprint Without Permission

# Put It on Paper

**By Mark Rasmussen**

## RUN It Right

*C-64; printer*

Screen Dump is a program I wrote to print out hard copies of lo-res and text screen displays when I'm using other programs. It's relocatable in memory, so it won't disturb most other Basic programs. Once you've loaded Screen Dump, it's there to use—all you have to do is press control-P. Just be aware that Screen Dump doesn't work with hi-res or bit-mapped screens.

The program loads into the top of Basic memory, where it's protected from most operations of other Basic programs. It uses only a small percentage of the program space in memory, so you'll still have plenty of room for other programs.

When you are preparing to use Screen Dump, make it the *last* utility you load into your C-64. Load the program and run it, writing down the number that appears on the screen after it runs. SYSing to that address will enable and disable the program. Next, load and run your Basic or machine language program. Now you can dump the screen to the printer by holding the control key down and typing P.

### THE PROGRAM

The Basic code in Screen Dump contains an instruction section and the data for the machine language routine. The unique part of the Basic code is the relocator that changes several locations in the machine language routine and lets you use Screen Dump with other utilities (like metaBasic) that also reside at the top of memory. You can run Screen Dump several times, and each time it'll be stored in a different memory location. When you press control-P, the locations will each print a screen, in consecutive order.

Screen Dump "hooks" into the memory area of the operating system that handles interrupt requests (IRQs)—60 or more of which occur every second. During an interrupt request, Screen



www.Commodore.ca
May Not Reprint Without Permission

Dump's machine language routine checks to see if both the control key and the P key are pressed at the same time. If they are, one or more screen dumps take place. If not, the interrupt handler goes on as usual.

The advantage of this technique is that the IRQs are generated continuously, whether you're in Basic Immediate or Programming mode or executing a machine language routine. That means Screen Dump will work any time until it's disabled or overwritten by another program. Some machine language programs and Basic programs will overwrite Screen Dump, so any time you plan to use Screen Dump, be sure to test it with the other program first.

**MAKING CHANGES**

Screen Dump performs the equivalent of OPEN 30,4,sa, where sa (secondary address) is omitted if you're in Uppercase/Graphics mode or is 7 if you're in Upper-/Lowercase mode. Accordingly, file number 30 should not be open if you are going to use Screen Dump.

You can change the numbers in the Open statement easily. The file number (30) is located in the variable BA in line 1400. You get the device number (4) by adding 70 to the value of BA, the secondary address for Uppercase/Graphics by adding 58, and the secondary address for Upper-/Lowercase by adding 65. You can change these locations either in the Data statements or in the lines that follow line 1400.

Screen Dump doesn't send linefeeds with carriage returns. If your printer dumps the screen on one line, replace the two 234s in line 1330 with 169 and 010, respectively. Then replace the three 234s in line 1340 with 032, 210 and 255.

My technique for hooking the interrupt handler is easy to use, and you can add other features to your C-64 with it as well. Having a relocatable utility is handy, because it enables you to store more than one wedge in memory at the same time.■



www.Commodore.ca
May Not Reprint Without Permission

# Dashing Off The Dots

**By Joe Novak**

## **RUN** It Right

*C-64; C-128*

Morse code has been around for a long time, an outgrowth of Samuel Morse's invention of the telegraph. It was instrumental in taming the West and in running the railroads, and is, of course, widely used by the armed forces, intelligence services and amateur radio operators around the world. A ham must, in addition to passing a test on electronic theory, be able to send and receive code to get a license.

If a ham license is your goal, Morse Code 64 or 128 will help, but even if you're not interested in amateur radio, you can still have fun learning the code. Keep in mind, however, that practicing on a computer will never replace working with another person.

Morse Code opens with a brief graphics display of code being sent from a naval ship. Then the Main menu appears. The first option is to view a code chart of the letters of the alphabet and numerals with their corresponding dot-and-dash ("dit" and "dah") codes. The other two options are for receiving and sending code.

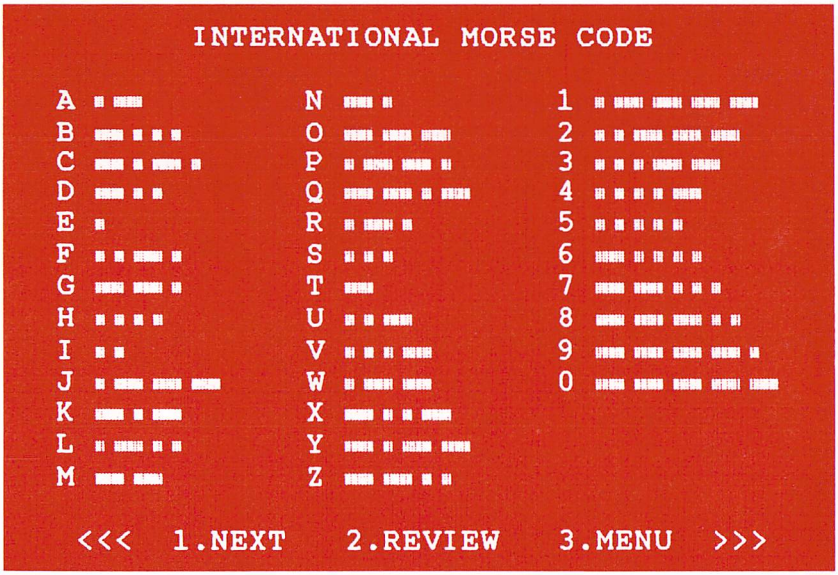
If you proceed to the chart screen, you'll find another menu at the bottom of the display. From it you can go to the next chart screen (for a list of punctuation marks and other miscellaneous codes), review the audio code for each character as it's displayed, or return to the Main menu.

### RECEIVING

If you choose the Receive menu, your choices are Random Characters, Library Message, Input Message or back to the Main menu. The first generates a random character (letter, numeral or punctuation mark) and simultaneously produces both audio



www.Commodore.ca
May Not Reprint Without Permission




and visual code for that character. The visual display is the ship again. When the prompt, a square arrangement of dots, appears, type in what you think the character was. The correct answer, with code, follows immediately. Then press any key to go to another character, or M to return to the Receive menu.

The library messages are a collection of sayings and other short blurbs that reside in a list of Data statements at the end of the program. When you choose the Library Message option on the Receive menu, one of this collection is output in code.

You can add to the collection or change the messages as you wish. Just remember that they can be no more than 255 characters long, a single X must always be the last Data statement, and if you use a comma, enclose the entire message in quotes. Morse Code will encode anything except parentheses and the short messages that appear at the end of the code chart (End of Message, for instance).

When using the Library Message option, you have to set the speed, in words per minute, at which you want the message sent. Fifteen is the maximum the computer can handle.

The next Receive menu choice



www.Commodore.ca
May Not Reprint Without Permission

is Input Message, an opportunity for you to type in an impromptu message that will be output in code for yourself or a friend. Don't use the comma and keep the message short—less than 255 characters. Once again, you must specify the sending speed. Press any key to start the code; then, when transmission is complete, the original uncoded message appears on the screen.

**SENDING**

Meanwhile, back at the Main menu, the final choice is Sending. You can use any key to send your code to the computer. If a quarter of a second elapses before you press the key again, the character you've typed appears on the screen. If you delay half a second, the program makes a word space. That may sound fast, but after a little practice it'll seem just the opposite.

Work on pacing—keeping your keystrokes uniform and consistent. When the program displays an asterisk, you've sent a meaningless sequence that can't be decoded. Since receiving Morse code is more difficult to learn than sending, because you can't work ahead, keep your progress even by sending messages no faster than you can receive them.

The C-128 version of the program is written with Basic 7.0 commands, and it will run in either 40- or 80-column mode. In programming the Sound statements, I used a musical note (D, fifth octave) to produce the clean tone, and I positioned text on the screen with the CHAR statement. The Fast command at the beginning of the program speeds up initialization to less than five seconds.

To encourage you to use Morse Code, I'll leave you with these final words:

···· ·— ···— · ··—· ··— —·

When you've figured them out, you'll be on your way.■



www.Commodore.ca
May Not Reprint Without Permission

# Math Square-Off

**By Michael Broussard**

## RUN It Right

*C-64; VIC-20; C-128 (in C-64 mode)*

Math Scramble is neither as easy as it looks when you read the directions, nor as difficult as it seems when you first try it.

### PREPARATION FOR PLAY

When the game begins, you are prompted to set the level of difficulty, from 1 to 4. Each level corresponds to an arithmetic function. Level 1 is addition; 2 is subtraction; 3 is multiplication; 4 is division.

If you choose 1, only addition will be used. If you choose 2, both addition and subtraction are required, and so on up to 4, at which point all four operations are employed. Remember that operations are chosen at random, so there is a chance that a game board will be generated that contains, for example, no multiplication, even though you specified level 4.

After you choose the level of difficulty, the computer asks you for the highest digit you want to use. Enter a number from 1 to 9; the problems will then include digits only up through the one you chose. For example, if you press 6, all numbers used will be less than or equal to 6. Thus, you can begin children with low digits and level 1, and then adjust the digits and level as they progress in ability.

The last question the computer asks before starting the game is whether or not you want to include negative numbers. If you answer yes, it's possible that one or more of the problems will involve negative numbers. If you answer no, all numbers will be positive. For example, you'll never get a problem like $1-4+8$, because, even though the answer is positive (5), you'll run into a negative intermediate result as you solve the problem.

Also, the program will never generate a problem that involves fractions, even in intermediate results.



www.Commodore.ca
May Not Reprint Without Permission

## HOW TO PLAY

Math Scramble is played on a grid. Each row and column is a math problem that you must solve within a limited amount of time. (A game lasts for 100 seconds or until you solve the puzzle.)

There are actually six problems on the grid—three across and three down. The answers appear in the appropriate places outside the grid. The six problems in this example are:

```
2 × 6 + 3 = 15
3 + 4 × 5 = 35
4 × 2 − 5 = 3
2 × 3 − 4 = 2
6 + 4 ÷ 2 = 5
3 + 5 + 5 = 13
```

You must solve the problems step by step, from left to right or from the top downwards. The answer to problem 2 is 35 because you add 3 + 4 to get 7, and then you multiply by 5 to get 35. This procedure differs from the standard mathematical convention, where multiplication and division are performed from left to right before any addition or subtraction is done.

The grid consists of arithmetic operations and numbers (randomly chosen) that are not in their correct positions. To solve the grid, you must swap numbers from one position to another until you create a sequence that equals the given answer.

Each grid position is numbered according to the following diagram:

```
1 2 3
4 5 6
7 8 9
```

When the computer prompts you to enter the numbers you wish to swap, type in their respective grid positions. For example, to switch the number in the upper-left corner with the number in the lower-right corner, you would type in 1 and 9. If you prefer, you may use certain keys on the keyboard as a keypad. These correspond to the numeric grid positions as follows:

```
1 2 3  :  I  O  P
4 5 6  :  K  L  :
7 8 9  :  ,  .  /
```

Instead of pressing, for example, the 3 and 7 keys, you could press the P and , keys. These keys are located in a cluster, and, with a little practice, you should be able to move numbers quickly, without taking your eyes off the screen.

A timer at the top of the screen steadily counts down from 100 and turns red when you have 20 seconds left. Once you solve the grid, your score is the number of seconds remaining on the timer. You lose if you don't solve the puzzle before you run out of time, and the small grid on the



www.Commodore.ca
May Not Reprint Without Permission

left side of the screen will show you the correct solution.

You may find that 100 seconds is not enough time for your second- or third-grader to solve the puzzle. You can lengthen (or shorten) the game time by changing the value assigned to the variable CLOCK in line 1 of the program.

During play, the grid will be updated each time you specify positions to swap. When a column or row contains the correct values in the correct order, the equation's answer will appear in inverse video, signifying that part of the grid is correct.

Be careful! The computer is looking for only one correct answer to a whole grid. If you enter a row or column whose sequence yields the correct result, but whose answer doesn't appear in inverse video, then either the numbers in the problem are not the right ones (even if the result is correct) or the numbers are in the wrong order.

For example, you may see a row that looks like this:

```
2 + 3 + 6   11
```

Although the numbers add up to 11, the order in which they appear may be incorrect. Perhaps the row should read:

```
3 + 2 + 6   11
```

Or, it may mean that the answer for which the computer is looking actually consists of other numbers:

```
4 + 1 + 6   11
```

Once the game is over, you'll be asked if you want to play again, and the box in the screen's upper-left corner will be updated to show the highest score so far.

Math Scramble is challenging for players of all ages. Try it on your friends who think they hate math—they'll be pleasantly surprised by how much fun it can be!■



www.Commodore.ca
May Not Reprint Without Permission

# Micro Artist

**By Douglas G. Gannon**

## RUN It Right

*C-128 (in C-128 mode); C-16; Plus/4; joystick*

One of the advantages of the C-128, C-16 and Plus/4 is that you can easily perform high-resolution graphics on them. The following programs let you draw high-res graphics on any of these machines, using your joystick.

### C-128 MODE

Plug your joystick into port 1 and load and run Micro Artist 128. You'll be prompted for five colors; select numbers 1 through 16.

Use the W key to set the pixel width of your line (one to eight pixels). Number keys 1–3 switch you between colors 1–3; 0 is for erase. You can modify the program for a wider number of pixels in line 120, and you can change the starting position of the dot in line 90 by changing the X and Y values.

It's interesting to note how colors are set in line 70; the use of the Joy function in line 100, with its related routine (lines 170–250); and the Draw statement in line 280, which is the heart of the program.

If you decide to change the colors after you've already started, just press the run/stop and restore keys, remove the ,1 at the end of the Graphic statement in line 80 and run the program again. This will cause the colors to bleed where the new color touches the old.

### C-16 AND PLUS/4, TOO

With Micro Artist +4 for the C-16 and Plus/4, you can draw lines of varying widths (one to eight pixels) and select from all 121 colors for your background or lines.

To change one of the C-16's or Plus/4's 16 primary colors, press C. To change luminance (there are eight levels), press L. To vary the line width, press W. The line is set at a maximum of eight pixels, but you can easily modify this number in line 160.

At the beginning of the program, you'll be prompted to set the background color and lumi-



www.Commodore.ca
May Not Reprint Without Permission

nance. If the screen is blank after the program continues, then your background is probably the same color as your lines. Just press C, and a dot should appear at the center of the screen. If you don't want the line to start in the middle of the screen, you may move it by changing the X and Y values in line 90.

To change the line from solid to textured and create a transparent appearance, move the joystick at a 45-degree angle from the center. You can erase an area by changing your line color to the background color and going back over the area. To exit the program, press the run/stop key, then the F1 key. In line 10, the F1 key has been programmed to return to the Text mode.

Other points of interest in this program are the flash-on and flash-off symbols used in the title (line 20) and the very easy-to-use Joy function in line 100, with its related routine in lines 200–280.

With the Micro Artist programs, you will be able to enjoy immediate results as you explore the graphics capabilities of your Commodore.■



www.Commodore.ca
May Not Reprint Without Permission

# Mega-Magic—November

**By Joseph Shaughnessy**

## RUN It Right

*C-128; 1571 disk drive*

Here are four short programs that enable you to use the 1571 disk drive's fast loading speed to load a C-64 Basic program into your C-128 *in 128 mode* and then run it in C-64 mode.

To make use of any of these programs, you have to prepare your C-64 disks for loading in C-128 mode in one of the following three ways:

*1.* Save to your C-64 disk an autoboot-maker program, such as Autoboot Maker (*RUN*, December 1985, p. 70) or the one on the 1571 test/demo disk.

*2.* Save to the same disk one of the two C-128 start-up programs with this article, SERMAKER PRG1 or SERMAKER PRG2.

PRG1 lists the disk directory to the screen, and instructs you to DLoad the program you want. After it's loaded, you type in either SYS 5000 to run the program in C-64 mode or SYS 5003 to list and edit the program.

PRG2 is more automatic. You enter the filename of the C-64 program in line 10 and save it to the same disk. Then, each time you load and run it in C-128 mode, it will load the C-64 program in 128 mode and then switch the computer into 64 mode to run it.

*3.* Then run SERMAKER PRG3 to create and save a program entitled Serendipity to each disk containing C-64 programs.

Once you've prepared a disk, insert it into the 1571 and press the C-128's reset button. Your C-64 program will fast-load in 128 mode.

You can load C-64 Basic boot programs in 128 mode unless they contain an auto-run feature. In that case you must BLoad, in 128 mode, the machine lan-



www.Commodore.ca
May Not Reprint Without Permission

guage program that's loaded by the auto-run boot; then load the boot afterwards in 64 mode.

Once the 64 machine language program is BLoaded in 128 mode, you can access 64 mode by typing in the direct command GO64. Unless the boot loads more than one program when it's executed, you can BLoad the main program in 128 mode and use the SYS command, followed by the starting address of the program, to execute it in 64 mode. PRG4 is a program for finding the starting address of a C-64 program saved on disk.

Unfortunately, C-64 programs that use an auto-run feature won't load in C-128 mode. Neither will programs that use LOAD "PROGRAM NAME",8,1 and a SYS command less than 4864 ($1300 hex).■



www.Commodore.ca
May Not Reprint Without Permission

# Basically Speaking—November

**By Scott M. Huse and William D. Taylor**

## RUN It Right

*C-64*

COMM AND-OR 64 is a program that produces quick reference tables designed to help programmers understand the logical (Boolean) operators AND and OR in the Basic language. While COMM AND-OR is a C-64 program, the concepts it illustrates apply to programming in any version of Basic.

Programmers usually use AND and OR to compare corresponding bits in two bytes (the operands) to see if the bits are on (1) or off (0). The logical AND generates a result of 1 only if *both* corresponding bits are 1; if only one or neither of the bits is 1, the result is 0. The logical OR generates a 1 if *either or both* operand bits are 1, and a 0 if neither is. The truth table in Table 1 presents all the combinations of the bit values 1 and 0 and their AND and OR results.

Table 2 details AND and OR operations on two sample decimal numbers, 170 and 37. The result, or "product," of ANDing 170 with 37 is 32 decimal, which means that all the product bits except bit 5 (2 to the fifth power) are off. When 170 is ORed with 37, the product is 175, which means that all product bits except 2 and 4 are on.

| AND | OR |
|---|---|
| 1 AND 1 = 1 | 1 OR 1 = 1 |
| 1 AND 0 = 0 | 1 OR 0 = 1 |
| 0 AND 1 = 0 | 0 OR 1 = 1 |
| 0 AND 0 = 0 | 0 OR 0 = 0 |

*Table 1. AND-OR truth table.*



www.Commodore.ca
May Not Reprint Without Permission

| Bit number | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit value | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Binary notation for 170 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Binary notation for 37 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| AND result | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| OR result | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

*Table 2. ANDing and ORing decimal number 170 with 37.*

Programmers commonly use AND and OR to test if certain bits are on or off. If you AND a byte with the value of a bit number, the result is that bit value if the bit was on, and 0 if the bit was off. For example, referring again to Table 2, if you want to know whether bit 4 is on in the byte containing the decimal value 170, AND the byte with 16 (the decimal value of bit 4). Since bit 4 was off in this case, your product is 0. If you'd tested for bit 3 (by ANDing with 8), the result would have been 8, since bit 3 was on.

To turn on a bit (or bits) in a byte, OR the byte with the sum of the decimal values of the bit(s) you want to turn on. For example, to turn on bits 0–3, OR the byte with 15 (8 + 4 + 2 + 1). To turn off a bit (or bits), AND the byte with 255 minus the sum of the values of the bit(s) you want to turn off.

To better understand how this works, type in and run COMM AND-OR 64; then input any number from 0 to 255. The computer will translate your number into binary notation, then AND your number with a randomly generated number and display the result. Next it will take that result and OR it with another randomly generated number, displaying the second result. Keep typing in numbers, and soon you'll be confident enough with these logical operators to write them into your own programs.■



www.Commodore.ca
May Not Reprint Without Permission

# Extra! Newsletter Graphics

**By David A. Hook**

## RUN It Right

*C-64*

The concept of a disk-based magazine was realized by *RUN* with an article entitled "Extra! Extra! Read All About It" in the July 1986 issue. The article presented three programs, Mag Production, Mag Reader and Mag Printer, that create a software newsletter, display it on your screen and even print it out.

The July version (6.4) of Mag Reader lets you include one graphics display, in either Doodle! or Koala format, as the cover of the newsletter. Now, for the ReRUN disk, we have included Mag Reader Version 7.2 in its entirety. With this update, you can include several pictures in Doodle! or Koala format, or both. The pictures appear as separate displays between text screens.

Perhaps you'd like to give each article its own cover picture or enhance a tutorial article with illustrations. You could also include a "gallery" of graphics displays, and your organization could sell advertising, in the form of Koala or Doodle! screens, as a source of revenue.

Mag Reader 7.2 lets you include pictures anywhere you want in your newsletter. It's totally compatible with Mag Reader 6.4.

### UPDATE DESIGN

I wrote Mag Reader 7.2 to accommodate a maximum of ten extra pictures, beyond the cover. I took into account the C-64's memory capacity, as well as the storage requirements for the various kinds of files involved. The relative file storing text for a newsletter will vary in size, depending on the number of pages, but a typical issue, with one cover picture, should occupy about half a disk. That leaves the other half for the extra graphics, and Doodle! and Koala



www.Commodore.ca
May Not Reprint Without Permission

pictures fill 37 and 40 blocks of disk space, respectively.

## CUSTOMIZATION

As with Mag Reader 6.4, you must customize Mag Reader 7.2 through Data statements each time you use it. The customizing information includes the filename of the text, the filename and type (Doodle! or Koala) of the cover page and title information for the specific issue, such as month or volume number. The steps for entering this information are described in detail in the July article.

Version 7.2 requires information pertaining to the extra graphics, as well. In the Data statement in line 7121, you must indicate how many extra pictures there will be, not including the cover page. Type in 0 if there are no extra pictures.

The Data statements beginning at line 7151 are for specific information on the extra graphics files—one statement per file. The statement must include the number of the text page *before which* the picture will appear, the picture's Doodle! or Koala filename, its type (D or K) and the color of the screen border that will surround it. The filename conventions are the same as those the July article describes for naming the cover picture. If your present copy of Mag Reader includes "extra" data lines, delete them.

Once you've typed this information into the Data statements, save Mag Reader to your newsletter disk. Then use a file copy program to transfer the picture files to the same disk, and your newsletter will be ready for reading.

## TECHNICAL TIDBITS

If there's a picture to be displayed between, for example, pages 3 and 4, the picture appears first when you hit the function key to bring up page 4. To get to page 4 when you've finished viewing the graphics, press the *space bar*, not the function key. If you press the function key, the program will probably skip, jumping to page 5. Always exit a picture screen with the space bar, unless, of course, you *want* to skip a page.

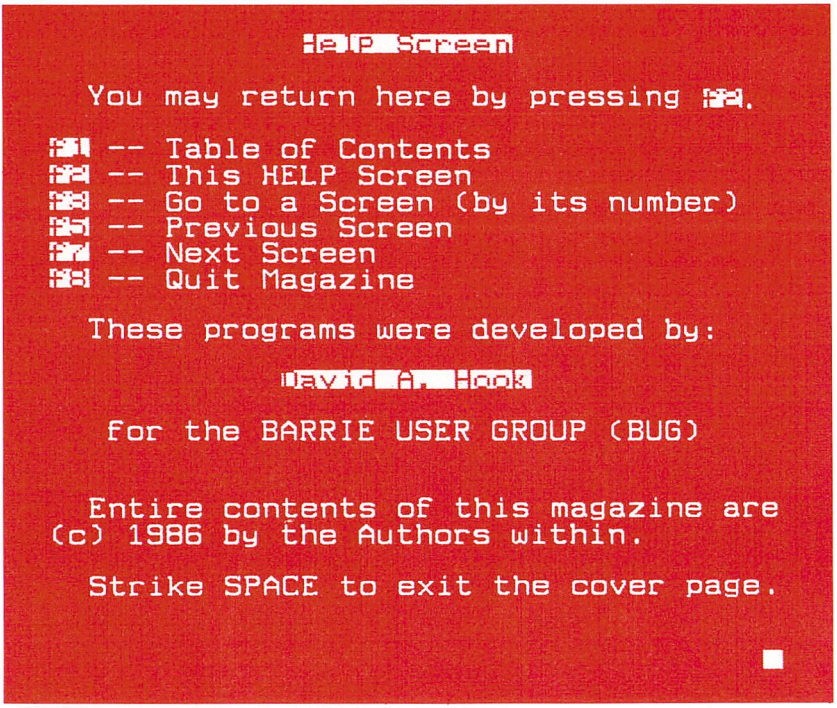
There's room for only one picture at a time in the C-64's memory, so whenever a picture is loaded, it replaces one that's already there. Even the cover page is replaced, and you can't get it back until you run the program again. If you back up to page 0, you'll see the picture that was last loaded, not the cover.

## GETTING THE ORIGINAL VERSION

If you don't have the July 1986 issue of *RUN* with the original article, you can get it by sending



www.Commodore.ca
May Not Reprint Without Permission




$3.50, plus $1 shipping and handling, to *RUN*, Back Issue Orders, 80 Elm St., Peterborough, NH 03458. If you'd rather not type in the program listings, get them on the July/August ReRUN disk by sending $21.47 to ReRUN, at the same address. The disk also includes eight other programs from the July and August issues of *RUN*, so it's a worthwhile investment.■



www.Commodore.ca
May Not Reprint Without Permission

# RUN Script 128

**By Robert Rockefeller**

## RUN It Right

*C-128; 80-column monitor; disk drive; ML Perfect Typist 2.0 program*

RUN Script 128 Version 2.40 is an 80-column word processor for the C-128. All features that were present in RUN Script 64 Version 1.0 (see the March and April 1986 issues of *RUN*) are still present in this version, but there are many enhancements and additions. There are 18 new dot commands and about 30 additional new features. RUN Script 128 gives you about 68,000 bytes for your documents. To use RUN Script 128, load and run "RS128". Be sure that your C-128 is in 128 mode.

What follows is a rundown of this powerful program's many functions and commands, with brief descriptions to get you going. A slash separating two keys means that you must press the keys simultaneously. Otherwise, press them in the sequence shown.

### CURSOR MOVEMENT

**Cursor keys:** move cursor in four directions.
**Shift/return:** moves cursor to start of line.
**Home:** moves cursor to upper-left corner of screen.
**Two homes in succession:** move cursor to top of document.
**CTRL/back arrow:** moves cursor to bottom-left corner of screen.
**Two CTRL/back arrows:** move cursor to end of document.

### ABORT OPERATION

**CTRL/Commodore:** use to exit most operations.
**Restore:** to abort and return to Text mode.

### INSERT TEXT

**Shift/INST:** insert one space.
**CTRL/i:** toggle in and out of Insert mode.
**Run-stop:** insert block of 200 spaces.
**Shift/run-stop:** erase leftover spaces from inserted block.



www.Commodore.ca
May Not Reprint Without Permission

## DELETE TEXT

**DEL:** delete character to left of cursor.
**F7:** delete character under the cursor.
**F8:** delete block. Position cursor at start of block; press F8; move cursor to end of block; press return. Block is saved in buffer.
**Two F6s:** retrieve previously deleted block from buffer. Press F6 twice; select insert point; press return.
**Two F8s:** delete from cursor to end of document.

## MOVE AND COPY TEXT

**F5:** move block. Use same procedure as for F8, above, but complete the move by placing the cursor at the desired insert position and pressing return.
**F6:** copy block. Use the same procedure as for F5.

## SEARCH AND REPLACE

Note: Unlike version 1.0, you don't need to insert null characters at the beginning and end of your search string.

**F2:** search for string.
**Two F2s:** search for multiple occurrences.
**F4:** replace string that was found in a search.
**Two F4s:** automatic replacement of string wherever found.

## COLORS

**CTRL/1:** text color.
**CTRL/2:** screen color.
**CTRL/3:** status line color.

## EDITING FUNCTIONS

**Return:** end a paragraph.
**CTRL/x:** interchange the two characters just before the cursor.
**CTRL/6:** toggle in and out of Capital Lock mode.
**CTRL/=:** selective replacement of line-padding. Default is dots.
**Help:** move cursor to another text area.
**CTRL/z:** five-spaced tab.
**CTRL/u:** select alternate character for block space insertion.

## F1 COMMAND KEY

Press F1 to evoke Command mode. Then, press the appropriate letter or symbol and follow on-screen prompts.

**d:** select device number.
**s:** save text as program file.
**@:** save with replace.
**l:** load program text file.
**a:** append program text files to end of document. Appended text starts at current cursor location.
**t:** select true or CBM ASCII.
**w:** save text as sequential or user file. Enter filename,s,w or filename,u,w.
**r:** load sequential or user text file.
**$:** list disk directory.
**>:** send disk command. Use Basic 2.0 DOS commands.
**<:** read error channel.
**f:** free bytes remaining.
**x:** exit to Basic.



www.Commodore.ca
May Not Reprint Without Permission

**p:** print text to screen, printer or disk. Follow screen prompts.
**0:** load new character set.
**1–9:** load new print sets.
**Cursor-up and cursor-down:** fast line scrolling.
**Cursor-left and cursor-right:** page scrolling.
**c:** toggle upper- or lowercase.
**R:** save table of redefined characters. (See also dot command .dc, below.)
**o:** print document; continuous feed.
**Z:** swap text areas for banks 1 and 0.
**M:** select macro characters to be turned off for printing.
**m:** load macro set, created with Define Macros program. (See the January 1987 issue of *RUN* or the January-February edition of ReRUN for macro details and listing.)
**g:** toggle Special Graphics mode.
**G:** select printer type.
**C:** select secondary address for printing dot graphics and sending macro command strings to printer.
**S:** split text into two areas. Select 1–18 K for area 2.
**A:** move between two text areas.
**T:** select secondary address used to print text to printer.
**L:** enable linefeeds.
**q:** save RUN Script machine language code and all your current settings to disk.

## DOT COMMANDS

RUN Script is a post-formatted word processor. This means that the text is not formatted until it's printed, so your screen display will vary from your printout. To specify output format, RUN Script 2.40 has about 40 dot commands, so called because each command must be preceded by a dot (a period). These commands are embedded in the text to specify margin widths, define headers and footers, and so forth.

A dot command is executed when text is printed, and only text following the dot command is affected. For example, if you don't place the dot commands to set margins until you're halfway through a page of text, the margins in the first half will have the default width when printed.

Four steps must be followed for dot commands to be interpreted correctly. First, the line immediately preceding a line of one or more dot commands must end with a return. Second, the line of dot commands must start in the first screen column. Third, multiple dot commands in a string must not be separated by spaces. Fourth, each string of dot commands must end with a carriage return. Following is an example of a string of dot commands:

```
.pw80.pl66.lm6.rm6.tm4.bm4.lj
```



www.Commodore.ca
May Not Reprint Without Permission

The dot commands are:

**.lj**—Left-justifies printed text.

**.rj**—Right-justifies printed text.

**.cn**—Centers printed text between the left and right margins. Your text must begin on the next line below the .cn command.

**.pl**—Sets the page length. This command must be followed by a number from 1–240, indicating the number of lines you want to constitute a full page. For standard-size, 8½-by-11 paper with a printer that prints six lines per inch, the setting would be 66. Therefore, you would enter .pl66.

**.pw**—Sets the page width, defined as the maximum number of characters that may be printed on one line. Most printers print ten characters per inch, which, using standard-size paper, gives 80 characters per line. If your printer has multiple character sets with different character densities, you'll have to adjust the page width accordingly to use an alternate character set. Example: .pw80.

**.lm**—Sets the left-margin width. The .lm directive must be followed by a number from 1–240. Example: .lm8.

**.rm**—Sets the right-margin width. The .rm directive must also be followed by a number from 1 to 240. Example: .rm8. If the sum of the left and right margins is greater than the page width, a margin error will occur.

**.tm**—Sets the top-margin depth. The .tm directive must be followed by a number from 1 to 240. Example: .tm6.

**.bm**—Sets the bottom-margin depth. The .bm directive must be followed by a number from 1 to 240. Example: .bm6. The top-margin depth plus the bottom-margin depth must be less than the page length.

**.hd**—Defines a header to be printed at the top of every page. The command may be followed by up to 255 characters and must end with a carriage return. This means that the carriage-return character cannot be part of the header string and that no dot commands can follow the .hd directive, since they would be interpreted as part of the header string. For this reason, and to enhance readability, I recommend that the .hd and .ft (see below) dot commands appear on lines by themselves. The # character following the .hd (or .ft) has special significance. Entered just once at the beginning of a document, .hd# automatically prints the page number of each page.

The .lj, .rj, .cn, .pw, .lm and .rm dot commands have no effect on the header. The header string is printed exactly as defined, starting in the first column on the page and continuing until it has been printed in its entirety.



www.Commodore.ca
May Not Reprint Without Permission

If, for example, you wish to center a title, the title must be preceded by the correct number of spaces.

You may embed macro characters (see discussion above) in the header string if you wish. The header will be printed at the line position equal to the .tm setting plus 1. So, if you set the top margin to 6, the header will be printed on the seventh line. Example: .hd RUN Script 2.40.

**.ft**—Defines a footer to be printed at the bottom of every page. The .ft directive works exactly like that of the header. The footer will be printed at the line position equal to the .pl setting minus the .bm setting minus 1. Example: .ft page number.

**.hs**—Defines the number of lines to be left between the header and the main body of text. The command must be followed by a number. Example: .hs2.

**.fs**—Defines the number of lines to be left between the footer and the main body of text. Example: .fs2.

**.ls**—Sets the line spacing. You can print one or more blank lines between each line of text. For double-spacing (one blank line between lines of text), set .ls to 1. Example: .ls1.

**.l+**—Indents text from the current left-margin setting. For example, if the left margin is set with .lm8 and you execute .l+3, text will henceforth be indented as though you'd set .lm at 11. To cancel an indent, use .l+0 or .l–0.

**.l–**—"Outdents" text, such as a subheading, to the left of the left margin. For example, if the left margin setting is .lm8 and .l–3 is executed, text will begin printing at the sixth column, just as though .lm had been set at 5. Outdents are canceled with .l–0 or .l+0.

**.fp**—Forces a new page. When this command is executed, no more text will be output to the current page. If a footer was defined, the footer and bottom margin will be printed, and then a new page will be started. Example: .fp.

**.fl**—Links files to be printed. The command must be followed by a device number and a filename, separated by a comma. When the .fl directive is executed, the specified file will be loaded from the specified device and begin printing. The permissible device numbers are 8 or 9 for disk. If two disk drives are used, one document could even slightly exceed 340,000 characters in length. Example: .fl8,next file.

**.p#**—Sets the page number of the next page to be output. Example: .p#45.

**.lf**—Prints a linefeed character after every carriage return. Some non-Commodore printers require this. Example: .lf. (In ver-



www.Commodore.ca
May Not Reprint Without Permission

sion 2.40, .lf1 enables linefeeds; .lf0 disables them.)

**.cm**—A handy dot command that lets you leave a comment for yourself that won't be printed. For instance, if you're in the habit of keeping all your old letters on disk, you can leave comments to yourself with dates and other information. Example: .cm July 19, 1985 RUN Script 64 article.

**.r+**—Indents the right margin. This code must be followed by a number from 0 to 255. For example, if you set the right-margin width to 10 spaces with .rm10, and then execute .r+5, the effect is the same as if you had executed .rm15. A right indent is canceled by executing .r+0 or .r-0. Example: .r+5.

**.r-**—Makes a right outdent. The command must be followed by a number from 0 to 255. For instance, if you've set the right margin to 10 with .rm10, and then execute .r-5, the effect is the same as if you had originally executed .rm5. A right outdent is canceled by executing .r+0 or .r-0. Example: .r-3.

**.bj**—Stands for "both justify" and prints the ensuing text with both the left and right margins justified. It does this by inserting extra spaces between words so that each line is flush with the left and right margins. You cancel the .bj command by executing an .lj, .rj or .cn command. Example: .bj.

**.pr**—Sends a sequence of up to 98 bytes to the printer using a specified secondary address. The secondary address must immediately follow the command; then the bytes to be sent to the printer, separated by commas, must follow after the secondary address.

Example: .pr2,36,36,36,46,57,57. This example will send bytes 36, 36, 36, 46, 57 and 57 to the printer using secondary address 2. On the MPS-1000 printer, this would set up a formatting string.

Example: .pr5,27,69. Here, bytes 27 and 69 will be sent to the printer using secondary address 5.

**.ta**—Stands for "text address," and must be followed by a number from 0 to 31. The number is the secondary address that will be used to print the ensuing text, and it lasts for only one printout. You can set the default (permanent) text secondary address with the F1,T command. The .ta command is useful with printers, such as the MPS-1000, that have features that can be accessed only through special secondary addresses.

Example: .ta1. This example sets the print secondary address to 1. On the MPS-1000, text



www.Commodore.ca
May Not Reprint Without Permission

printed using secondary address 1 is formatted according to a previously-defined format string (which can be sent to the printer with the .pr command).

**.ca**—Defines a new, temporary secondary address. This is the secondary address that will be used to send macro strings to the printer and to print dot-graphics characters. You can set the default (permanent) secondary address with the F1,C command. Example: .ca5.

**.dc**—Stands for "define character." This command takes three forms that enable you to redefine any character on the keyboard except @, to a total of 127 characters, then to undefine them. The total is enough to redefine the entire keyboard if you wish. All redefined characters are erased before each printout.

Example: .dca,66. To define a character, follow the .dc command with the character to be redefined, a comma and the new decimal value of the character. The Commodore ASCII decimal value for the letter "a" is 65. Here, the value has been changed to 66, which is the value for "b." Therefore, at printout time, all the a's in the document will print out as b's.

Example: .dca–. The minus sign following the character undefines that character. This example would undefine the character "a", removing it from the table of redefined characters.

Example: .dc@. This would erase the entire table of redefined characters, effectively undefining all of them.

**.lr**—Stands for "load redefined characters." This command is used only after you've performed three steps. First, you must define a number of characters with .dc commands. Second, print the current document; third, save the table of redefined characters to disk with the F1,R command. Then you can use .lr to load the saved characters during a later printout. This is useful when you have to redefine many characters—to access special characters on a printer, for example. The .lr command must be followed by a device number, a comma and a filename.

Example: .lr8,filename. This example would load the file of redefined characters named "filename" from device 8, the disk drive.

**.lc**—Stands for "load characters," and enables you to load a new character set or a new print set during printout. The .lc command must be followed by a number from 0 to 9 (0–6 for a C-64) that specifies the set to be loaded. A 0 loads a new set onto the screen; the numbers 1



www.Commodore.ca
May Not Reprint Without Permission

through 9 load a new set into the printer. After the number comes a comma, followed by the device number from which the set is to be loaded, then another comma and the filename of the character or print set.

Example: .lc0,8,special set. Here, the new set would go to the screen from device number 8, the disk drive.

**.el**—Stands for "empty lines," and prints the specified number of carriage returns. If the number of empty lines to be printed is greater than the number of remaining lines on the page, a force page is executed instead.

Example: .el10. Here, ten carriage returns will be printed.

**.st**—Stands for "stop." This command works only with the printer or the monitor, not the disk drive.

Example: .st. When not followed by any parameters, the .st command terminates Continuous mode during printout and starts Single Sheet mode after the current page is done. The command may be placed anywhere within the text area.

Example: .st5. When followed by a parameter, .st interrupts continuous output at a specified page number, anywhere from 1 to 65535, and enters Single Sheet mode. In this example, if continuous output has been selected with F1,p or F1,o, the first four pages will print in Continuous mode, then at page 5 the "next output (c/p/s) ?" prompt will appear.

**.ff**—Stands for "form feed." The .ff command must be followed by number 0 or 1. If you select 1, the bottom margin will be printed by sending the form-feed character to the printer. The Default mode, .ff0, prints the bottom margin with carriage returns. Example: .ff1.

**.fc**—Stands for "force conditionally," and must be followed by a number from 0 to 255. If less than the specified number of lines remain on a page when .fc is executed, no more text will be printed on that page. Instead, a force page will be executed, printing the footer (if any) and bottom margin immediately and resuming text output on the following page.

Example: .fc20. If less than 20 lines remain when .fc20 is executed, the page will be forced.

**.dg**—Stands for "define graphics character." This command lets you change any character, in any print set or the screen character set, without resorting to a character-set editor. Example: .dg0,1,255,0,0,255,0,0,255,0.

The command must be followed by ten byte numbers, separated by commas. The first byte number specifies the set in which



www.Commodore.ca
May Not Reprint Without Permission

the character to be changed is situated. As always, 0 specifies the screen character set, 1–9 specify a print character set.

The second byte is the screen-code value of the character to be changed. The 1 in the example selects the "a" character.

The last eight bytes define the character itself. The sample setting would change the "a" into three parallel bars.

**.gc**—Stands for "graphics characters." The .gc command is followed by two parameters, separated by a comma. The parameter settings are saved within RUN Script, and are permanent until the computer is turned off, or until you execute another .gc command. Example: .gc1,6. This is the default setting, which is correct for the MPS-803 printer.

The first parameter is a number, 0 or 1, that specifies the orientation of dot-graphics characters printed when you select either a print set or Special Graphics mode. Only one setting is correct for any one printer. If you select the wrong setting, all characters printed in Graphics mode will be printed upside down. The only way to determine the correct setting for your printer is by experimentation.

The second parameter sets the width of Graphics-mode characters in dots. Standard Commodore characters are eight dots wide. So, if you set this parameter to 6, only the first six dots of each character will be printed. If you set this parameter to 10, ten dots will be printed—eight dots of character and two dots of space.

Many non-Commodore printers accommodate three dot densities in Dot Graphics mode: single density with 480 dots per line, double density with 960 dots per line and quadruple density with 1920 dots per line. If you wanted to print 80 characters per line, how many dots wide must each character be for each of these densities?

Single Density mode: $480 \div 80 = 6$ dots wide.

Double Density mode: $960 \div 80 = 12$ dots wide.

Quadruple Density mode: $1920 \div 80 = 24$ dots wide.

Obviously, none of these are correct. To use Single Density mode, you need a print set only six dots wide. If you use Double Density mode, you can print all eight dots of each Commodore character, but four-dot spacing will remain between each dot-graphics character.

You can use double density with a dot width of ten if you're using the elite (96 characters per line) character set for printing normal text. There'll be two dots of space between each dot-



www.Commodore.ca
May Not Reprint Without Permission

graphics character, but you may find this acceptable.

Quadruple Density mode will not work properly, regardless of characters per line.

To make use of the full eight-dot width of a Commodore character and print 80 characters per line, you need a printer with a dot density of $8 \times 80 = 640$ dots per line. Use a dot width of eight if you have such a printer.

In general, use the following formula to calculate the correct dot width: dot width = dots per line ÷ characters per line. If the result is a dot width of less than eight, you'll need a print set that uses only that many columns of a Commodore character.

There's a problem with printing dot-graphics characters with MPS-803-compatible printers. As I said, standard Commodore characters are eight dots wide, and they're also eight dots high. Unfortunately, the MPS-803, 1515, 1525, MPS-801 and MPS-1000 (Commodore mode) need characters that are only six dots wide and seven dots high. There's no way around it—they need special print sets that use only the left six dots and the top seven dots of Commodore characters. Hopefully, *RUN* readers who design such print sets will share them with others.

**.gb**—Stands for "graphics begin," and defines the string of bytes that is sent to the printer to activate Graphics mode. The string is sent before each character is printed in Dot Graphics mode. Example: .gb8.

The MPS-803 and its compatibles require just one character, CHR$(8), to activate graphics mode. This character will put the printer into Graphics mode until it receives a byte value of less than 128, at which point it will return to normal Text mode. Most other printers handle graphics differently, however, and require that more than one character be sent.

Example: .gb27,75,6,0. These four characters will tell the printer to print the next six bytes in Dot Graphics mode. Then RUN Script will send the left six dot columns of a character.

Remember that the string must tell the printer how many bytes of graphics data will be sent. If the dot width is set to 6, six bytes will be sent; if the dot width is set to 10, ten bytes will be sent. On most printers (not the MPS-803), if the dot width is changed, you'll have to redefine .gb also.

Example: .gb27,76,8,0. This example selects the Double Density Graphics mode. With the dot width set to 8, dot-graphics characters will print with a density of 120 characters per line.

**.ge**—Stands for "graphics end." The string of bytes follow-



www.Commodore.ca
May Not Reprint Without Permission

ing the .ge command will be sent to the printer after each dot-graphics character. Some printers require this command to return to Text mode.

Example: .ge15. This setting, the default, is correct for MPS-803 printers.

Example: .ge. This is the correct setting for most non-Commodore printers. No bytes will be sent after the graphics bytes.

## ERROR MESSAGES

**Macro Not Defined**—An undefined macro character has been encountered. This usually means that you made a typing error when entering the macro character. The cursor will rest on the incorrect character.

Macro characters are a means of customizing RUN Script to take advantage of special features of non-Commodore printers. I explain macro definition later in this documentation. However, it's not necessary to understand macro characters to use RUN Script.

**Illegal Quantity**—A dot command's numeric argument is too large or too small. The cursor will appear near the illegal number.

**Output Error**—A hardware error has occurred during printing. If output is to the printer, the error could mean the printer is out of paper or not turned on.

**Syntax Error**—RUN Script is unable to recognize a dot command. The cursor will be near the offending command.

**Margin Error**—The margins have been set to illegal values. Check the dot commands preceding the cursor position.

**Text Area Full**—The text area is full. At this point, you must save your text and erase what's on the screen to continue.

**Device Not Present**—The device referenced in the input/output operation does not respond. Check to see if you used the correct device number and make sure the device is turned on.

**File Not Found**—The file you attempted to load was not found in the load device.

**String Not Found**—A match was not found during a string search.

**Save Error**—A status error occurred while the program was attempting to save text.

**Load Error**—A status error occurred while the program was attempting to load a text file.

## PRINTERS

RUN Script is set up to expect a Commodore printer connected to the serial bus. A combination of a non-Commodore printer and an interface that emulates a Commodore printer will also work.

If your system includes neither of these, you can still use RUN



www.Commodore.ca
May Not Reprint Without Permission

Script by putting the printer interface into Lock mode and using the RUN Script F1,t function to output true ASCII. Another way is to use the Define Macros program (see *RUN*, January 1987 or the January-February 1987 ReRUN) to create a customized set of macros that RUN Script can use to control your printer. Note that you cannot use an RS-232 printer with RUN Script. If you have such a printer, you must output your text to disk and then use a Basic program to print it.

**DAISYWHEEL PRINTERS**

RUN Script 2.40 can do underlining and double-strike characters with daisywheel printers that recognize the back-space character. Type F3, then the ( character to start underlining; type F3, followed by ) to end underlining. Type F3, then [ to start double-strike; type F3, followed by ] to end double-strike. (See also the .bs command.)

To change the print wheel, type F3, then *. When RUN Script encounters the reversed * character, it will stop until you press a key.

**GRAPHICS MODE**

Most dot-matrix printers can print dot-addressed graphics and ordinary text on the same line, and RUN Script 2.40 takes advantage of this feature. When you put RUN Script into a graphics mode, instead of outputting normal text, it uses the printer's dot-graphics capability to print each character. This allows printers such as an MPS-803 to print italics, boldface or underlining, and print foreign character sets such as French, Russian or Greek—in fact, almost any kind of text.

There are two ways to create graphics with RUN Script. One is through a "print set," which is half of a normal character set. A character set contains 256 characters, 128 non-reversed and 128 reversed, so a print set contains 128 characters. How does it work?

You select a print set with the F3 key, the same key that selects macro characters. After you press F3, choose a numeral from 0 to 9. The numeral will appear at the cursor position in the text, in reverse field.

Pressing 0 selects normal Text mode, which is the default; pressing 1–9 enables the Graphics mode. Within that mode, 1 specifies the first print set, 2 specifies the second set, and so on, up to the maximum number of sets your computer can have—nine for the C-128.

For example, say you chose print set 3 and the next character to be printed is an a. The letter a is the second character in the



www.Commodore.ca
May Not Reprint Without Permission

standard Commodore character set (see the *C-64 Programmer's Reference Guide*, page 376, for a listing of the character set), so RUN Script will go to set 3, take the second character of that set, and print it using dot-addressed graphics.

Print sets and the character set are loaded into memory by the boot program, which also loads the RUN Script ML program and initializes the RUN Script system.

You also can print graphics with the Special Graphics mode. It's activated by the F1 key, then g, then answering y at the prompt, "enable special graphics (y/n) ?". You disable the Special Graphics mode by answering n.

When the Special Graphics mode is functioning, the business-graphics characters of the character set, which are accessed by simultaneously pressing the Commodore logo key and any other key, will be printed using the printer's Dot Graphics mode. This occurs only when normal Text mode is selected. All other characters will be printed as normal text characters.

Since RUN Script's character set is used to display text on the screen, this mode has the advantage that characters will appear on the screen exactly as they will be printed. Of course, to get characters other than the Commodore business-graphics characters, you have to alter the relevant characters with the .dg command or with a character-editor program.

Printing RUN Script graphics requires an MPS-803, 1515, 1525, MPS-1000 in Commodore mode, or other compatible printer. RUN Script 2.40 is designed to be used with these printers. However, a 1526 or MPS-802 printer won't work with RUN Script graphics.

If you have a non-Commodore printer that's not compatible with the MPS-803, you'll have to customize RUN Script 2.40 to the printer with the various customization commands available. (See the F1,q command, above.)

For more information about the Graphics mode, see the .gc, .gb, .ge and .dg dot commands I mentioned earlier.■



www.Commodore.ca
May Not Reprint Without Permission

# ML Perfect Typist 2.0

**By Jim Borden**

## RUN It Right

*C-128*

The C-128 machine language listings in *RUN* consist of a series of program lines, each of which begins with a four-digit line number, followed by 13 pairs of hexadecimal characters (letters or numbers, referred to herein as digits). ML Perfect Typist is provided for accurate and easy entry of these listings.

In an ML listing line, the first eleven pairs of numbers after the line number constitute the program's machine code. The last two pairs are checksum values used by ML Perfect Typist. You cannot type in line numbers or spaces; these are automatically entered. Only the 0–9, A–F, insert, delete, cursor-right, cursor-left, and return keys are active during ML line data input. To speed your typing, the C-128's keypad has been modified so that function keys 1, 3, 5 and 7 and the + and – keys produce, respectively, the letters A through F, needed for entering hexadecimal code. You can use the keypad, the normal keys or any combination of the two for entering lines.

### ENTERING NEW LINES

To type in a machine language listing, first load and run ML Perfect Typist in 128 mode. It will automatically go into Line Entry mode and display 0001. You can then begin to enter your ML program. Several keys require some explanation.

The cursor-right key will move the cursor only one space past the last digit you've typed in. Also, if you try to use the insert key to enter a space, a question mark will fill the inserted space. You must then type a digit over the question mark in order for the cursor keys and the return key to work. You



www.Commodore.ca
May Not Reprint Without Permission

can delete the question mark or any other digit with the delete key. This all sounds more complicated than it is.

When you type the second digit of the 13th pair, an automatic return enters the line for you. If the line is correct, you hear a bell, and the next line number is displayed. If the line contains an error, you hear a buzz, and the cursor moves to the first column. In that case, you can move across the line with the cursor keys and correct the error. When you're satisfied with the line, press the return key. The line is then checked, and you'll hear a bell (correct) or a buzz (error).

The delete key has a special function if the cursor is in the first column of the line. Pressing the delete key in that case will pull all the digits to the right of the cursor back one space to the left. This is used mainly to delete an entire line and start over. When the cursor's in any column but column one, the delete key operates normally.

## SHORT LINES

The last line of an ML program is the *only* line that can be a short one, so it's impossible to enter any subsequent line unless the current line is a full one. If you enter a short (last) line, and it has the correct checksum, you'll hear both a bell *and* a buzz. So, if the last line of the program is a short one, be *sure* you hear the bell-and-buzz combination; if you do not, the code of that line will not be added to the ML code you have entered.

## THE MENU

While editing a line, you can call up the following menu of six available options by pressing CTRL/e:

*1) Load*—Use this option to load either a finished program or one you are working on. The program will load any ML code—complete files, incomplete files, even complete files created by the original ML Perfect Typist. Be sure that your disk drive is on before attempting to load a file.

*2) Save*—This option allows you to save what you've entered so far. Unlike the original ML Perfect Typist, ML Perfect Typist Version 2.0 always uses exactly the same method for saving programs, whether they're partial or complete. Again, be sure that your disk drive is on before entering a filename.

*3) List*—This option will produce a hard-copy listing like those published in *RUN*. You'll be prompted to list to the screen or to the printer. If you list to the screen, the line number will appear one line above the data. If



www.Commodore.ca
May Not Reprint Without Permission

you list to the printer, be sure your printer is on-line before selecting the Printer option. Printer output prints the entire line in a format identical to the one in *RUN*. The printer output also sends a form-feed (CHR$(12)) after every 60 lines.

You can pause a listing to the screen with the shift key or exit it with the stop key. If the printer is off, you'll hear a buzz, and the screen will display the last line.

*4) Edit Old Line*—Although it's an unlikely occurrence, it is possible to get a correct checksum by inadvertently swapping the positions of two or more digits in a line. If you finish the ML program, yet it does not work properly, load it into ML Perfect Typist and list it with the List option. When you find the line with the swapped digits or other error, select the Edit Old Line option and enter the four-digit line number (e.g., 0025, not 25). The line will be displayed for you to edit. After you correct it and press the return key, you'll hear the bell and will automatically be returned to the Enter New Lines option. Be sure you resave the corrected program with a new filename.

*5) Enter New Lines*—This is the option you'll use most of the time. Edit mode works as explained above. ML Perfect Typist automatically returns you to this mode after you choose any option but Exit.

*6) Exit*—This option returns you to Basic. You'll get a prompt to ensure that you save the program you've entered before you exit to Basic.

In all options except those involved in entering program lines (options 4 and 5), you are in the Basic screen editor. All keys, including the color keys, for example, are active during input, so be careful what keys you press. The function keys on the C-128 that were defined as A, B, C and D will return to your definitions when you exit to Basic. Finally, you can employ either the 40- or 80-column display, but only the left half of the 80-column screen is used, so it's easier to read if you use 40-column mode.■



www.Commodore.ca
May Not Reprint Without Permission

# Sum Fun

**By Myron B. Achtman**

## RUN It Right

*C-64; C-128*

Shut the Boxes is a dice game for the C-64 and C-128 that challenges the wits of both adults and children. The objective of the game, which can be played by one to four people, is to close the lids of nine boxes according to the "roll of the dice" and to accumulate the *lowest* possible score in the process. The program, written entirely in Basic, includes sound effects and some simple animation techniques.

Each round starts with all nine boxes open. The boxes are numbered 1–9, and after each roll you choose a box or combination of boxes to close whose numbers add up to the face value on the dice.

With a roll of 10, for example, you could select boxes 3 and 7; 1 and 9; 1, 4 and 5; 1, 2, 3 and 4; or any other combination that totals 10. If things are going well and the total value of the boxes you have left to close is 6 or less, the program rolls only one die.

Don't worry if you make a mistake when selecting the numbers. The program will let you know you've made an error and let you correct it. When choosing numbers, it's not necessary to press return after each selection.

The program prompts you for another choice when you haven't selected enough boxes yet, and it prints "that's a match" when your choices equal the face value on the dice.

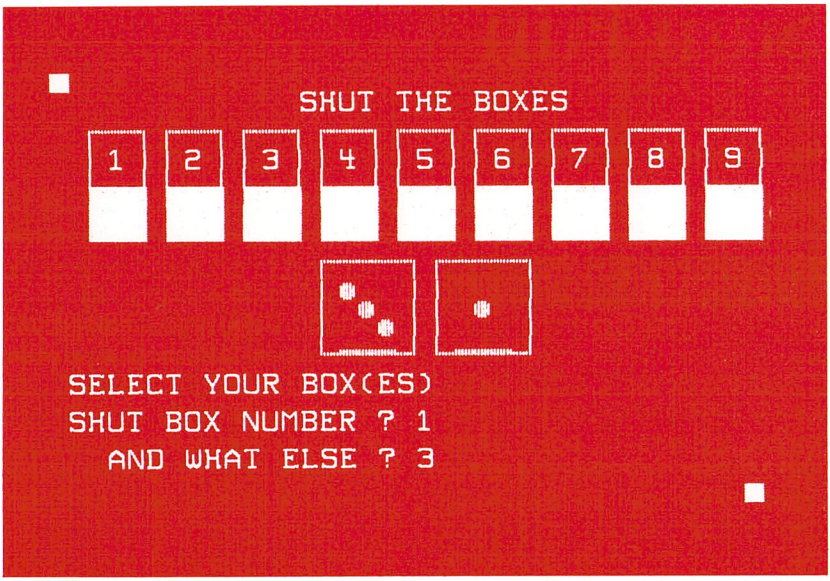
At any time in the game when there's no box or combination of boxes left whose value equals the roll of the dice, you must press the F1 key. This terminates the round and displays your cumulative score. Each participant has three rounds of play to a turn.

### SCORING

Your score is a concatenation, in ascending order, of the numbers of the boxes left uncovered. For example, if boxes 1, 5 and 8 remain uncovered, then your



www.Commodore.ca
May Not Reprint Without Permission




score is 158, not 14! You cannot cheat at Shut the Boxes, because the program constantly checks the choices you make against what is available on the screen.

There's no need to panic if your score starts getting high. At any time during the game, you may be lucky enough to close all nine boxes, and then your score will automatically go back to 0!

At the end of your turn, your score is displayed. If it's the "best low score so far," the bell tone sounds and your score remains in the upper-left corner of the screen. The best low score is always visible in this position, as a reminder of the score to beat.

## THE STRATEGY

As simple as Shut the Boxes may sound, there are strategies to keep in mind. For example, avoid choosing the number 2 box early in a round. If you select 2 during the first roll and your next roll comes up "snake eyes" (a pair of 1s), you'll be stuck with a very high score!

In general, it's best to match the dice with single boxes early in the round. For example, if you roll an 8, you're better off selecting box 8 than 1, 3 and 4. The time to choose multiple



www.Commodore.ca
May Not Reprint Without Permission

boxes is after you've closed about half the lids.

Remember, the more boxes you leave uncovered, the higher your score. If you leave boxes 5, 6 and 9 open, 569 points will be added to your score; if you leave boxes 3, 4, 5 and 8 open, your score increases by a disastrous 3458 points!

Adult players usually are convinced they can combine their intuition and skill with a little help from Lady Luck to win Shut the Boxes. Young players learn from relating numbers to the patterns on the dice and from recognizing numerical combinations that will achieve a desired result. All in all, the game is an enjoyable combination of strategy and luck.■



www.Commodore.ca
May Not Reprint Without Permission

# Mega-Magic—December

**By Scott M. Huse and William D. Taylor**

## RUN It Right

*C-64; disk drive*

Error Channel Monitor is a vector-driven utility that monitors the status of your disk drive's error channel and displays that status on the top line of the screen. It's written for a C-64 with a disk drive.

A vector is a program pointer that resides in a certain location in memory. It stores the two-byte address of another memory location to which a program should jump. It can also include a JMP instruction, for a total of three bytes.

Basic 2.0 contains many vectors in RAM for use as programs run. You can intercept execution and divert it to other assignments by altering these vectors. In this case, the program changes the vector at addresses $302 and $303 (decimal 770 and 771), which usually points to address $A483, to point to address $C000 (decimal 49152), which is where Error Channel Monitor resides.

When you run the program, the screen will clear, the top line will display the error-channel status, and the word "Activated" will appear.

Error Channel Monitor will continue to monitor and display the disk drive's error-channel status as long as the computer and disk drive are both on. It won't interfere with most Basic programs, so you can use it when you're writing and debugging your own programs.■



www.Commodore.ca
May Not Reprint Without Permission

# Basically Speaking—December

**By David Hook**

## RUN It Right

*C-64*

There's an adage in the computer world that no program is ever finished; it's merely abandoned. In my case that's certainly true, as version number 20 might not even be the "final" one. I'll admit that not all those changes are improvements, and often I need to go back to an earlier version to fix something that used to work, but no longer does.

To determine what changes have been made so that I can figure out what to fix, I usually have to compare listings from various stages in the program's evolution—a tedious and frustrating task if I did it by hand. However, I have a program that does it for me.

### A LITTLE BACKGROUND

In the early days of the Commodore PET, I read *Cursor* magazine, and in its September 1980 issue, there was a program by Glen Fisher called Compare. As the name suggests, the program compared two Basic programs in memory.

While I've found the original Compare very useful, its speed leaves something to be desired. So I've recoded the key functions into machine language to increase speed and optimize performance.

### RUNNING THE PROGRAM

When you run this program, called Compare Basic, you'll be



www.Commodore.ca
May Not Reprint Without Permission

prompted to enter the filenames of the programs you want to compare. Default answers are provided at the prompts for designating the device and drive numbers of your disk drive.

Before it starts the comparison, the program makes a disk error check and also checks to make sure the programs to be compared are written in Basic. The algorithm for doing this is based on the fact that the C-64 and the C-128 store the value 1, which represents the "low" byte of the start of Basic, as the first character in the disk file. If Compare Basic doesn't find a 1 at the start of the file, execution aborts.

You can choose whether to send the results of the comparison to the screen or a printer. If you opt for the screen, you can press any key to pause and resume the output.

As the program works, near the bottom of the display you'll see the lines being scanned. The two line numbers appear side by side with a separating colon. The comparison continues until the lines in both programs have been exhausted.

## HOW IT WORKS

The program compares the two subject programs on a line-by-line basis. For example, if each program has a line 10, then both line 10s are read into memory and compared character by character. If they match exactly, then Compare Basic continues on to the next pair of lines. If two lines differ only by nonessential spaces outside of quotation marks, they are treated as identical. Since spaces outside quotes are ignored by Basic, they're ignored by Compare Basic, too.

If one or more significant characters differ, both lines are printed in full. A minus sign will precede the line from program 1, and a plus sign will precede the line from program 2. Compare Basic removes all optional spaces from both lines to facilitate spotting the differences. This has the side effect of merging words together in REM statements.

If a line appears in one program but not the other, that line is output to the screen or printer. Again, a minus or plus sign precedes it.

## PERFORMANCE AND LIMITATIONS

The comparison of a 90-block program takes less than five minutes—an eight-fold improvement over the original Compare. Of course, shorter programs will process more quickly.

Keep in mind that, although Compare Basic can work with



www.Commodore.ca
May Not Reprint Without Permission

Basic 7.0 programs written on the C-128, it won't decipher any instructions, such as Color, Window and Dload, that don't appear in Basic 2.0. Also, if you've renumbered either of the subject listings, the comparison won't work; line numbers must match.

Compare Basic's speed will enable you to compare two programs faster than you could print out a hard copy of just one of them! The results are certain to be more helpful, too.■



www.Commodore.ca
May Not Reprint Without Permission

# Peg Solitaire 64

**By Daniel Miller**

## RUN It Right

*C-64*

HI-Q is a popular board game in which you try to remove a number of pegs from a cross-shaped board in a series of jumps so as to leave one peg in a designated location. An 18th century French nobleman supposedly devised the game while imprisoned in the Bastille. Peg Solitaire 64 is the C-64 version of this famous puzzle.

The program incorporates 14 puzzle variations to test and challenge you. In order to view the various screens, choose option O from the main menu. Press function key F1 to cycle through the puzzle screens; F7 returns you to the menu to make your selection.

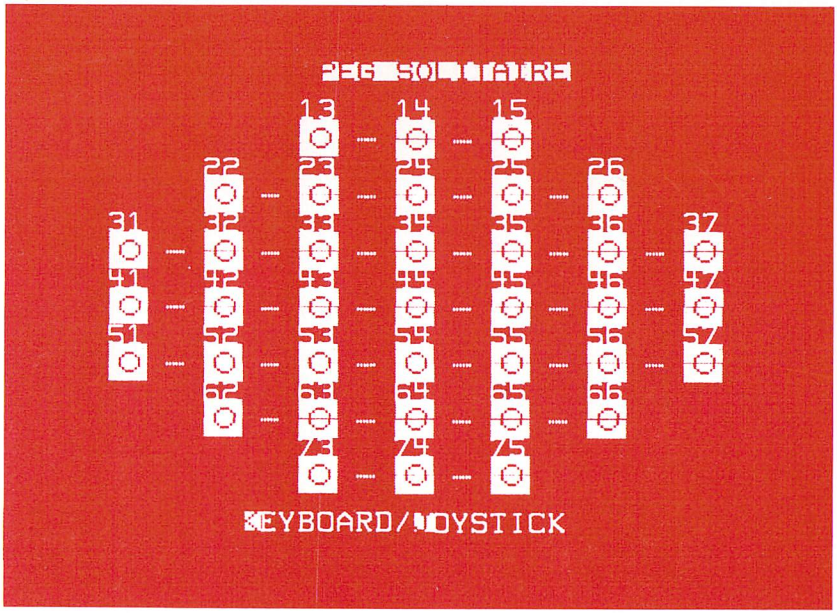
With each puzzle, your aim is to remove the playing pieces in the minimum number of moves. Puzzle tiles are white; empty spaces are black. In several puzzles, there are green corner tiles, but these are not used in play. You make a move only by jumping over a neighboring piece to an empty space, thereby removing the piece you jumped over, much as in checkers.

The playing screen is a field of seven rows by seven columns, and a number in a yellow-bordered, square sprite cursor designates your location on the puzzle board. You enter your moves with either the keyboard or a joystick (connected to port #2). With the keyboard, use I, J, K and M to move the cursor up, left, right and down, respectively.

To begin play, move from the home position to a white-tile location. Then press the fire-button on the joystick or the asterisk on the keyboard to enter that "From," or beginning location. Now make your jump, by moving the cursor, and record your "To" location by again hitting the fire-button or asterisk. If your move is acceptable, the white tile jumps to the "To" position, and the middle tile disappears from the board. Double, triple or more jumps in sequence, as long as



www.Commodore.ca
May Not Reprint Without Permission




each one is legal, are considered a single move.

Try a few moves with the Latin Cross puzzle. This, the simplest puzzle, is a good one to practice on if you've never tried these puzzles before.

To exit from any puzzle, press F7. One option then returns you to the main menu, where you may either select another puzzle or enter P to end the program. With another option, you may go instead to the Solution Screen, where you'll see the moves required to solve the puzzle. If you want a printed copy of the solution, make sure your printer is on-line and again press F7. Press F1 to restart the program.

If you get completely stumped while doing a puzzle, press F1, and the puzzle will immediately reset and go into Auto-Solve mode. The pieces then move and jump on their own, and the proper moves are displayed at the bottom of the screen.

I hope you'll enjoy playing Peg Solitaire 64.■



www.Commodore.ca
May Not Reprint Without Permission

# Please send me back issues of ReRUN

____ Summer Edition
____ Fall Edition
____ Winter Edition
____ January/February 1986 **
____ March/April 1986
____ May/June 1986
____ July/August 1986
____ Productivity Pak II
____ September/October 1986

____ Cassette version(s) at $11.47*
____ Disk version(s) at $21.47*

* *Prices include postage and handling. For foreign air mail, please add U.S. $1.50 per item and $25 per subscription. Prepayment only.*

** *All 1986 editions contain 128-mode programs and are available on disk only.*

☐ Payment Enclosed ☐ MC ☐ VISA ☐ AE

Card # Exp. Date

Name

Address

City State Zip

Signature

**ReRUN • 80 Elm Street • Peterborough, NH • 03458**

# BEAT THE RUSH!

**Please send me:**

☐ **1 year (6 issues) for $89.97**
☐ **January/February 1987 ReRUN disk for $21.47.***

***Available in February 1987.***

***Includes programs for C-64 and C-128 (in both 64 and 128 modes).***

*Price includes postage and handling. For foreign air mail, please add U.S. $1.50 per item and $25 per subscription. Prepayment only.*

☐ Payment Enclosed ☐ MC ☐ VISA ☐ AE

Card # Exp. Date

Name

Address

City State Zip

Signature

www.Commodore.ca
May Not Reprint Without Permission

**ReRUN • 80 Elm Street • Peterborough, NH • 03458**

# 16 *RUN* Programs Included on this Disk:

## Education ▸ Graphics ▸ Word Processing ▸ Spreadsheet Manipulation ▸ Games ▸ Programming Aids ▸ Print Utility

### *From the November RUN:*

- CalcAid 64
- Put It on Paper
- Dashing Off the Dots
- Math Square-Off
- Micro Artist
- Mega-Magic
- Basically Speaking

### *From the December RUN:*

- Extra! Newsletter Graphics
- RUN Script 128
- ML Perfect Typist 2.0
- Sum Fun
- Mega-Magic
- Basically Speaking

### *PLUS Bonus Program:*

- Peg Solitaire 64

If any manufacturing defect becomes apparent, the defective disk will be replaced free of charge if returned by prepaid mail within 30 days of purchase. Send it, with a letter specifying the defect, to:

**ReRUN • 80 Elm Street • Peterborough, NH 03458**

Replacements will not be made if the disk has been altered, repaired or misused through negligence, or if it shows signs of excessive wear or is damaged by equipment.

The programs in ReRUN are taken directly from listings prepared to accompany articles in *RUN* magazine. They will not run under all system configurations. Use the RUN It Right information included with each article as your guide.

The entire contents are copyrighted 1986 by CW Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1986 CW Communications/Peterborough

CW COMMUNICATIONS/PETERBOROUGH

www.Commodore.ca
May Not Reprint Without Permission