

May-June-July 1990 Edition

# RE RUN

## RUN Programs on Disk

For the C-64 and C-128



Plus: Extra Bonus Programs!

[www.Commodore.ca](http://www.Commodore.ca)

May Not Reprint Without Permission

# Introduction

*May-June-July '90 ReRUN*

AS YOU'VE PROBABLY already noted, this edition of ReRUN covers three months instead of the usual two. That's due to the fact that we merged the June and July issues of *RUN* into one. Subscribers can rest assured that ReRUN's normal production schedule is uninterrupted by this move—they will receive the six issues they ordered this year.

We continue our healthy mix of C-64 and 128 programs in this edition. The first program is RUN Copy 128, one of the most useful two disk drive file copying utilities for the C-128 that I've seen in quite a while. Patterned after RUN Copy 64 (*RUN*, November 1989), RUN Copy 128 uses both 40- and 80-Column modes. It's a fast and easy way to copy all different types of files between disk drives, send disk commands and display directories. Owners of 1581 drives will find it particularly helpful.

Also from the May issue is a C-64 program entitled Basic Takes the Wrap. With this handy utility, you position the cursor anywhere on the screen, and it aids in printing text within programs or in Direct mode. Simply follow the easy-to-use syntax, enter the text, and there's your message, properly formatted exactly where you want it on the screen!

Dodge 'Em, written by the prolific and imaginative game programmer Tony Brantner, is a fast-paced C-64 arcade game. The first program from the June/July issue, The Palette-able 64, was created by longtime Magic contributor, Richard Penn, and accurately displays 128 colors on the 64's screen.

Another Tony Brantner original, Bull's-Eye!, is a commercial quality C-64 game that lets you play darts against an opponent. Escape Hatch 128 is a utility that adds a new Escape function to the C-128. Pressing Escape £ with Escape Hatch activated in either 40- or 80-Column mode splits the screen and "remembers" where the split occurred.

If you're a GEOS user, you'll appreciate DA-Runner, which hails from the June/July geoWatch column. This utility, written by Jim Collette (Q-Link's geoRep Jim), is a handy program that lets you

exit a GEOS application, run a Desk accessory and then return to the program you were originally running.

We're offering four bonus programs on this edition of ReRUN! First up is Tony Brantner's Power-Punch. A one- or two-player game, Power-Punch pits you against another player or the computer at various levels of boxing expertise.

We're also including an application for the C-128, called Ledger Check, that lets you monitor your checkbooks and simple business ledgers. It's practical for a variety of financial account monitoring applications.

ALU (Assembly Language Utility) 64 is useful for scanning areas of memory, placing messages in memory for display, performing number conversions and generally assisting in day-to-day programming. Finally, for some more mental stimulation, try Pegs, a C-128 40-column strategy game that has you moving pegs around boards of various shapes.

All of us here at *RUN* will continue to support your favorite computers. I'll be back in a few weeks!



*Technical Manager  
RUN Magazine*

# Directory

PAGE	DOCUMENTATION	DISK FILENAME	FILE TYPE
		* MENU 128 _____	BASIC
		MENU 64 _____	BASIC
1	<b>RUN COPY 128</b>	* RUN COPY 128 _____	ML
		RUN COPY.HEX _____	BASIC
4	<b>BASIC TAKES THE WRAP</b>	PARSER _____	BASIC
6	<b>DODGE 'EM</b>	DODGE 'EM _____	ML
		DODGE 'EM.HEX _____	BASIC
7	<b>THE PALETTE-ABLE 64</b>	128 COLORS _____	BASIC
8	<b>BULL'S-EYE!</b>	BULL'S-EYE _____	ML
		BULL'S-EYE.HEX _____	BASIC
9	<b>ESCAPE HATCH 128</b>	* EDITOR SWAP _____	BASIC
		EDITOR SWAP DEMO _____	BASIC
10	<b># DA-RUNNER</b>	DA-RUNNER _____	BASIC
11	<b>£ POWER-PUNCH</b>	POWER-PUNCH _____	ML
12	<b>£ LEDGER CHECK</b>	* LEDGER CHECK _____	BASIC
13	<b>£ ALU 64</b>	ASSEMBLY LANG.UT _____	BASIC
14	<b>£ PEGS</b>	* PEGS _____	BASIC

\* — C-128 mode only

# — Requires GEOS

£ — Bonus program

Before you run a program, carefully read the documentation that pertains to it.



# 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. C-128 users need only press the shift and run-stop keys. When all the programs are displayed on the screen, you can run the one you select by pressing a single key.

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

It's simple to make a copy of a Basic program. Just load it into your computer as outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

Copying an ML program is not so simple. You cannot simply load and save an ML program; you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.



# RUN Copy 128

*By Ed Parry*

RUN COPY 128 IS A C-128 multfile/multidrive file copier that works with any two C-128-compatible disk drives. It operates just like its C-64 counterpart (published as "Friendly File Copier," *RUN*, November 1989), with the following exceptions:

1. RUN Copy 128 can copy 100, not 200, files at one time.
2. RUN Copy 128 automatically detects 40- or 80-Column mode and reacts accordingly.
3. The escape key replaces the left-arrow key for resetting the program at any input prompt.
4. File copying, with either a 1571 or 1581 disk drive, is several times faster than with RUN Copy 64, because those drives use fast serial input/output in 128 mode.
5. A bell indicates copy completion and any problems.

If you're unfamiliar with RUN Copy 64, here's everything you need to know to get the C-128 version of the program up and running.

## PRELIMINARIES

To use RUN Copy 128, load and run the program from Menu 128. Loading is fast, and soon the self-explanatory main screen appears, showing fields such as filename, file type, file size, files chosen to copy and files actually copied. The screen also offers the three options explained below.

## THE COPY FILES OPTION

Copy Files, the main RUN Copy 128 option, is invoked by pressing the C key. When the program prompts you for the source and destination drive numbers, specify each by using the cursor-right key to increase the default number or the cursor-down key to decrease it, and then pressing the return key to enter your choice. You can input any numbers from 6 (for a hard drive or RAM disk program) through 30. After you've pressed the return key, the



program automatically checks for the presence of each drive and, if one is missing, displays an error message.

When it has identified acceptable drive numbers, the program asks you for a copy pattern. Any standard directory pattern is acceptable, with the default being \* (all files). Examples of acceptable patterns are shown in Table 1.

Next, the program reads the source directory, displays the first matching filename and asks if you want that file copied. Press Y to place the filename in the program's internal list of files to be copied, or N to exclude the filename from the list. Pressing A aborts the copy option and resets the program.

Once you've specified the files, copying begins. RUN Copy 128 uses the standard disk read/write routines, so you can copy files easily from any standard Commodore disk drive to any other.

During copying, the R/W field indicators flash back and forth to indicate which phase, read or write, the program is in. In addition, the # field shows the file number currently being copied, the To Copy field shows the total number of files to be copied, and the Copied field shows the number of files already done. The Copy Complete message appears when the entire process is finished.

You can abort a copy operation at any time by holding down the run-stop key. It may take several seconds for the abort to occur, so keep holding the key down until you see the Copy Aborted message.

Utilities, such as JiffyDOS, that speed up disk read/write operations work well with RUN Copy 128.

## THE DISK COMMAND OPTION

This option lets you issue standard disk commands, so you can scratch files, format disks, validate disks, and so forth, from RUN Copy 128. Several examples of acceptable disk commands appear in Table 2. If you need further help, see the section on disk commands in your drive manual.

Press the @ key to activate the Disk Command option. Then use the cursor and return keys to specify the number of the drive that should receive the command. Here again, acceptable numbers range from 6 through 30.

The program verifies the presence of the drive and then prompts you for the command. Type it and press return. After execution, the program displays the disk status as returned by the drive.

## THE DIRECTORY COMMANDS OPTION

The third RUN Copy 128 option, for displaying disk directories, is activated much like the Disk Command wedge. Specify a drive device number with the cursor and return keys; then, at the \$0> prompt, enter any valid directory pattern (see Table 1) or press return to show the entire directory.

In addition to the normal directory information, RUN Copy 128 shows how many files are in the list. At the Pause message, press any key to return to the main screen.

If you have two disk drives, you'll find RUN Copy a handy utility.

**Table 1. Examples of acceptable copy patterns.**

<b>Command</b>	<b>Pattern</b>
*	All files in the directory
*=P	PRG files only
*=S	SEQ files only
A*	Files whose names start with A

**Table 2. Examples of acceptable disk commands.**

<b>Command</b>	<b>Function</b>
>i0	Initializes a disk (loads the BAM)
>v0	Validates (cleans up) a disk
>n0:target disk,RC	Formats a disk as the target, with an ID of RC
>n0:erase disk	Erases a formatted disk
>u0>m0	Puts a 1571 drive in 1541 mode
>u0>m1	Puts a 1571 drive in 1571 mode
>s0:filename	Scratches a file



# Basic Takes the Wrap

*By Howard I. Goldman*

THE FIRST COMMAND a beginning Basic programmer learns is Print. Yet, even if you have years of experience, setting up an attractive screen display can be a chore. You must keep track of line lengths yourself, since Basic provides none of the automatic parsing (word-wrapping) you take for granted in your favorite word processor. And arranging text with graphics can cause endless frustration.

Things are easier with Parser, a short machine language routine that takes much of the hassle out of managing text displays. It automatically word-wraps while printing text strings to the screen, so you don't have to break messages into several lines of Print statements. Better yet, it lets you specify an exact display format, so that you can easily integrate text with character graphics.

Parser is intended for use in your own programs, not on its own. Hence, when merging it into another program, you'll probably need to renumber Parser's lines.

The program is a Basic loader that reads the machine code in its Data statements and pokes the code into memory.

By default, the machine code is placed starting at memory location 49152. However, you can put it at any address where you're sure you have 141 bytes of memory free (the cassette buffer, or the free area at 679, for example). All you have to do is change the value of SA in line 20.

Once you've run the loader, Parser is ready to use in either Immediate or Program mode. The syntax is:

`SYS SA,<text$>,<left margin>,<characters per line>,<starting row>`

## THE PARAMETERS

All of Parser's parameters may be either literal constants or expressions (using variables and numeric or string functions). An explanation of each parameter follows.

SA is Parser's starting address in memory (the value assigned to variable SA in the loader program).

TEXT\$, the message to be printed, can be any string expression—a string variable, a literal string enclosed in quotes, or a combination of both concatenated with a plus sign. A combination expression might look like:

F\$+T\$+“PRESS ANY KEY”

The complete expression may contain no more than 254 characters.

The left-margin value specifies the starting column for each line of text and ranges from 0 to 39. This parameter is useful for positioning a block of text to the right of a character graphic. If the graphic extends to, say, column 14, you might set the left margin of the text to column 16.

The characters-per-line value, which also ranges from 0 to 39, specifies the maximum number of characters allowed on each line. (Of course, the number of characters actually printed on a line depends on how the words wrap.) This parameter provides an indirect way to set a right margin, enabling you to position a block of text on the left side of the screen.

Finally, the starting-row value, ranging from 0 to 24, lets you position the text vertically. Note that Parser outputs a carriage return after printing the string, so, if your text prints down to the last screen line, the screen will scroll.

## PROGRAM NOTES

I've made Parser as compatible as possible with other machine language routines you may use in your programs. It's relocatable, as I have mentioned, and it doesn't touch zero-page locations 251–254, which are used by many machine language programs.

Because of the C-64's 40-column screen width, the total of the left-margin and characters-per-line values you specify cannot exceed 39. You cannot, for example, specify a left margin of 10 with 32 characters per line. Basic will return an Illegal Quantity error if you exceed the limit.

Make certain that the value you supply for characters-per-line is greater than the length of the longest word in the string. If, for example, you specify a line length of 11, and there's a 15-letter word in the string, Parser will go berserk and fall into an endless printing loop.

## 80-COLUMN PRINTERS

With a couple of modifications, Parser can be used to send output to a printer. Use the statement:



POKE SA+73,32:POKE SA+48,81

where, again, SA is Parser's starting address in memory. This statement provides a maximum left-margin plus line-length total of 79, instead of the 39-character screen limit.

Open a file to the printer and redirect output there with the CMD statement before calling Parser; then close the output channel when you're finished. The statement:

POKE SA+73,29:POKE SA+48,41

will reset Parser for screen printing.

---

RUN it right: **C-64; one or two joysticks**

# Dodge 'Em

*By Tony Brantner*

EVER HAVE DAYS WHEN EVERYONE on the road seems to be looking for an accident? Your defensive driving skills get lots of practice, don't they? In Ram Rod, those skills are put to the test as you try to remove the dotted lines from the middle of the road while dodging a computer controlled crash car. One or two people can play at a choice of three difficulty levels.

The game screen shows four concentric rings of highway, each marked with a dotted line. Breaks in the rings enable you to drive from one road into another. The red car, which is controlled by a joystick in port 2, belongs to player 1; the green car in the two-player version belongs to player 2 and is controlled by a joystick in port 1; the computer's car is white. Running scores are displayed in the center of the screen.

At the start of each game, you can adjust the difficulty level, displayed at the top-left of the screen, by pressing F1. To change the number of players, shown at the top-right of the screen, press F3. F5 starts the action.

A game lasts five rounds, with each player having one turn per round. When you press the firebutton to begin your turn, your car starts traveling counterclockwise around the course, while the white

crash car moves clockwise. Since you have no brakes, you must rely on acceleration and steering to avoid the crash car and burn up dots. Use the firebutton as a gas pedal, and, when you reach an opening, use the joystick to change lanes.

You receive ten points for each dot you erase, and your turn ends when you clear all the dots from the course or collide with the computer's car.

---

RUN it right: **C-64; color monitor**

# The Palette-able 64

*By Richard Penn*

IF YOU THOUGHT your C-64 was limited to 16 colors, get ready for a surprise. Palette 64 produces an extravaganza of 128 distinct hues, the built-in 16 plus 112 more mixtures!

A demonstration program operating in Hi-Res mode, Palette 64 offers three screen displays that are accessed with the function keys. Press F1 or F3 to see a large or small palette of the 128 colors. Each palette is a matrix, with the 16 built-in colors running horizontally and vertically and a different hue filling the cell at each intersection. Press F5 to see an assortment of color scales, or progressions, and F7 to exit the program. The color-scale display also appears first when you run the program.

The 112 "mixed" colors are created through a "dithering" technique, in which pixels of two different colors alternate rapidly in a checkerboard pattern to produce the appearance of a third color. Dithering is usually done in Multicolor mode, but in Hi-Res mode it produces smoother color blending because of the smaller pixels. If you have some knowledge of programming in Basic, an examination of this program can show you how the dithering technique is accomplished.

When you run Palette 64, it will change the way you think about color on the C-64!

# Bull's-Eye!

By Tony Brantner

IF YOU'RE AIMING FOR a two-player game that's fun and easy to learn, then you're right on target with Bull's-Eye, a computerized dart game for the C-64. A single joystick, plugged into port 2, is shared by both players. If you'd like to practice by yourself, you just take both players' turns.

The dart board is divided into 12 radial sectors, each worth one to 12 points. If you hit one of the rings that cross the sectors, you'll compound your sector score. The yellow-and-green outer ring doubles the score; the yellow-and-black inner ring triples it. The red bull's-eye is worth 50 points, while the black ring that immediately surrounds it is worth 25. You get nothing for hitting the black outer edge.

Throwers alternate turns, with each turn consisting of three throws. The dart is always moving on the diagonal, but you can relocate it to some extent with the joystick lever. A little practice will give you the feel of the dart's path. When it's a little below your aiming point, release it by pressing the firebutton. At the instant you "throw" it, the dart moves straight *up* about an inch on the screen before the hit—depicted by a little cross—is registered.

To win, you must be the first player to accumulate *exactly* 301 points. The scores for both players, initially 301, are displayed in boxes to the right and left of the board. As play proceeds, both scores decrease. To win, you must make a final hit that brings your score precisely to zero. If you go below zero, you "break," and your score reverts to what it was at the start of your last turn.

# Escape Hatch 128

*By Richard Curcio*

ONE OF THE BEST features of the C-128 is its screen editor. Editor Swap offers significant enhancements to the screen editor and its windowing capabilities by adding a new command—Escape £. Editor Swap is a Basic loader that pokes the Editor Swap machine language into memory. When run, the loader displays the range of memory the machine language occupies.

Editor Swap gives the C-128 an additional escape function, executed by pressing the escape key and then the £ key (Escape £), or by including `PRINT CHR$(27) "£"` in a 128-mode program. This function is similar to Escape X, which toggles the C-128 between 40- and 80-Column modes.

However, this command swaps editor values without switching display modes. It allows you to jump between windows in 40-Column mode, switch to 80 columns and do the same thing, and then return to 40 columns and find the windows still intact. To disable the function, press the run-stop/restore key combination.

## THE DEMO

After you run the demo program, it adjusts itself to 40 or 80 columns and assigns the escape-key code, `CHR$(27)`, plus £ to variable E\$. Then it creates a window in the left half of the screen, prints E\$, changes the character color and opens a window in the right half of the screen. Next, it jumps from the right side to the left, listing itself where the previous list operation ended. The Sleep command slows things down for observation. Note that Escape L is printed in the left window and Escape M in the right, so the left window scrolls while the right does not.

The demo also creates upper and lower windows in different colors, flashing and underlining in 80-Column mode, and prints inside the windows some numbers that show these attributes. Again, one window scrolls while the other doesn't.

In 80-Column mode, entering `CHR$(27)"R"` (Escape R) reverses

the entire screen, and clearing the windows with a different color cursor creates two different backgrounds. Underlined and flashing characters are available *only* in 80 columns.

If you wish to use a character other than £, enter POKE ES + 24, followed by the CHR\$ value of the character you want to use. Appendix E in your *C-128 System Guide* lists the CHR\$ codes. If you want to use one that's not listed, enter PRINT ASC (x), where x is the desired character.

---

RUN it right: **C-64 or C-128, with the appropriate GEOS disks**

# DA-Runner

*By Jim Collette*

JUST AS YOU ARE READY to paste a few pictures in geoPublish, you realize that you forgot to copy Photo Manager onto your geoPublish disk. You waste time searching for the Photo Manager, and finally waste more time copying it to the work disk.

DA-Runner to the rescue! With this short desk accessory (DA), you can load any *other* DA you want from the first 50 DAs on any disk drive. As an added bonus, color is turned off while the DA runs.

I had implemented this concept in my GEOS Font Editor 2.5, which always lets you pick DAs from different drives, and it worked so well that I developed DA-Runner so it could be employed in other programs.

Place a GEOS-formatted disk containing the application program you will be using in the drive and run the DA-Runner program. Be sure to make it one of the first seven DAs on the disk, so that it will appear when you load a desk accessory within a program.

Then, whenever you need to run a DA, simply select DA-Runner instead of the desk accessory you want to load. When DA-Runner executes, it displays a list of the first 50 DAs on the current disk. The Disk and Drive buttons allow you to change disks and drives, so you can insert various disks. Once you locate the desired desk accessory, click its filename and select the Open icon. Your desk accessory will then be loaded.



When the desk accessory you're using is done, you then need to restore control to the original application. If you removed the disk that DA-Runner was on, you'll be prompted to re-insert it into the appropriate drive. When you do so, control will be returned to the original GEOS program you were using. Also, don't remove a disk before quitting a desk accessory, for that will usually crash the system.

---

RUN it right: **C-64; one or two joysticks**

# Power-Punch

*By Tony Brantner*

**STRAP ON THE GLOVES** and slug it out with a friend or one of three computer-controlled fighters on your C-64. When playing against the computer, plug your joystick into port 1; when playing another person, plug joysticks into both ports 1 and 2.

At the first screen, use the joystick in port 1 to select from the four opponents: Player 2, another person; 64-Spar, the weakest computerized boxer; 64-Pro, a stiff challenger; and 64-Champ, a true fighting machine. Move the joystick forward or back to change the flashing selection and press the firebutton to specify your choice. Then settle into your front-row seat.

The fighter in the yellow trunks is controlled by the joystick in port 1; his opponent, in the green trunks, is controlled by either the computer or the joystick plugged into port 2.

Once the bell sounds, you can move your fighter forward and back (left and right on the screen) by pushing the joystick in the corresponding direction, then score points and weaken your opponent by landing four kinds of punches. Hold the firebutton down and move the joystick left for a jab, down for a body punch, right for a cross or up for an uppercut. You can also defend against your opponent's punches. Without pressing the firebutton, push the joystick up to cover or down to duck. Just remember that if you cover, you're open to a body punch and if you duck, you might take an uppercut.

Barring a knockout, a fight lasts for three rounds. The fighters'





fatigue levels are indicated by the colors in the "fatigue boxes" located below the scores. Each box starts out green, then changes to light green, cyan (blue-green), light blue, blue, purple and red, as the fighter tires. Once a box turns red, the fighter is in danger of being knocked out. The boxers do regain energy while fighting and get fully rested between rounds.

Each punch a fighter lands earns points, and the running scores are displayed at the top of the screen. You win by either scoring a knockout or accumulating the highest score. Press the firebutton to set up a rematch.

---

RUN it right: **C-128 (in 80-Column mode); 1571 or 1581 disk drive; printer optional**

# Ledger Check

*By William Wright*

TO USE LEDGER CHECK, just enter the value of each transaction, then press the return key on the regular keyboard or the enter key on the numeric keypad. Other than the starting balance entry, positive entries will be listed as deposits and negative entries as withdrawals.

Ledger Check has four menus: Screen menu, for moving around in the ledger and manipulating entries; Disk menu, providing five common disk functions; Print menu; and Quit menu. The function keys that activate the menus are listed at the top of the ledger screen. After accessing a menu, you can scroll through the options by pressing the cursor-up, cursor-down or zero key. Then press the return or enter key at the option you want to choose. To leave an option, press any menu function key; to leave a menu, press the function key that activated it. The cursor-right key will also exit a menu if no option requiring input is active.

The best way to familiarize yourself with Ledger Check is to type in some entries and try all the options in each menu. Note that it's best to set the withdrawal number at the beginning of the program, so the program has a reference number upon returning to the start.



I use a Panasonic KX-P3131 daisywheel printer with Ledger Check. To use a different printer, you may have to modify the printer portions of the program in lines 100–140 and lines 1550–1820.

---

RUN it right: **C-64**

# ALU 64

*By Eric Bryant*

GEARED TOWARD BOTH fledgling and advanced assembly language programmers, ALU (Assembly Language Utility) 64 can also aid Basic programmers by giving them a better understanding of the inner workings of the computer.

ALU's main menu offers four submenus: Conversions, for converting decimal numbers to hexadecimal and binary; Change Memory, for poking values and messages into memory; Examine Memory, for displaying blocks of memory in either decimal, hexadecimal, binary or ASCII form; and Exit Utility, for leaving ALU.

Whenever you want to escape a submenu or any ALU function and return to the main menu, just press F1. In fact, a good rule is: When in doubt, press F1.

## CONVERSIONS

The Conversions submenu offers four options with which to convert decimal numbers to hexadecimal or binary, and vice versa. Press the number, 1–4, of the desired function and enter the value to be converted. Note that hexadecimal numbers are not preceded by a dollar sign. You may want to first experiment with options 1 (decimal to hex) and 3 (decimal to binary) if you're not totally comfortable with hexadecimal and binary numbers.

## CHANGE MEMORY

Using the two options in the Change Memory submenu, you can alter the contents of specified memory locations:

Change Locations lets you poke numbers into a block of memory by just answering the starting address prompt.



Insert Message lets you enter a message that the computer will translate into machine code and poke into a memory location. Exercise care when assigning a starting address, so the message won't overwrite important code; memory locations 828-1023 (\$0376-\$03FF) and 49152-53247 (\$C000-\$CFFF) are relatively safe. The message can be displayed in either ASCII or its screen code equivalent, used when displaying a character on-screen.

## EXAMINE MEMORY

The two options in the Examine Memory submenu let you view the contents of the computer's memory for analysis:

Examine Locations lets you see the values stored in specified memory locations in either decimal, hexadecimal, binary or ASCII form. (ASCII is chosen in the second menu that automatically appears.) Enter the starting address and the type of output wanted, and the values will scroll by. Any key except F1 pauses the listing.

Examine Bits shows which of the eight bits in a byte are on and which are off. To see how it works, make sure the screen text is in uppercase (press shift/Commodore to toggle the case), then enter 53272 for the memory location to examine. The resulting display has two components: the value of the byte stored in that location and the number sequence 0-7, representing the eight bits in the byte. Toggle (shift/Commodore) the display to lowercase and enter 53272 again to notice a change in the value of the byte.

## EXIT UTILITY

The Exit Utility submenu lets you exit ALU 64. Choose option 1 to return to a machine language monitor or option 2 for Basic.

---

RUN it right: **C-128 (in 40-Column mode); joystick**

# Pegs

*By Leonard Morris*

PEGS IS A TRIO of logical C-128 games that can entertain children as well as adults. I've called the games Linearpegs, Interpegeion and

**Transpegion.** A joystick in port 2 is required for each one.

**Linearpegs** is straightforward, literally—the rectangular game board sports a row of nine holes. Initially, four red pegs occupy the four holes on the right, while four blue pegs occupy the four holes on the left. The object of the game is to swap the red pegs with the blue ones, moving the red pegs to the left and the blue pegs to the right. In a turn, you can move one peg into the next hole or over another peg into the hole beyond. To make a move, position the white pointer over the desired peg and press the firebutton.

In cross-shaped **Interpegion**, the board contains 17 holes, four in each arm and one in the middle, and the last two holes in each arm hold a pair of like-colored pegs. The object is to move each pair of pegs from its initial position at the end of one arm to the end of the opposite arm—without backing up. This may sound easy, but I assure you it's very difficult.

To move the blinking peg into the next hole or over another peg into the hole beyond, just press the firebutton. Your next move will then be in the pair that's 90 degrees clockwise. You can change the peg that's blinking in a pair by moving the joystick in any direction. When you can't make a move, the game is over.

The object of **Transpegion** is to make all eight pegs in the octagonal game board red, after starting with an assortment of red, green, blue and yellow pegs. Sounds easy, too? Well, the hitch is that you can't change individual pegs; you must change a whole row of pegs at once. Since each row starts out containing different colors, and all pegs cycle through the same color sequence, it makes your task of turning all eight pegs red that much harder. Cycle the colors of the blinking row by pressing the firebutton, and change the row that's blinking by moving the joystick in any direction.

In all three Pegs games, pressing the space bar at any time will stop play and return the game selection screen. ■

# **RE** **== RUN**

EDITOR-IN-CHIEF  
**DENNIS BRISSON**

TECHNICAL MANAGER  
**TIM WALSH**

MANAGING EDITOR  
**SWAIN PRATT**

SENIOR EDITOR  
**BETH S. JALA**

ASSOCIATE EDITOR  
**HAROLD R. BJORNSEN**

COPY EDITOR  
**PEG LePAGE**

ART DIRECTOR  
**HOWARD G. HAPP**

DESIGN AND LAYOUT  
**ANN DILLON**

TYPESETTING  
**DEBRA DAVIES**  
**KEN SUTCLIFFE**

FULFILLMENT CONSULTANT  
**DEBBIE BOURGAULT**



[www.Commodore.ca](http://www.Commodore.ca)

May Not Reprint Without Permission

# 11 Programs Included on this Disk

## *From the May RUN:*

- ▶ RUN Copy 128
- ▶ Basic Takes the Wrap
- ▶ Dodge 'Em

## *From the June/July RUN:*

- ▶ The Palette-able 64
- ▶ Bull's-Eye!
- ▶ Escape Hatch 128
- ▶ DA-Runner

## *Plus: Extra Bonus Programs!*

- ▶ Power-Punch
- ▶ Ledger Check
- ▶ ALU 64
- ▶ Pegs

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:

**RUN Special Products • 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 1990 by IDG Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1990 IDG Communications/Peterborough

