

ENRICH  
OHAUS

COMMODORE 64 FOR THE BEGINNING BEGINNER

Russ/Foley  
EN79226

**C=64**<sup>TM</sup>

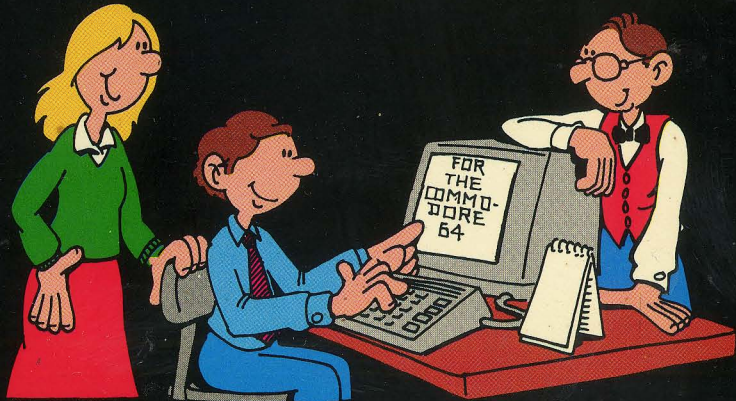
**C=64**

**COMMODORE 64**

**C=64**

**C=64**

**FOR THE BEGINNING  
BEGINNER**



**ENRICH**  
THE GOOD IDEA PEOPLE  
INSTRUCTIONAL  
MATERIALS  
DIVISION OF  
**OHAUS**<sup>®</sup>

ANOTHER GOOD IDEA BOOK FROM



# STATEMENTS & COMMANDS

**DIRECTORY**—Shows a list of all programs on a disk.

**END**—Ends program execution.

**FOR . . . NEXT**—Marks beginning and end of a program loop.

**GOSUB . . . RETURN**—Transfers the program to the specified subroutine then returns to next statement after GOSUB.

**GOTO**—Transfers the program to the specified line.

**IF . . . THEN**—Tests a conditional expression.

**INPUT**—Inputs information from keyboard.

**LET**—Assigns value to variable.

**LIST**—Lists program lines on the monitor.

**LOAD**—Loads BASIC program file from cassette tape or disk.

**NEW**—Erases program from memory.

**POKE**—Places specified value into memory address.

**PRINT**—Prints an item or list of items on the display at current cursor position.

**REM**—Remark: instructs computer to ignore rest of line.

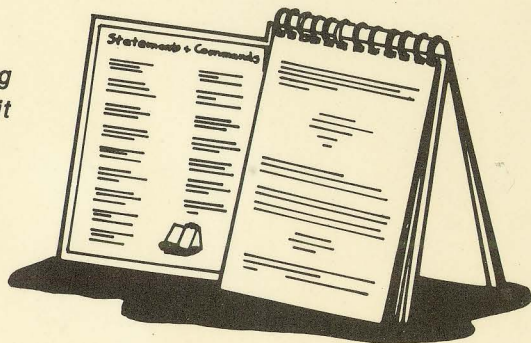
**RUN**—Starts execution of current program, or a portion of it.

**SAVE**—Stores current program on cassette tape or disk.

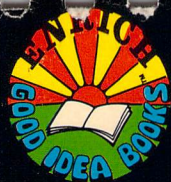
**SCRATCH**—Eliminates a program file from a disk.

**VERIFY**—Makes sure a program has been SAVED accurately, either on tape or disk.

*This book's unique binding enables the user to prop it up by the computer when in use. Note illustration.*







**C=64** <sup>T.M.</sup>

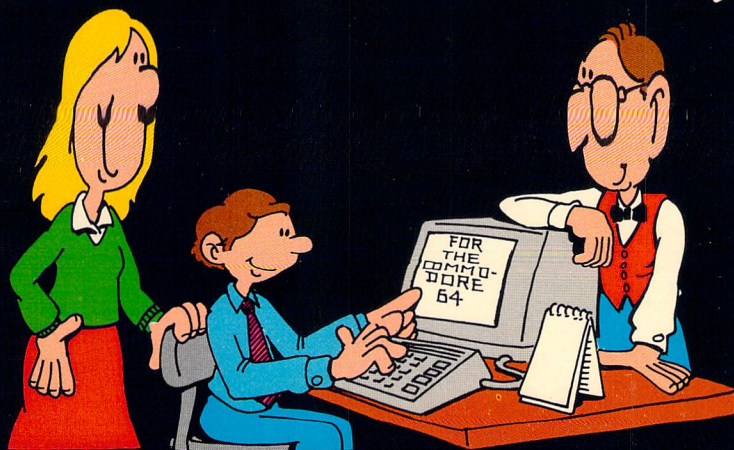
**C=64**

**COMMODORE 64**

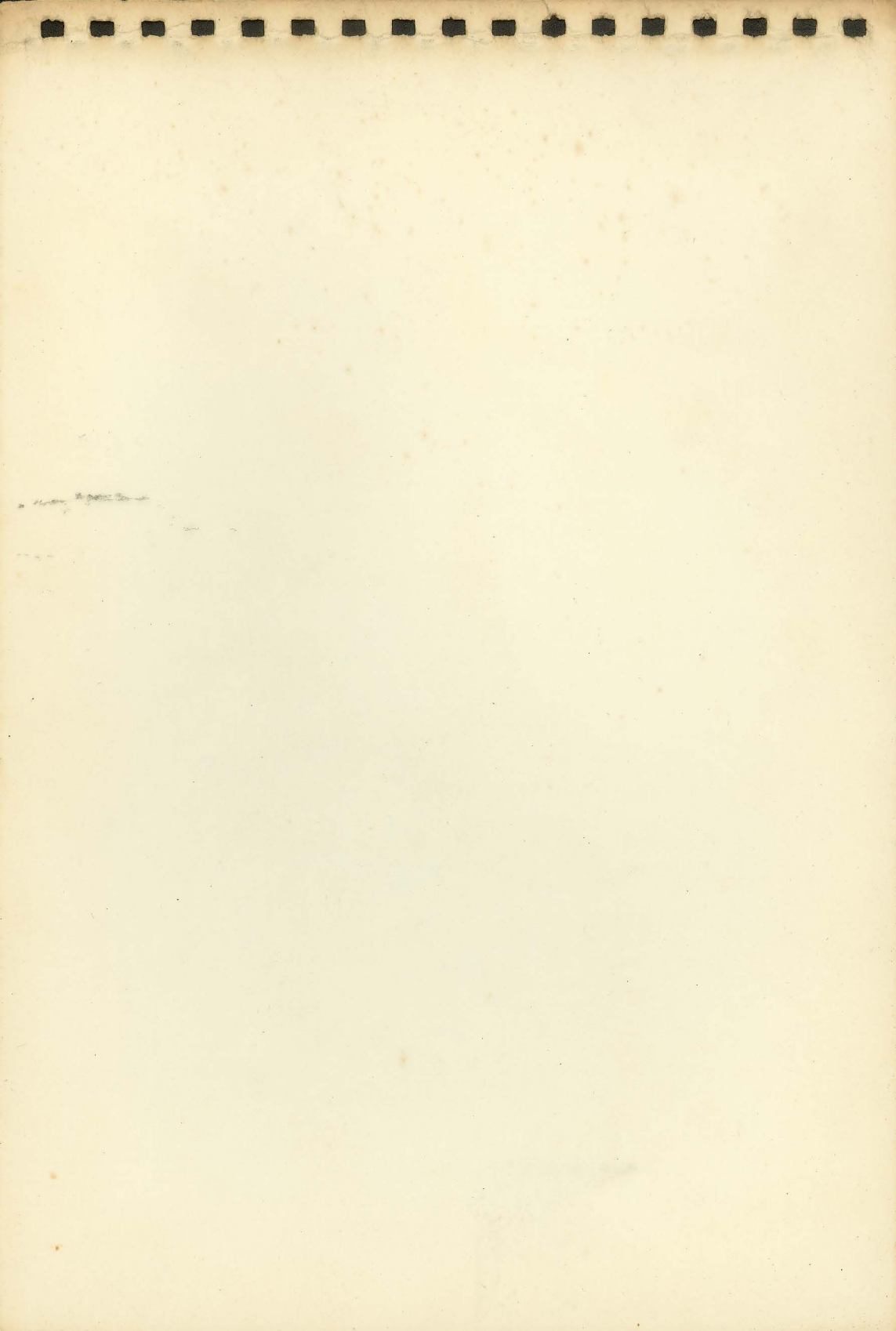
**C=64**

**C=64**

**FOR THE BEGINNING  
BEGINNER**



**ANOTHER GOOD IDEA BOOK FROM** **ENRICH**  
THE GOOD IDEA PEOPLE  
INSTRUCTIONAL MATERIALS DIVISION OF **OHAIUS**®





**C64  
FOR THE  
BEGINNING BEGINNER**

---

*An easy and helpful introduction  
to computers and programming*

**Flora Russ  
Matthew J. Foley**

ENRICH/OHAUS  
San Jose, California

Graphic Design by Kaye Graphics

Editorial Direction by Contemporary Ideas, Inc.  
Edited by John Deubert, Keith Marshall and Jim Hugaard

Typography by KGN Graphics

Published by  
ENRICH/OHAUS  
2325 Paragon Drive  
San Jose, CA 95131

For information on rights and distribution outside the U.S.A., please write ENRICH/OHAUS at the above address.

Copyright © 1984, ENRICH DIV./OHAUS. All rights reserved under International Convention. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.


Commodore 64 is a trademark of Commodore Business Machines

ISBN: 0-86582-162-3  
Catalog No. EN79226  
Printed in the United States of America

10 9 8 7 6 5 4 3 2 1



# TABLE OF CONTENTS

	<i>Page</i>
INTRODUCTION .....	4
CHAPTER 1—MEETING YOUR COMMODORE 64 .....	5
Introduces you to the computer and keyboard. Starts with the on/off switch, the keys, colors, cursor controls, and how to draw pictures on the screen. Also shows you how to give commands the computer will obey.	
CHAPTER 2—WORDS AND PICTURES .....	31
More commands the Commodore 64 will do for you. Changing the screen background color and border color. Doing arithmetic and controlling printout.	
CHAPTER 3—REVVING UP YOUR COMMODORE 64 .....	47
Shows you how to use the Commodore disk drive, cassette player, and printer. Discusses plug-in cartridges, joysticks, the care and feeding of magnetic “floppy disks,” and the disk drive’s “DOS Support Program.”	
CHAPTER 4—PROGRAMMING .....	79
Introduces putting several commands together for the computer to do in order. How to change programs, and plan pictures with the graphics grid.	
Shows you how to use variables to store numbers and words, make the computer ask questions and remember answers, count numbers, repeat things, and pause.	
CHAPTER 5—LIGHTS, SOUND, ACTION! .....	120
Example programs showing the use of random numbers, decision making, animation, advanced graphics, and music synthesis. Ends with a discussion of how to continue learning more about your computer.	
APPENDIX .....	161
1. Background/Foreground Color Combination Chart	
2. Color Chart using CTRL or  Keys	
3. Color Chart Using POKE	
4. Key to Special Symbols	
5. Screen POKE Codes	
6. Designing Pictures for the Commodore 64	
7. Connecting the TV	
GLOSSARY .....	171

# INTRODUCTION

Welcome to the world of computers! This book is about the Commodore 64 computer. It is designed for beginning computer users of all ages who want to know something about computers in general and the Commodore 64 computer in particular. Without being a programming book, these pages “show off” many of the C64’s qualities in an easy going, friendly manner.

This book will guide you through a hands-on experience beginning with how to turn the Commodore 64 on and how to use the keyboard and TV screen. You’ll find pages of guided practice on how to communicate with the Commodore 64. There are numerous programs to type into the computer that demonstrate its ability to handle text (letters) and arithmetic (numbers). Many of the programs feature Commodore’s graphics capabilities (patterns and pictures).

This book provides a starting place. We suggest you bring a sense of adventure and play to the experience. We hope you will “play” with the Commodore 64 as you work through the book. Play “what if” games. “What if I change this number?” “What if I add another line here?” Let concerns of “right” and “wrong” go. Discovery is the goal.

By the time you finish this book, expect to feel very comfortable using the Commodore computer equipment. You’ll discover that most of what you have learned is applicable to other microcomputers as well. You will be familiar with much of the BASIC computer language vocabulary and be able to perform some modifications on existing programs. Also you will have a good start on learning to program a computer.

When you want to type on the computer, you may prop up this book on its cover next to your Commodore 64.

Let’s go!



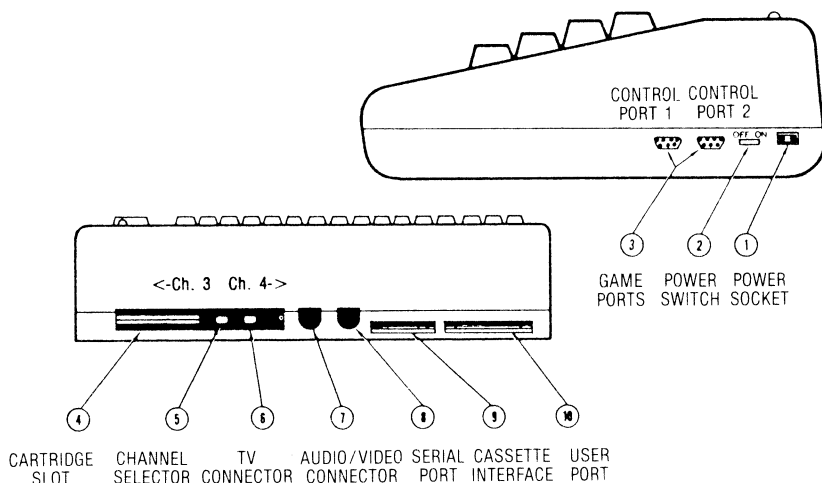
# MEETING YOUR COMMODORE

In this first chapter, we'll introduce you to your Commodore 64, show you how to hook it up, and help you learn to use the keyboard and TV screen.



The Commodore comes in several pieces, but it's easy to hook them all up. First find the big piece: the computer itself (pictured above). It looks like nothing but a typewriter keyboard, right? But inside is a computer Albert Einstein would have loved to own!

If you look on the right-hand side of the computer, you will find several plugs and a little switch marked "ON." If you turn the C64 clear around, you'll find seven more plugs and slots and sockets of different sizes. For now we are only concerned with the two or three needed to get your Commodore running!



## Getting Turned ON

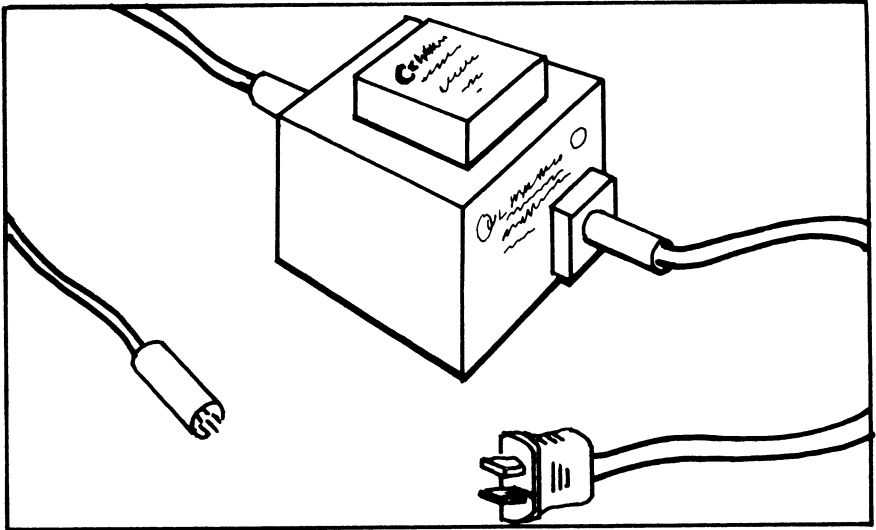
Your computer must be hooked up to a television or monitor. Put the TV and computer near each other so they'll be easy to work with. If your Commodore is already attached to your TV, you may continue with this section. If you want some help connecting your Commodore 64 to your TV or monitor, please turn to the appendix on page 161.

Before we continue setting up your C64, make sure there are no special game cartridges attached to the back of the computer. We will tell you all about how to use cartridges, disk drives, and tape players in Chapter Three. In order to give yourself the best introduction to the Commodore 64, you should first see how it behaves without all the extras.

Take a look at the right-hand side of the C64. The ON/OFF switch is there. The *Commodore User's Guide* calls this a "rocker switch" because it rocks back and forth. Push on the side away from the word "ON," to make sure the C64 is "off." Now you are ready to attach the power supply for your Commodore.



Always make sure the computer is turned *OFF* before plugging or unplugging anything on your computer.



The power supply is the small heavy box that came with your computer. It has two cords coming out of it. One cord has an ordinary wall plug on it. The other has a curious round plug with many small pins inside it. This fits into the socket marked “POWER” on the right-hand side of the C64. Connect the cords to the computer and to an electrical outlet. We’re almost done!

Now turn the computer ON. A little red light at the top of the keyboard should start shining to let you know the power is on. If it does not, take a look at the trouble-shooting chart on page 9 of your *Commodore User’s Guide* for some help in fixing the problem.

If you are using a “monitor” TV screen, turn it on and you should see a picture. However, if you are using a regular television set, you must now decide what channel to use on the TV. You have a choice of either channel 3 or channel 4. Choose the one which is *not* used in your town. If both channels have a station, choose the one with the weakest signal.

Tune your TV to that channel, make sure the antenna switch-box is set to "COMPUTER," and turn the TV set on. Set the volume at a normal level. You might see a clear picture, or you might have nothing but static on your screen.

The computer must also be set to use the proper TV channel. If you were lucky, the C64 may already be set to the channel, 3 or 4, which you chose to use. If so, you should now be seeing a picture on your TV. If not, take another look at the back of your computer. Next to the TV connector is a "channel selector switch." This is a tiny sliding switch in a slot on the back of the C64. Use a screwdriver (not a pencil or anything breakable) to push this switch from its present position to the other side. Now a picture should appear on your screen. If it does not, read the trouble-shooting chart on page 9 of the *Commodore User's Guide*, and check the connections and settings of your computer and TV.

Adjust the tuning and color of the TV so you get a good picture. If the picture stays fuzzy, you can try the following:

- move the power supply as far away from the TV as convenient,
- make sure the wires attached to the TV antenna screws are firmly attached,
- try using the other channel, 3 or 4, on the TV,
- try using the other position of the channel selector switch on the back of the Commodore 64,
- look at the other suggestions on pages 9 and 10 of the *Commodore User's Guide*,
- get help from your dealer.

## **“I’m READY.”**

Turn the C64 on, if you haven’t done so already, and let’s take a look at the message it’s displaying.

We will assume throughout this book that you have a color TV. Most black & white TV’s will work all right, but the display will not be as interesting.

The TV screen should have a dark background with a light blue frame around it and some light blue printing at the top. The first line says

```
*** COMMODORE 64 BASIC V2 ***
```

This just announces that the machine is a Commodore 64 (you knew that!), and that it uses the language called BASIC Version 2, to which this book will introduce you. Commodore is proud of the fascinating little computer they’ve built!

Next the C64 says

```
64K RAM SYSTEM 38911 BASIC BYTES FREE
```

The C64 is telling you how much memory it has (64K, or about 64,000 “bytes”), and how much of that memory is FREE right now (38,911 “bytes”). What’s a byte? A byte is a little piece of memory big enough to store one typed letter or number. Your computer, when it is first turned on, has enough free memory to store a program 38,911 letters long! That’s almost twenty pages of typed material! Don’t worry, you will start out with short programs at first.

The most important part of the message is the third line:

**READY.**


Whenever you see the word **READY**, it means the Commodore 64 is listening to you, and is **READY** to do what you tell it. No computer can do anything by itself. A person has to tell it what to do. The Commodore 64 looks up to you for instructions, and right now it is saying it is **READY** to obey you.

## The Keyboard

Let's get started!

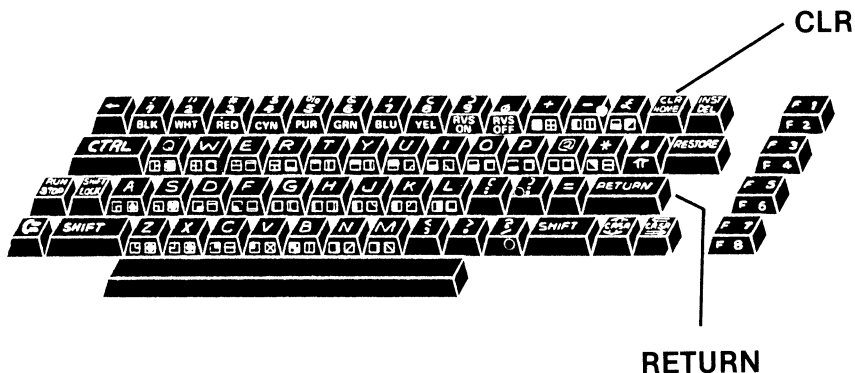
The C64 has a keyboard that looks very much like a typewriter keyboard. Press a few letters, or try typing your name. Notice that when you type a letter, that letter appears on the TV screen. Pretty convenient!

Do you see a blinking blue square? This little "blinky" is called the cursor (pronounced "ker-ser"). The blinking cursor shows where the next letter is going to be typed on the screen. Try typing the letter **K**. See that the "K" appears where the cursor was, and the cursor moves one space to the right. This works for any letter.

If you get a funny little symbol that looks like this: , or some other shape which is not a letter, then you have a problem which is very easy to fix. Find the **SHIFT LOCK** key on the left side of the keyboard and press it once firmly. This will "un-LOCK" the key so that the C64 prints letters when you press letter keys. Pressing the **SHIFT LOCK** key once more will "lock" it again. You can change back and forth between letters and shapes this way. We will discuss the shapes in a few pages. For now, make the C64 print letters.



So far the C64 hasn't done anything except show what you've typed. The computer never does anything else until you press a special key labeled **RETURN**. Look at the picture below, then find the **RETURN** key.



Type a few more letters

**GHJKL**

Now press **RETURN**, and what happens?

Oops! The Commodore says

**?SYNTAX ERROR  
READY .**



Don't worry! This is just Commodore's way of telling you it doesn't understand. Well, GHJKL isn't very understandable!

You can tell the C64 doesn't mind, though, because it says "READY" again.

This is the way the Commodore always behaves. If you tell it something it understands, it will try to obey you. If it can't understand, it will say "?SYNTAX ERROR" and then report "READY" again.

You can get rid of the "ERROR" message and the other stuff on the screen by holding down the **SHIFT** key and, while still holding it, press the **CLR/HOME** key once. Try it.

The "CLR" on the **CLR/HOME** key stands for "clear," and that's what this key does. It clears up the whole screen and you get to start over again! Notice that the cursor also goes up to the top left corner of the screen. This is the cursor's "home." Whenever you clear the screen, the cursor will "go home." If you ever want the cursor to go home **WITHOUT** clearing the screen, you can make it do that too. Just press **CLR/HOME** **WITHOUT** holding down the **SHIFT** key. But to clear the screen, you must hold down the **SHIFT** key while pressing **CLR/HOME**.

*REMEMBER:* It is difficult to hurt the computer! It will try to do what you want. But if it can't, it will just give you an ERROR message and say "READY." So go ahead and experiment! It's fun to try different things. If something doesn't work, just try again. And you can always "CLR" (clear) the screen.

## Using Colorful Language

The Commodore 64 can use all the colors of the rainbow. To print different colors on the screen, you will need to find the **CTRL** key. Look at the picture below, then find the **CTRL** key on your keyboard.






The **CTRL** key cannot cause any color changes by itself. You must hold it down *while pressing another key*. To start with, how would you like to make the C64 print in green?

























Hold down the **CTRL** key and press the number **6** key. Immediately, the cursor changes from light blue to bright green! Try typing a few letters. They print in green. (If this doesn't work at first, press **RETURN** and try again.)

Now go ahead and press **RETURN**. You'll probably get another **ERROR** message, and a "READY" message, also in green. The computer seems to really like its new color!

Printed on the front of each number key, 1 through 8, is the name of a color. To get a particular color, just hold down the **CTRL** key and press that number. For example, to get yellow printing, hold down **CTRL** and press the **8**. The cursor changes color, and now everything you type will appear on the screen in yellow. Experiment with the different colors, and see what choices you have.

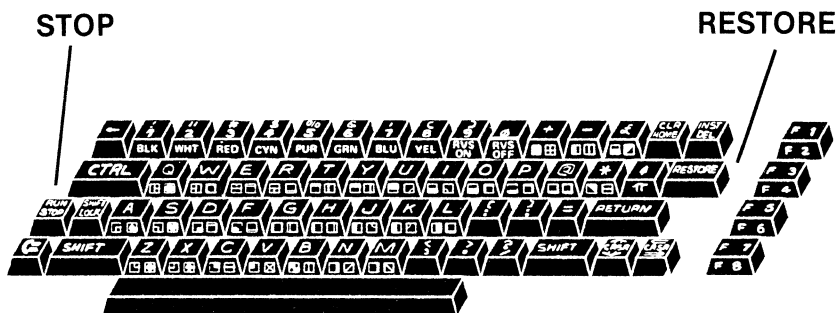
There is another special key similar to the CTRL key. At the bottom left of the keyboard is a key marked ™. This is called the "Commodore key" because it is marked with Commodore's own symbol. This key does not do anything by itself. You must hold it down while pressing another key.

If you hold down the  key and press one of the number keys, the color of the cursor will change again. This gives you eight more colors to play with. The sixteen colors you can get by using the CTRL key or the  key with the number keys 1 through 8 are:

KEYBOARD	COLOR	DISPLAY	KEYBOARD	COLOR	DISPLAY
<b>CTRL 1</b>	BLACK		 <b>1</b>	ORANGE	
<b>CTRL 2</b>	WHITE		 <b>2</b>	BROWN	
<b>CTRL 3</b>	RED		 <b>3</b>	LT. RED	
<b>CTRL 4</b>	CYAN		 <b>4</b>	GRAY 1	
<b>CTRL 5</b>	PURPLE		 <b>5</b>	GRAY 2	
<b>CTRL 6</b>	GREEN		 <b>6</b>	LT. GREEN	
<b>CTRL 7</b>	BLUE		 <b>7</b>	LT. BLUE	
<b>CTRL 8</b>	YELLOW		 <b>8</b>	GRAY 3	

If you make the cursor the same color as the background, you will discover something very odd. It looks like the cursor has completely disappeared, and you are typing in invisible ink! This is because the background of the screen is the same color as the letters. It's like trying to see a black cat at midnight, or a white rabbit in a snowstorm! The cursor and letters are there, but they disappear against the background.

If you would like to get back to normal after all this experimenting, there is an easy way. You can make the Commodore do a "reset." To reset your C64, hold down the **STOP** key (on the left side of the keyboard) with one hand, and press the **RESTORE** key (on the other side of the keyboard) with your other hand.



Press these keys at the same time to reset your Commodore 64.

The TV screen should suddenly go blank, and the word "READY," in the normal light blue color should appear in the upper left corner of the screen. The C64 will now print in normal light blue until you change the color again with the CTRL or the **C** keys.



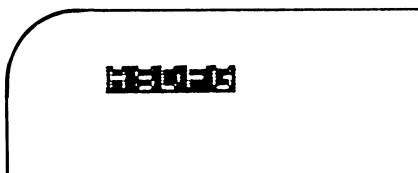
## Going Into ReVerSe

You probably noticed that while changing colors, we only used the number keys 1 through 8. What about 9 and 0? If you look at the front of the **9** key, it is labeled “RVS ON.” The **0** key says “RVS OFF.” RVS stands for “reverse,” and now we’ll see what that means.

Hold down the **CTRL** key and press **9**. The cursor doesn’t seem to change. However, if you let go of the CTRL key and type some letters, you’ll see there’s been a big change in the way letters are printed on the screen. Normally the C64 prints with ordinary letters on a plain background.

**ASDFG**

After you press **RVS ON**, letters print in reverse colors!



There are two ways to turn off this reverse printing. First, you may press **RETURN**. This will always turn off “RVS.” (You’ll probably also get an ERROR message.) A second way to stop “RVS” is to hold down the **CTRL** key and press **0** (zero). This gives you **RVS OFF**, and the C64 will print normally again.

Try typing this line:



**RVS ON** PLEASE **RVS OFF**  
LET ME USE THE **RVS ON** C64 **RVS OFF** !


You can also change colors in the middle of a sentence. Try playing with the colors and the “Reverse” function to make pretty and interesting sentences. Perhaps there is a new form of poetry ready to be started here!


Remember that you can erase the screen just by holding **SHIFT** and pressing **CLR**. If you want to reset everything on the computer back to normal, you can press **STOP** and **RESTORE** together.

## Up and Down

There is one more thing about your keyboard which you should know before starting to use the C64.

In most cases the  key (Commodore key) cannot do anything by itself. You must hold it down while pressing another key in order for anything to happen. But there is one thing the  key can do which is rather unusual.

Make sure there is some typing displayed on the TV screen. Press down on the  key and the **SHIFT** key together, once. You do not need to touch any other keys. The letters that were upper-case should now be displayed as lower-case. If you had any of the little picture symbols on the screen, they will be displayed now as upper-case letters.

If you again press the  and **SHIFT** key down at the same time, the screen should go back to the way it was. You can go back and forth between these displays as often as you like. When the screen is showing only upper-case letters and picture symbols, it is in the “graphics mode,” because the screen is showing the little pictures. (“Graphics” means “pictures.”) When the screen is showing both upper-case and lower-case letters, and maybe some picture symbols, it is in the “text mode,” because ordinary writing has both upper-case and lower-case letters in it. (“Text” means “writing.”)

For most of the exercises in this book, you will want to leave the screen in the graphics mode, so that letters are displayed in upper-case, and picture symbols are available. This is the way they are after a “reset,” after pressing **STOP** and **RESTORE** together. Do not use only the **SHIFT** or **SHIFT LOCK** key to get all upper-case! It may look the same, but the effect on the computer is very different.

## Getting the Commodore 64 to Understand

So far, every time you press **RETURN**, the C64 gives you a ?SYNTAX ERROR message. You already know that this is because the computer is not able to understand your meaning. How can we tell the Commodore something it *will* understand?

The C64 does not speak English or Spanish. Instead, it understands a special language called Commodore BASIC. This language is specially made to be easy to use.

Have you ever trained a dog? When you give “instructions” to a dog, you use special command words, like SIT, STAY, or ROLL OVER. To give instructions to the C64 computer, you also use special command words. But the C64 is much more well-behaved than any animal. It always does exactly what it’s told!

The rest of this book will tell you about the command words the C64 understands, and how to use them.

## The PRINT Command

The first command you will learn to give the C64 is PRINT. Type the line below exactly as it looks. If you make a mistake, just press **RETURN** and start over again. Type:

```
PRINT "HELLO FRIEND"
```

Now press the **RETURN** key. (Remember, the computer never does anything until you press **RETURN**.) The TV screen should look like this:

(What you typed) **PRINT "HELLO FRIEND"**

(What C64 printed) **HELLO FRIEND  
READY.**



If you got an **ERROR** message or a "0" on the screen, check to make sure you spelled **PRINT** correctly and used quotation marks before **HELLO** and after **FRIEND**.

As you can see, the computer obeyed your **PRINT** command. In fact, it printed so quickly, you probably didn't notice it was doing anything! It **PRINTed** the message you told it to print.

Try:

**PRINT "JOE SMITH WAS HERE" RETURN**

The C64 should print **JOE SMITH WAS HERE  
READY.**



The C64 is able to obey a **PRINT** command, because the word **PRINT** is a command in the **BASIC** language. The computer will print anything between the quote marks after a **PRINT** command, whether it makes any sense or not. It will obey you exactly! And it will **PRINT** so quickly it seems to take almost no time at all!

Try some **PRINT** commands of your own. Be sure to

1. type **PRINT** first,
2. put what you want the computer to print in quotation marks,
3. press **RETURN** at the end of the command.

Go ahead and try it on your own!



There are several other commands in the BASIC language. They are all easy to use, but if you use them cleverly you can make the computer do some very interesting things. Already you can make it print anything you want.

Before you learn more commands, you need to know about some special keys on the keyboard.

## Fixing A Mistake

Suppose you want to give the command

```
PRINT "OOPS"
```

but you accidentally type

```
PRINT "OOPS"
```

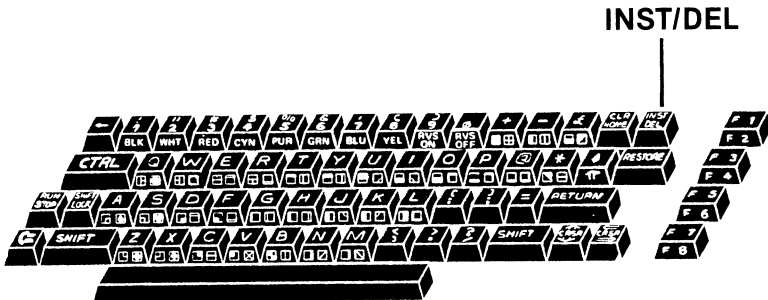
The word **PRINT** is misspelled. If you press **RETURN** you will get a “?SYNTAX ERROR” message. *If you did not push RETURN here is how you can correct it:*

Your screen should have:

The cursor is still sitting here because you have not pressed **RETURN** yet.

```
PRINT "OOPS"█
```

Now find the special key marked **INST/DEL**.



The “DEL” on this key stands for DELETE, which means “ERASE.” Press the **DEL** key and notice what happens. The cursor moves backward one space and deletes a character! Try it.

Now comes the tricky part: Press the **DEL** key and hold it down gently. The cursor waits a second, then starts running backward as fast as it can go, gobbling characters as it moves!

This is fun to watch, too. If you keep holding the **DEL** key down, the cursor will move clear to the beginning of the line and jump up to the end of the line above it. If there is anything on that line, you will probably get an ERROR message when you press **RETURN**. Try it.

Well, that was fun, but let’s see how to use the **DEL** key to fix mistakes instead of just erase them. Once again, type

```
PRINT "OOPS"■
```

and don’t press **RETURN**.

Now press the **DEL** key until you move the cursor back to here:

```
PRI■
```

Now type the rest of the correct command:

```
PRINT "OOPS"■
```

If you make another mistake, just use the **DEL** key.

Press **RETURN** and see that the C64 obeys your corrected command.

You have already learned how to use one other special key, the **CLR** key. Since your screen is probably full of stuff, let's use it again. Hold down the **SHIFT** key and press **CLR**. The whole screen should become "clear."

## SHIFTing Around

You have probably noticed that the **CLR** key and **DEL** key both have *two* words on them. The **CLR** key has CLR ("clear") on the top, and HOME ("home") on the bottom. To get the top function, CLR, you have to hold the **SHIFT** key down while pressing **CLR**. To get the bottom function, HOME, you don't use **SHIFT**. In the same way, the **DEL** key has INST on the top, and DEL on the bottom. So far, all we've used is the DEL ("delete") function. Since DEL is printed on the bottom half of the DEL key, we *don't* use **SHIFT**. If you do hold down **SHIFT** and press the **DEL** key, you will get the INST function, which stands for "insert." This opens up space so you can insert letters into the middle of a word or line.

## SHIFT and ⇧ with Letters

**NOTE:** If your TV screen is showing any lower-case letters, please change your computer to upper-case mode, by pressing down on **SHIFT** and ⇧ together, once, so that ordinary unshifted letters are displayed as upper-case. Otherwise, this whole section won't work!

Another simple way to get into upper-case mode is to use "reset" by pressing **STOP** and **RESTORE** together.

The **SHIFT** key on the C64 does not work quite the same way it does on a typewriter. On a typewriter you normally get lower-case print, and if you shift you get upper-case. Well, with the C64 in upper-case mode, your *normally* get upper-case. If you use the **SHIFT** key you get something out of this world!

Try this. Type some letters:

ASDFGHJ

Now hold down the **SHIFT** key and type the same letters. You get:

▲♥-| | ^



That's pretty interesting. Now instead of **SHIFT**, hold down the **⌘** key while typing the same letters:



▣ ■ | | |

What are those funny little symbols good for?

These special symbols are called “graphics characters” and you can use them to draw pictures. Before moving on to the next section, type out all the different graphics characters you can get from the Commodore keyboard, using the **SHIFT** and **⌘** keys.



As you have probably noticed, the graphics characters are shown on the fronts of the keys on the Commodore keyboard. The  key gives you the left graphic character, and the SHIFT keys gives you the right graphic character. This is easy to remember, because the  key is on the left and the SHIFT key is to the right.

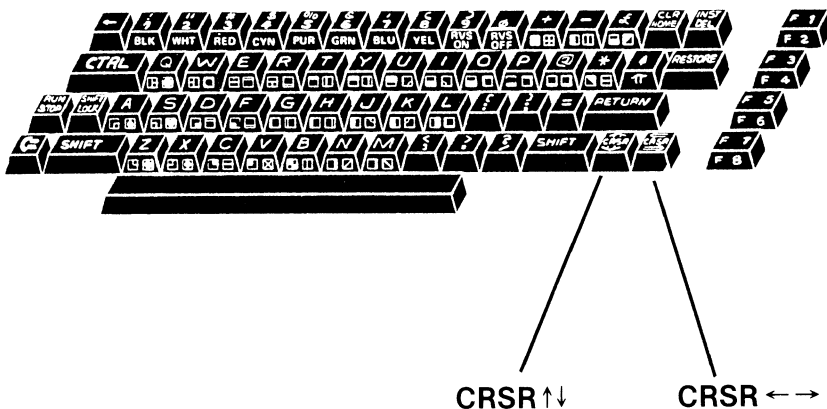
Notice that if you accidentally press the SHIFT and  keys at the same time, you will change the screen display mode. In upper-case (“graphics”) mode, unSHIFTed keys are upper-case, while SHIFTed keys are graphics characters. In lower-case (“text”) mode, unSHIFTed keys are lower-case and SHIFTed keys are just upper-case. The graphics characters you get while holding down the  key stay the same in either upper or lower-case modes.

## Drawing Pretty Pictures

The graphics characters are printed on the fronts of the keys on the C64. Take a look at them to see what they will look like on the screen. You can choose from these little shapes and pictures to create your own drawings.

Before you start drawing with the C64, let’s CLEAR the screen. Use the **CLR** key (with **SHIFT**), and your “canvas” will be completely erased.

Now, suppose you want to put a diamond right in the middle of the screen. Right away there is a problem: how do you get to the center of the screen? The C64 has some more special keys, which let you move the cursor anywhere on the screen. These are called “cursor control keys,” and they are marked **CRSR**↑↓ and **CRSR**←→.



Press the RETURN key to put the cursor on a new line.

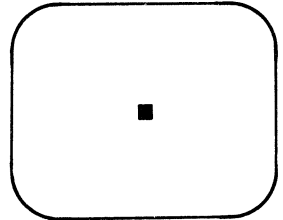
The cursor control keys each have two functions, like the **DEL** and **CLR** keys. Let's start with the **CRSR** key. The symbol on the bottom half of this key is an arrow pointing down: ↓. Press this key a couple times and you'll see that each time the cursor moves down one space. This key is called "CURSOR DOWN." Now press the key and gently hold it down. The cursor will run all the way down to the bottom of the screen, as though you had pressed it many times. Remember that the **DEL** key worked the same way. This is called "auto-repeat" because the key repeats automatically when you hold it down. All of the cursor control keys have auto-repeat. Of course, you can get the same result by pressing the key many times in a row.

Now, if you hold the **SHIFT** key down and press **CRSR** key, you should get the function on the top half of the **CRSR** key. This will be ↑. Go ahead and hold down the **SHIFT** key and press the **CRSR** key once. The cursor moves up. Press it again, and the cursor moves up another line. This is called "CURSOR UP." If you hold the CURSOR UP key it will auto-repeat, too.

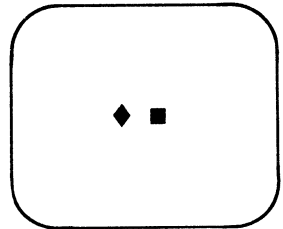
Use the CURSOR UP and CURSOR DOWN to put the cursor about half-way down the side of the screen.

What about the other cursor control key, the one marked **CRSR**  $\leftrightarrow$ ? Try it. If you press the **CRSR**  $\leftrightarrow$  by itself, you get **CURSOR RIGHT**. If you hold the **SHIFT** key down while pressing it, it becomes **CURSOR LEFT**. Both these keys also will auto-repeat. Notice what happens if you try to go too far right or left.

Our original goal was to put a diamond in the middle of the screen. Now that you know how to use the cursor controls to move the cursor anywhere you want, move it to the center of the screen.



The next character you type will go where the cursor is. The “diamond” shape is a **SHIFT Z**, so hold down **SHIFT** and type a **Z**.



Suppose you want a *red* diamond next to it. Change the cursor to **RED** by holding down **CTRL** and pressing **3**. Now type another diamond. It should be red!

If you change your mind and want a flower there instead, you can use the **DEL** key to erase the diamond and print a **SHIFT X**, which looks something like a small flower. Go ahead and do this. Add a few more flowers of different colors to make a whole garden.

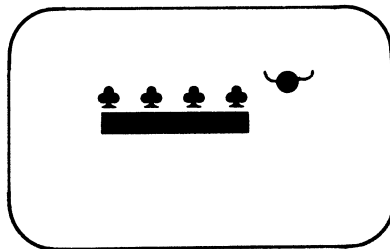
Would you like to add some earth for the flowers to stand in? Move down one line with a **CURSOR DOWN**, then type several **CURSOR LEFT**s. Now we are going to make a brown bar under the “flowers”, underlining them with “earth.”

How do we get the color brown? The eight colors you can get with the **CTRL** key are black, white, red, cyan, purple, green blue and yellow. But remember, there is another set of colors you can get with the **☞** key. (Take a look at page 13 to see what the other colors are.) Hold down **☞** and press **2**. The cursor should turn **BROWN**.

Now, how do we get a solid bar of color? Turn the **RVS ON**, by holding down **CTRL** and typing **9**. Now type several spaces with the space bar. The cursor should leave behind a trail of solid brown. Put brown "earth" under all the flowers. Then turn the **RVS OFF** with **CTRL 0**.


As a last touch, let's add a "bird" to the picture. Move **UP** several spaces, change the cursor to **WHITE (CTRL 2)**, then hold the **SHIFT** key down and type **J Q K**.

You should now have a picture something like this:



By combining the color and graphics characters of the C64, Commodore has made it very easy to draw pictures on your computer. Our little drawing was fairly simple, but it does show some of the things you can do.

## More Special Keys

We have seen most of the keys on the keyboard just print a letter or number. If you hold the **SHIFT** or  keys, they print different “graphics characters.” There are also some special keys. You already know how to use some of them:

**SHIFT** **CLR**

CLEAR

**CLR**

HOME

**DEL**

DELETE (erase)

**RETURN**

RETURN

**SHIFT**

SHIFT

**CURSOR/** 

Cursor DOWN

**SHIFT** **CURSOR/** 

Cursor UP

**CURSOR/** 

Cursor RIGHT

**SHIFT** **CURSOR/** 

Cursor LEFT

**STOP** **RESTORE**

RESET

**CTRL** number

colors

 number

more colors

 **SHIFT**

change mode

**RVS ON**

Reverse ON

**RVS OFF**

Reverse OFF

There are a few other special keys on the board.

One key that is useful for making lots of graphics characters is the **SHIFT LOCK** key. If you press it once, it “locks” and makes the Commodore 64 think you are holding the **SHIFT** key down. Now, all the letter keys will print their graphics characters. Watch out, though! When the **SHIFT LOCK** key is locked you

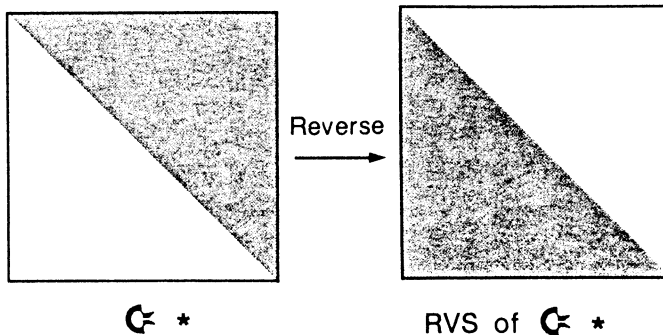
can only use CURSOR UP and CURSOR LEFT. To use CURSOR DOWN or CURSOR RIGHT, you have to unlock the SHIFT. The DELETE key also requires that the **SHIFT LOCK** key *not* be locked.

To unlock the **SHIFT LOCK**, just press the key. It will release.

You can also use RVS with graphics characters. A flower on a dark background (**SHIFT X: ♣.**) will be a black flower on a bright background if you turn on REVERSE first. The circle (**SHIFT Q: ●**) will become a hole: ◻.

Another use for RVS is to get solid bars of color. When you press the space bar, you normally get a completely empty space. If you turn on REVERSE and then type some spaces, you get a completely filled-in space. The filled-in spaces will be the same color as the cursor. If you want a green bar, type **GRN (CTRL 6)**, then **RVS ON (CTRL 9)**, then as many spaces as you want.

There are some graphics characters you can only get by using RVS. For instance, turn off the RVS by typing **RVS OFF (CTRL 0)**. Now type **⌘ \***. The \* key is on the second row of keys, near the RESTORE key. This will print a ▼ symbol. How would you print a ▲ symbol? Look at all the keys, and you see will that the ▲ symbol isn't on any of them. But ▲ is just the REVERSE of ▼! If you press **RVS ON** and then type **⌘ \*** you will get ▲. Try it.



The only other special keys are **STOP** and the INST function on the **DEL** key. There is also one more special function of the **CTRL** key. You'll find out how to use these in the coming chapters.

You may want to play with the C64 for a while before going on. Try drawing your own pictures. Put shapes and lines together to make a house or a person on the TV screen. You have already learned a lot about your Commodore!

## Summary

You've learned quite a bit in this chapter. You know that the Commodore can follow certain commands, and you've learned one of these commands: PRINT.

You've also learned a lot about the keyboard, how to fix mistakes, and how to move the cursor around the screen. You can change the colors the computer uses to print on the screen. You are aware of the graphics characters the C64 has and how to make pictures with them.

Play with these new skills for a while before moving on. You may also want to take a break after all the typing you've been doing! Explore the different things you can do with the graphics characters and the PRINT command. Remember, the C64 can't be hurt by typing on it; and it's fun to see what it will do when you try different things. When you're "READY" for more, we'll continue.

## Vocabulary

cursor  
auto-repeat  
READY  
PRINT “anything”  
graphics characters  
colors  
reset

**RETURN**

**DEL**

**CRSR ↑↓**

**CRSR ←→**

**CLR/HOME**

**SHIFT**

**SHIFT LOCK**

**⌂**

**CTRL**

**RVS ON**

**RVS OFF**

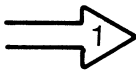
**STOP**

**RESTORE**



# WORDS & PICTURES

Before you go on, let's introduce you to what the next chapters will look like.



Information for you to read, such as instructions and explanations, will be in normal print. This paragraph is an example of instructions to be read.



Instructions for you to type into the Commodore's keyboard are indented and usually printed in upper case letters, like this:

```
PRINT "THIS IS A DEMONSTRATION"  
PRINT "OF INSTRUCTIONS TO BE"  
PRINT "TYPED INTO THE COMMODORE 64"
```



Graphics symbols which appear on the screen are shown in a table when these symbols are used.

SYMBOL	MEANS
⌘	SHIFT CLR
#	SHIFT 3
⌘	⌘ 3
⌘	CTRL 3
♦	SHIFT S
⌘	⌘ S



This is a final reminder to press the **RETURN** key after typing each instruction. In the example at arrow 2 above, press the **RETURN** key after typing in the first line. Press **RETURN** again after the second line and once more after the third line.

## A Rainbow of Color

In this section, you will see the 16 different colors available on the Commodore 64. You have already seen how to change the color of the printing. You can also use any of these colors for the background or the border of the screen.

In the following instructions, the word **POKE** followed by the number 53281 tells the computer to do something with the color of the screen. The number following the comma is a code for one of the colors the Commodore 64 knows. The colon (:) permits you to put more than one instruction on a line.

Type each line and press the **RETURN** key. Then stop a moment to see the result before typing the next line. Remember that when the screen is changed to the same color as the cursor, you cannot see what you type; just type the next line, but very carefully.

```
POKE 53281,1: PRINT "WHITE"  
POKE 53281,2: PRINT "RED"  
POKE 53281,3: PRINT "CYAN"  
POKE 53281,4: PRINT "PURPLE"  
POKE 53281,5: PRINT "GREEN"  
POKE 53281,6: PRINT "BLUE"  
POKE 53281,7: PRINT "YELLOW"  
POKE 53281,8: PRINT "ORANGE"  
POKE 53281,9: PRINT "BROWN"  
POKE 53281,10: PRINT "LIGHT RED"  
POKE 53281,11: PRINT "GRAY 1"  
POKE 53281,12: PRINT "GRAY 2"  
POKE 53281,13: PRINT "LIGHT GREEN"  
POKE 53281,14: PRINT "LIGHT BLUE"  
POKE 53281,15: PRINT "GRAY 3"  
POKE 53281,0: PRINT "BLACK"
```


To get your screen back to its original condition, press **STOP RESTORE**. Now that you have seen the different background colors, you may choose whichever one you like best.

## Things to Try:

- If you use 53280, instead of 53281, in the lines above you will change the color of the border. For example: POKE 53280, 7 will change the border to yellow.
- Experiment to find a combination of colors that you like for the screen and the border. Use the chart in Appendix 1 to help you. Then you can always set the Commodore 64 for these colors immediately after you turn it on.

## Color Your Words

Now that you know how to change the color of the background and the border of the Commodore 64, let's concentrate on the "foreground." The foreground color is the color of the cursor as well as the color of the printed characters. Everything you type will remain one particular color until you change it.

The foreground can also be any of 16 different colors. You get eight colors by pressing the CTRL key with the keys numbered 1 through 8. The other eight colors are obtained by using the Commodore  key with the keys numbered 1 through 8.

The simplest way to change the color of the foreground is directly from the keyboard.





The cursor is changed to yellow when you press:

**CTRL 8**

and changed to light gray when you press:

 **8**

Usually though, it is necessary to change the cursor color from within a PRINT command or a POKE command. Try typing the PRINT commands below.

SYMBOL	MEANS
	CTRL 3
	CTRL 6
	↶ 3
	↶ 6

After you type the first quote, hold down the **CTRL** key and then press the **3** key. The Commodore 64 gives you a strange symbol. Check the chart above to make sure that is the symbol you should get. Then, type the word RED and another quote before you press the RETURN key.

```
PRINT "▬RED"
PRINT "▮GREEN"
PRINT "⊗LIGHT RED"
PRINT "■LIGHT GREEN"
```

*Things to Try:*

- Continue to try all 16 possible foreground colors with your favorite background color.
- You can also control the color of the foreground by typing POKE 646 followed by a comma and a number 0 through 15. For example:

```
POKE 646,1: PRINT "WINTER MEANS SNOW FOLLOWED
BY"
POKE 646,5: PRINT "SPRING AND NEW LIFE"
```

Again, remember that if you make the foreground the same color as the background, the letters will be invisible.

## Clearing the Screen

You already know that when the **CLR/HOME** key is pressed, the cursor moves to the home position, the upper left corner of the screen. When you use the **SHIFT** key with the **CLR/HOME** key, any characters on the screen will disappear as the cursor moves to the home position. Remember:

To clear the screen directly from the keyboard, hold down the **SHIFT** key and press **CLR/HOME**.

### **SHIFT CLR**

You can also clear the screen inside a **PRINT** command. Whenever you type **PRINT** and then a quote ("**␣**"), the Commodore 64 notices that you typed the quote. After that, it won't obey the **CLR/HOME** right away. Instead, it waits until you push the **RETURN** key.

SYMBOL	MEANS
<b>␣</b>	<b>SHIFT CLR</b>

Try this line:

**PRINT "␣"**

Notice the table above. The **␣** is NOT a graphics character. It is the **CLR** symbol which has been "stored" in the **PRINT** command. You will often use this form to clear the screen, throughout this book.

## Exploring the Graphics Keys

When you turn on the Commodore 64, you are in graphics mode. In this mode, the screen can be used to display words and pictures.


We've seen that each letter key has two additional symbols on the front. These are called graphics characters. These characters can also be used in PRINT statements.

Type:

```
PRINT "I ADORE MY 64"
```

Now, hold down the SHIFT key while typing the portion above enclosed in quotes. It should look like this:


```
PRINT "\ ⬆ ⬇ ⬈ ⬉ \ | & $"
```


This time hold down the Commodore (  ) key while typing the same line again. This time it should look like this:

```
PRINT "⬆ ⬇ ⬈ ⬉ ⬈ ⬉"
```

The first example above uses text characters or letters. The other two examples use the graphics characters printed on the fronts of the keys.

### *Things to Notice:*

- The graphics symbol on the right front of a key is printed when you use the SHIFT key. The symbol on the left front is printed when you use the Commodore (  ) key.






- When you use the 6 key and the 4 key with the SHIFT key, the characters located above the numbers are printed.
- When you use the 6 key and the 4 key with the Commodore  key, special symbols appear on the screen. When the Commodore 64 tries to PRINT these special symbols, the color of the cursor changes.

Remember that you can combine the graphics characters to make up larger pictures.

## Printing Letters on the Commodore Screen

You have used features of the Commodore 64 which allow you to display all the characters found on the keyboard in sixteen different colors. In addition, you can place these symbols anywhere you wish on the screen by using the cursor (CRSR) keys, and you can highlight them by reversing the background color and the foreground color.

Use this chart to help find the correct symbol to use.

SYMBOL	MEANS
	SHIFT CLR
	CTRL 9 (RVS ON)
	CTRL 0 (RVS OFF)
	CRSR↓
	CRSR→

Let's start with a blank screen.

PRINT "□"

Then type:

```
PRINT "HI THERE"
```

You can highlight a message on the screen by using the RVS-ON key to reverse the background color and the foreground color.



Press **CTRL 9** here.

```
PRINT "HOW ARE YOU?"
```

Press the CRSR→ key 10 times inside the quotes to move the words 10 spaces from the left margin.

```
PRINT "          FINE, THANK YOU."
```

Press the CRSR↓ key 10 times to move the cursor down 10 lines before printing.

```
PRINT "          GLAD TO HEAR THAT."
```

Now, it's time to start:


```
PRINT "PUTTING IT ALL TOGETHER"
```



## Writing with Upper and Lower Case Letters

When you are in graphics mode, all text that you type into the Commodore 64 appears on the screen in capital letters. There are two ways to transform the Commodore 64 so that you can get text in both upper and lower case. To look at them, you first need some text on the screen:


```
PRINT "THIS IS UPPER CASE"
```

You already know the first way to change the display mode. Press the Commodore (  ) key and while holding it down, press the SHIFT key. All the text on your screen should change to lower case letters. Now, type:

```
print "this is lower case"
```

When the Commodore 64 shows lower case letters, you will need to hold the SHIFT key whenever you press a letter that you want to be in upper case. (Note: When you are in lower case the word PRINT must also be in lower case.)

```
print "This is UPPER and lower case."
```

You can make letters that are already on the screen alternate between upper and lower case by using the Commodore (  ) key with the SHIFT key together. Try it a few times and watch the screen. Notice that capital letters in text mode change to graphics characters.

POKE can also be used to change from upper to lower case. Type:

```
poke 53272,21  
PRINT "I LOVE MY COMPUTER"
```

Now try another POKE so the Commodore 64 can go from upper to lower case:

```
POKE 53272,23  
print "I love my Commodore 64"
```

When you are through, return to graphics mode by typing:

```
poke 53272,21
```

## Strings and Numbers

Your Commodore 64 can work with two kinds of information: words and numbers.

Any number, letter or graphics symbol on the keyboard is called a character. A "string" is any group of characters enclosed in quotation marks. For example:

```
PRINT "COMMODORE 64 CAN WORK WITH WORDS AND  
      NUMBERS"  
PRINT "THIS IS A STRING"  
PRINT "2"  
PRINT "3+2"  
PRINT "◆◆◆◆"  
PRINT "ALL OF THESE EXAMPLES ARE STRINGS"
```

Your Commodore 64 treats words and numbers differently. You can tell the following are numbers because there are no quotation marks.

```
PRINT 2
PRINT 3+2
```

In the line above,  $3 + 2$  is not really a number; it is two numbers with a plus sign between them. This is called a “numerical expression” because the Commodore 64 first must do some arithmetic with the numbers.

You can combine strings with numbers or numerical expressions.

```
PRINT "3 + 2 =" 3+2
PRINT "COMMODORE IS A PERFECT" 10
PRINT "A SMILE IS WORTH A $" 1000*1000
```

Try a few of your own.

*Things to Notice:*

- In a PRINT statement, words must be enclosed in quotation marks.
- In a PRINT statement, numbers can be in or out of quotes and follow these rules:
  - Numbers between quotes are printed exactly as they look.
  - Numbers not in quotes are treated as numerical expressions. Numerical expressions are computed before being printed.

## Using the Commodore 64 to do Arithmetic

When the following arithmetic problems are typed in, the Commodore 64 will immediately do each calculation and print the answer. As you go through this page, experiment with math problems of your own.

Use + to add numbers:      **PRINT 3 + 6**  
                                  **PRINT 123 + 14**  
                                  **PRINT 10.96 + 5.04**

Use - to subtract numbers: **PRINT 12 - 8**  
                                  **PRINT 865 - 421**  
                                  **PRINT 2000 - 1984**

Use \* to multiply numbers: **PRINT 5 \* 7**  
                                  **PRINT 36 \* 87**  
                                  **PRINT 43216 \* 22**

Use / to divide numbers:    **PRINT 24/2**  
                                  **PRINT 521/99**  
                                  **PRINT 3746.8/646**  
                                  **PRINT 2/3**

The Commodore 64 gives you one more decimal place than most hand-held calculators and rounds off the last decimal place in the answer.

*Things to Try:*

- **PRINT 8 + ((3 \* 4)/6) \* (15 + 3)**

An expression can be written using parentheses. The arithmetic in parentheses is done before the arithmetic outside of parentheses.

- **PRINT 8 + 3 \* 4/6 \* 15 + 3**

If an expression is written without parentheses, the Commodore 64 does multiplication and division first, and then does addition and subtraction.

- **PRINT 1000 \* 1000 \* 1000**

Scientists express very large and very small numbers by "scientific notation." The Commodore 64 uses a similar system called "E notation."

1E + 09 means  $(1 \times 10^9)$  which is 1000000000 or 1 billion.

If you have never seen scientific notation before, don't worry about it. You'll probably see it very rarely.

## Playing with Punctuation

You can control the placement of words and numbers on the Commodore's screen.

Type each line below and take a moment to see how the Commodore 64 displays the words and numbers before typing the next line. Some of the instructions use commas, and some use semicolons. Once you see how they work, experiment with some lines of your own.

```
PRINT " "
PRINT 1,2, "BUCKLE MY SHOE"
PRINT 3,4, "SHUT" , "THE" , "DOOR"
PRINT 5;6; "PICK UP STICKS"
PRINT 7;8; "LAY" ; "THEM" ; "STRAIGHT"
PRINT TAB(12); "9,10, A BIG FAT HEN"
```

### *Things to Notice:*

You have seen three ways to control the placement of text on the Commodore's screen:

1. The comma divides a line of the screen into four columns each of which holds 10 characters. The characters following a comma are printed in the next column. Sometimes the next column is on the following line.
2. The semicolon prints strings right next to each other and numbers very close together.
3. TAB moves the cursor to the space named in the parentheses before anything is printed.

## **Another Way to Give Instructions**

There are two ways to give instructions to the Commodore 64. So far, you have been using a way called "immediate mode." In immediate mode, the Commodore immediately performs each instruction, as soon as you press RETURN. In "programming mode," the Commodore 64 saves the instructions for later and performs them only when commanded to do so.

Type this:

```
PRINT "☺"  
PRINT "THIS IS IMMEDIATE MODE"  
PRINT "THINGS GET DONE IMMEDIATELY"
```

The three instructions you just typed in immediate mode can be put into programming mode by starting each line with a number.

Now type this, and don't worry if nothing seems to happen after each line.

```
NEW
10 PRINT " "
20 PRINT "THIS IS PROGRAMMING MODE"
30 PRINT "EACH LINE IS STORED IN MEMORY"
40 END
```

You have just typed your first "program." A program is a set of step-by-step instructions that tell the computer what to do.

### LIST

When you LIST this program, the instructions that are currently stored in memory are displayed on the screen.

### RUN

When you type RUN, the Commodore 64 performs the instructions in order. In the above program, line 10 clears the screen, lines 20 and 30 print what you have put inside the quotation marks, and line 40 ends the program.

It is considered a good programming practice to begin every program with PRINT " " to clear the screen, and to end every program with END.

#### *Things to Try:*

- Type LIST. Type RUN. Type LIST again. Type RUN once more. You can RUN and/or LIST your program again and again. The program stays in memory until you erase it by typing NEW or by turning off the computer.
- Type NEW and press the RETURN key to erase memory. Type LIST and press RETURN. Then, type RUN and press RETURN. Since you erased the instructions stored in memory, there is nothing to LIST or RUN.

## Summary

You don't have to completely understand everything in this chapter, but try to remember what abilities your Commodore 64 has. Feel free to go back and play with any of the examples. As you work with your computer, you will begin to use more and more of its potential—and of yours too!

In this chapter you explored a lot of the Commodore 64's abilities. You learned a new command, POKE, and saw how to use it to change the colors of the screen's background and border, and foreground characters.

You also learned how to do arithmetic on the Commodore, and how to use the PRINT instruction in new ways. By using the comma ( , ), semicolon ( ; ) and TAB with PRINT, you can control the way things look when they are printed on the screen.

Finally, you began to explore programming.

Chapter 4 will continue with more programming, but first you get to look at ways to “rev up” your Commodore.

## Vocabulary

POKE	PRINT with comma ( , ) and PRINT with semicolon ( ; )
screen background color	TAB
screen border color	immediate mode
foreground character color	programming mode
graphics mode	program
text mode	NEW
string	END
number	LIST
numeric expression	RUN
arithmetic ( +, -, *, / )	



# REVVING UP YOUR COMMODORE

Part of the fun in owning a Commodore 64 is that you can add to your computer system. By adding a tape player or disk drive, you can write long programs and save them, or use games, business programs, and education programs written by other people. Joysticks add to the fun of many games, and you can even add a printer or speech synthesizer to your computer.

In this chapter, we will discuss the various devices you can plug into your C64, and what they are good for. These extra devices are often called “peripherals.” Some of them are very inexpensive, while others cost more than your Commodore.

## Easy Plug-ins

One of the most common peripherals for the C64 are plug-in cartridges. These cartridges contain professionally written programs that take control of your computer. These cartridges can turn your C64 into a great video game, a word processor, a recipe file, or a math tutor. You can even add a new programming language, like Logo, with the proper cartridge. Most cartridges cost between \$15 and \$50.

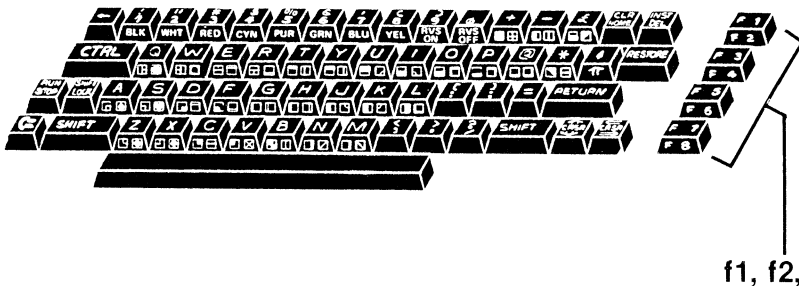
Using a plug-in cartridge is very easy. The most important thing is:

*Always make sure the computer is turned OFF before plugging in any cartridge. If you plug or unplug a cartridge with the power on, you may damage the cartridge and the computer.*

With the power turned OFF, turn the computer around and look at the back. Find the cartridge slot at the back of the computer. (Take a look at the diagram on page to make sure you have the correct slot.) The cartridge slot looks like a wide green mouth with two rows of smiling gold teeth.

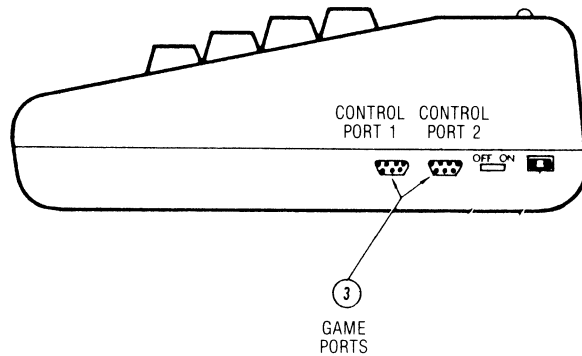
When you press a cartridge into the cartridge slot, the gold teeth make electrical contact with the circuits inside the cartridge, and your computer suddenly becomes able to do whatever the cartridge tells it. When you use a cartridge, make sure you plug it in with the label facing up! Now that the cartridge is plugged in, just turn the computer on, and it should start running the program built into the cartridge. Please be sure to read the manufacturer's instructions that come with each plug-in cartridge, to find out how to use it.

Some cartridges use the special "function keys" on the right side of the keyboard. These are the keys labeled **f1**, **f2**, **f3**, etc. If you need these keys, the instructions that came with the particular cartridge should tell you how to use them.



Two other very common, inexpensive peripherals for your Commodore 64 are joysticks and game paddles. These let you control the C64 in the same way an arcade video game is controlled, by moving a control stick or rotating a knob, instead of having to type keys. Commodore and several other manufacturers make these devices for the C64. Some Atari joysticks and game paddles are also compatible with your computer.

Joysticks or game paddles are easily plugged into the “Game Control Ports” on the right side of the computer. These “ports” are the little slots with pins sticking out of them, next to the ON/OFF switch of the C64. These are shaped like rectangles with slanted sides. And remember:



***Always make sure the computer is turned OFF before plugging in or unplugging any peripheral!***

In order to use the joystick or game paddle to control your Commodore, you must have a special program which makes the computer pay attention to these devices. Many video game cartridges require the use of a joystick. However, if you want to use joysticks in your own programs, you have to do a little advanced work. After reading this book and practicing some programming, you can learn to program the joysticks, game paddles, and special function keys, by reading the *Commodore 64 Programmer's Reference Guide*, published by Commodore.

## **External Memory: Cassettes and Disk Drives**

If you enjoy the programs in this book and want to write some of your own, one of the most useful things you can add to your computer is “external memory.” (This just means memory outside of the main computer.) You need external memory because when you turn off the Commodore, everything in its memory vanishes. Now, all the programs in this book are nice and short. You probably won’t mind typing them in every time you want to use them. But you will probably write some long programs. If you spend a whole hour typing in a long program, you won’t want it to just vanish! But you can’t leave your Commodore on forever. What can you do?

You can “SAVE” your program in external memory! External memory, besides being outside of the main computer, is permanent and does not vanish when the machine is turned off.

Commodore has two ways to SAVE a program:

1. Record it on a cassette tape using a Datasette Recorder.
2. Record it on a “floppy diskette” using a Disk Drive.

Besides saving your own programs, you can use other people’s programs. Programs saved on disk or tape can be traded, borrowed, or sold. Computer stores sell programs on disk and tape as well as in cartridge form, and some libraries and computer clubs have tapes available to loan out for free.

The next several sections discuss how to save and use programs on magnetic tape or disk. If you do not have either a Commodore cassette player or a disk drive, or if you don’t want to learn about these right now, you may skip forward and continue with the rest of this chapter on page 74.

We don't expect you to remember everything you read in these sections. Just remember enough so that you have a general idea of the things you can do with your Commodore. Then, you can come back and follow the step-by-step instructions in these pages when you need them.

## **Cassette Tape Recorder**

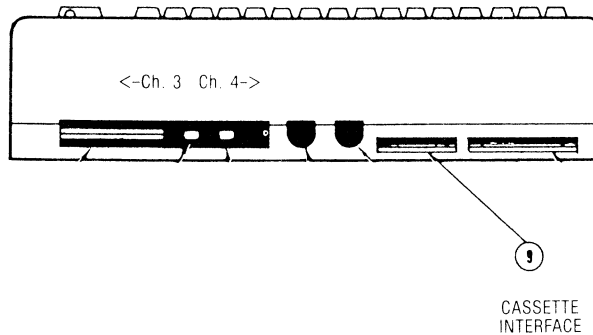
If you have a Commodore 1541 disk drive, and you are not interested in learning about the Commodore cassette tape recorder, you may skip ahead to page 57, to learn about saving and using programs on disk.

The least expensive way to buy and keep programs is on magnetic tape. You can buy pre-programmed cassette tapes from many computer stores. These tapes have programs for your C64 stored on them. You can also borrow program tapes from many public libraries and user clubs. Be sure you get only programs meant for Commodore 64's!

The tapes are made just exactly like the cassette tapes on which you record music or voices. In fact, if you have a computer tape from a store or library, try playing it on your home stereo system or regular cassette player. You will hear very fast, funny-sounding whistles and beeps. This is how the C64 stores program information on tape.

In order to use a cassette tape as permanent, external memory for your Commodore, you must have a special cassette tape recorder, called a Commodore Datasette. An ordinary cassette recorder will not work, because the C64 requires a variety of special abilities in the recorder. If you want to save your own programs, you will also need a blank cassette tape. Don't spend extra money on special "computer quality" tapes. Any reasonable quality of ordinary cassette tape, like you might use to record music, will work fine. However, short-playing tapes are most convenient. Long-playing tapes can make it difficult to "find" your programs if you store many programs on the same cassette.

The special Commodore tape player plugs into its own slot in the back of the C64, as shown below.



As we have said before, the computer must be OFF when you plug in the cassette player or any other device! Otherwise it might be damaged. The cassette player only plugs in one way. If it is hard to plug in, don't force it; try turning the plug over.

## To Save On A Cassette Tape

If you are mostly interested in using programs you borrow or buy, you may skip ahead to the next section, page 55. In this section we will learn how to save a program which you have typed into the Commodore.

If your tape recorder is not plugged into the C64, you will have to turn off the computer and connect it. Do this *before* typing in your program, or else you will have to type the program again! Always set up your recorder before you turn on the C64 and leave it plugged in while you are programming. Then when you want to save something you will be able to do so.

After you have typed a program into the computer, and are ready to save it on tape, follow these instructions carefully:

1. Place a blank tape in the recorder and close the lid.
2. Press REWIND to get to the beginning of the tape.
3. Reset your counter to 0000, by pressing the small button next to the counter numbers on the recorder.
4. Press PLAY to move the tape forward until the counter has moved 4 or 5 numbers. Then press the STOP button on the cassette recorder. You are now ready to save your program.
5. Decide what you want to name your program. The title must not be longer than 16 characters, and should not contain punctuation marks.
6. Type:

**SAVE "name"**

**RETURN**

The C64 will print:

**PRESS PLAY AND RECORD ON TAPE**

7. Press down firmly both the PLAY and RECORD buttons on your recorder and hold them down until they stay.
8. The screen will go blank while the recorder is running. When the program has been copied, the recorder will stop and READY will appear on the screen with the cursor below it. If you get an ERROR message on the screen, rewind the tape and try again. You might also make sure the recorder is plugged in firmly.

9. Use a pen to write on the paper label of your cassette. Write down the name of the program, and the counter numbers for the beginning and ending of the program on the tape.
10. It is a good idea to save all programs twice. Then you have a backup recording in case something happens to the first one. You can copy your program again on the same tape. Just move the tape forward 4 or 5 counts beyond the end of the first recording, and repeat steps 6 through 9.

Most cassette tapes are long enough to store ten or twenty programs in this way. However, if you are going to save more than one program on a single cassette, it is very important that you keep a written record of the beginning and ending counter numbers for each program. If you forget, and let the computer SAVE a program on a part of the tape that already contains a program, the old program will be erased and lost. The same thing happens if you try to record two songs on the same piece of cassette tape. The song that was there before gets erased and the new one recorded over it.

## **VERIFY a Tape**

While your program is still in the 64's memory, you may want to check the tape and make sure the program really was SAVED. To do this, you use the VERIFY command. You must type the program's title exactly as you did when SAVE-ing it.

1. Rewind the tape to the beginning of the program, then press STOP on the recorder.

2. Type:

VERIFY "name"

**RETURN**

The C64 will print:

**PRESS PLAY ON TAPE**



3. Press the **PLAY** button. The C64 will now search for the named program, find it, and verify that it was correctly saved. This takes about the same amount of time that it took to **SAVE** it in the first place.
4. If all goes well, the C64 will print the message

**OK  
READY .**

5. If all is not well, then either the C64 won't be able to find the program (give it couple of minutes, then press **STOP** on the keyboard), or it will say

**?VERIFY ERROR**

In either case, go back and record the program again.

## **To Load A Program From Cassette Tape**

Now that you've got your program stored on tape, you need to know how to get it back into the Commodore.

**SAVE**-ing a program does not erase it from the C64, but just copies it onto the tape. Likewise, recalling a program does not erase it from the tape, but just copies it into the C64's program memory.

To get a **SAVEd** program back into the C64's memory, you use the **LOAD** command. Here is how:

1. Rewind your tape, reset the cassette recorder's counter, and fast-forward (if necessary) to the place on the tape where the program is stored. Stop a few counts before the actual beginning, to make sure you don't go too far. Press the **STOP** button on the recorder.

2. Type:

```
LOAD "name" RETURN  
PRESS PLAY ON TAPE
```

The C64 will print:

```
LOAD RETURN
```

Note: If this tape contains a program borrowed or purchased from a library or store, you probably do not need to know the exact name of the program. Just rewind the tape all the way, and type the command

```
LOAD RETURN
```

and continue from there.

3. Press the PLAY button. The screen will go blank while the C64 searches for the program title. When it finds it, it prints

```
FOUND name
```

4. To actually load the program, press the **C=** key at this point. The screen will again go blank, and the computer will load the program.

5. When it finishes copying the program it prints

```
READY.
```

You may then press the STOP button to turn off the tape player. The program is now in memory, and you can start it by typing the command RUN on the keyboard.

6. If the tape player seems to run forever and the C64 cannot find your program, it may find your second copy of it. (You did make two copies didn't you?) Otherwise, you can stop it by pressing STOP on the recorder and then **STOP** on the keyboard. Then try again from step 1.

## “Floppy Disk” Drive

If you purchased a disk drive with your computer you will be saving and loading programs on disks. They are much more convenient to use than tapes, but also take a little more preparation. If you do not own a Commodore disk drive, you may still be interested in the way the disk commands work. However, if you do not wish to learn about the disk drive, you may skip to page 74.

The user's manual that comes with the Commodore 1541 is pretty difficult to read, but don't panic! The basic commands needed to SAVE and LOAD programs on floppy disks are really very simple, and we will go over them here.

## Preparing A Disk Drive

The Commodore disk drive is an oblong box about the same size as the computer itself. On the front are a couple of little lights, and a slot with a little door. On the back is a switch, a power plug, and two curious round sockets. The disk drive should come with a short connector cable and a power cord.

First of all, use the power cord to connect the disk drive to an electrical outlet. If the drive makes any noise or if the lights turn on, it means the disk drive is ON. Please turn it OFF. The drive has a “rocker switch” on the back much like the computer's. The disk drive should be OFF at this time.

On the back of the Commodore 64 are two round sockets. One is similar to the sockets on the back of the disk drive. In the diagram on page 6, it is labeled "Serial Port."

As we have said before, the computer must be OFF when you plug in the disk drive or any other device! Otherwise it might be damaged.

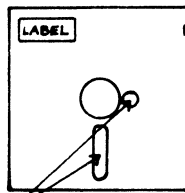
With the computer and disk drive OFF, plug the connector cable into the serial port on the back of the C64. Plug the other end into either socket on the back of the disk drive.

Now, in turning your system ON, there is a simple rule to follow: the computer is turned on LAST. Also, make sure there are no disks in the disk drive before turning it on.

Open the disk drive door by pressing the door catch gently inwards. The door will pop open, and if there is a disk inside it will be ejected. Turn on the disk drive, then turn on the computer. On the disk drive, a green and a red light will blink, and the drive will whirl and click for several seconds. When it stops, the red light will go out, and the green light should be shining steadily. Your disk drive is ready to go.

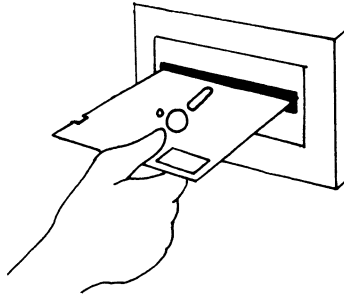
## Taking Care Of Disks

All "floppy disks" look alike. They are little square plastic devices with a disk of brown magnetic material inside them. It is very important not to bend the disk or touch the shiny surface you can see through the holes in the outer cover (see illustration).



Don't touch!

To put the disk in the disk drive, open the disk drive door by pressing gently on the door catch. It will pop upwards. Hold the diskette with the label up and towards you:



Gently slide the disk all the way into the disk drive slot. It will click into place. Gently close the door latch on the slot. If the door is hard to close, make sure the disk has been pushed in completely.

Always make sure there is no disk in the disk drive before you turn the disk drive ON. When you switch the disk drive ON or OFF, there is a little burst of magnetism which can partially erase a disk! Also, never open the disk drive door when the disk drive motor is running. This can also partially erase a disk.

## **DOS Support Program For Disk Drive**

Commodore Business Machines has produced a program which they give away for free, called the DOS Support Program. This program is also known by many other names, such as DOS Manager, C-64 DOS, DOS Wedge, Universal Wedge, EZ Wedge, or Disk Support Program.

This program is *extremely* useful! The first thing that many experienced programmers do, as soon as they turn their computers on, is LOAD and RUN the DOS Support Program. Then they have it set up and ready to use whenever they want it.

If you buy a Commodore 1541 disk drive, it should include a demonstration disk which will have the DOS Support Program stored on it. You can find its name by getting a directory of the disk:

Put the demonstration disk in your disk drive, close the door latch, and type

**LOAD "\$",8**

Then, when the computer prints READY again, type

**LIST**

The screen will show a list of all the programs on the disk. If the list goes by too quickly, hold down the **CTRL** key, and it will slow down. You can also stop it by pressing **STOP**. To repeat the listing, just type LIST again.

Once you find the name of the DOS Support Program on your disk, do the following:

Type

**LOAD "C-64 WEDGE" , 8**

(put the program's name here)

The computer should respond

**SEARCHING FOR C-64 WEDGE**

(or whatever it is called on your disk)

**LOADING  
READY.**

When it prints READY, type

**RUN**

After a few moments, the screen should show a message indicating the DOS Support Program is loaded. It will remain invisibly in your machine until the C64 is turned off. You can go ahead and use other programs, use the disk drive, or whatever else you want; you can change disks, or use the command NEW. The DOS Support Program will not interfere and will not be erased.

You do not *have* to have the DOS Support, but it is a convenience. If you do not have a demonstration disk, or if you want more information about the DOS Support Program than we can give here, contact any Commodore computer store. (You will probably need to find a store that specializes in Commodore equipment, not just a department store that sells many kinds of computers.) Most Commodore dealers will do their best to help you get the most from your Commodore equipment. They can also sell you a copy of the DOS Support Program for about \$5 or if you bring in your own disk, they will let you copy it for free.

You can also get free copies of the DOS Support Program, as well as many other free programs and help with your Commodore problems, by joining a local "Users' Group." A users' group is just a group of ordinary people who are interested in Commodore computers. They join the group to share their experiences and help each other with their computers. If you are interested in knowing more about users' groups, see the book "*Free Software for your Commodore*," by Dave and Dorothy Heller, published by ENRICH (the Good Idea People who brought you *this* book). Your Commodore dealer can also probably direct you to the nearest Commodore Users' Group.

## What To Do With Disk Errors

Before you can use the Commodore disk drive, there are several more bits of background knowledge you need. First of all, the green light is a “ready” light. It shows that power is on. The red light is more important. When the red light is on, it means the disk drive and the computer are talking to each other. Whenever you give the disk drive any command, such as SAVEing or LOADING a program, the red “busy” light will turn on. The red light has a second purpose. When an error or problem occurs, the red light will start blinking, and continue blinking until you find out what the problem is. So:

- Green light means power is on.
- Red light off means the disk drive is just sitting there waiting.
- Red light steadily on means the disk drive is busy. Never open the disk drive door, or insert or remove a disk under this condition! (NOTE: Some editions of the Commodore manual say “Never remove the disk when the *green* drive light is on!” This was a typographical error; the green light is *a/ways* on while the power is on.)
- Red light blinking means a problem has occurred. You may remove the disk if necessary, but don’t until you’ve tried some of the remedies we will discuss.

Eventually you are going to make a mistake and get a blinking red light on your disk drive. What do you do?

“When in danger  
or in doubt,  
run in circles,  
scream and shout!”



Computer errors can be a little frustrating. On the other hand, it is usually possible to figure out what went wrong.

First of all, if you have the DOS Support Program ready, it is very easy. Give the single character command

> **RETURN**

This should get an error message describing the problem. (By the way, error messages are sometimes hard to understand. If so, look at the number on the left of the error message. This is the disk error code number. Appendix B of the 1541 User's Manual, entitled "Description of DOS Error Messages," gives a paragraph of explanation for each code number.)

If you are using a regular SAVE or LOAD command, and something goes wrong, you will often get an error message printed on the screen. Try to figure out what the message is trying to tell you. If you don't get any message but the flashing red light, you'll need to try some other things:

Make sure you typed everything in the command correctly, including the name of the program, the quote marks, and comma. Even if you can't figure out what went wrong, try the command again. You might have made some error that you didn't catch. If it still won't work, try the INITIALIZE command (page 74). This will reset your disk drive, and then you can try your command again.

When you try to read the disk error message using the DOS Support Program's ">" command, or when you try to retype a correct command, the blinking red error light should go out. If it does not, you need to use the INITIALIZE command (page 74).

If by any chance the INITIALIZE command does not succeed in turning off the error light, you must open the disk drive door, remove the disk, turn the disk drive off and back on, put the disk back into the drive, and again use the INITIALIZE command.

If nothing at all seems to work, you are probably not using the command correctly. Review the examples and try to find the problem.

## Can You Record On Your Disk?

Every disk has a tiny notch on one edge, called the write-protect notch. If this notch is covered up with a piece of tape or label, then the disk becomes “write protected,” meaning you cannot record on it. The disk drive detects the covered slot, and refuses to record anything on the protected disk.

Write protection is usually used on disks containing expensive or important programs. If you try to SAVE or erase programs on a disk like this, you will get an error message.

## Is Your Disk Formatted?

Information is stored on disks in small pie-shaped segments that have been set up on the disk in advance. This “setup” is called “formatting the disk.” All disks must be formatted before they can be used with your computer. However, a disk will usually only be formatted *once* in its lifetime. When you buy a brand new, blank disk at the computer store, it needs to be formatted before you can save your own programs on it. But if your disk already has something on it (like a fancy new video game you want to play), and you re-format it, you will erase it!

Every brand of computer uses a different disk format, so you cannot use an Apple disk in your Commodore, even though they look exactly alike. Remember that formatting a disk *always* erases everything on it.

If your disk is already formatted, you may skip to the next section, page 66. To format a disk, follow these instructions:

## Formatting a Disk

1. Place the new, blank disk in your disk drive, and close the door latch.
2. If you do not have the DOS Support in your computer, type the commands below. Get all the commas and quotation marks right. And, of course, press **RETURN** after each line.

Type:                    **OPEN 15,8,15**

The C64 prints        **READY.**

Now type                **PRINT#15,"NEWØ:DISK,AA"**

(There should be no spaces in this command.)

The disk drive will turn on and click to itself rather noisily. Formatting a disk actually takes a whole minute and a half. Wait patiently until it is done. When it stops, type

**CLOSE 15**

If you have the DOS Support already in your computer, you don't have to type these three commands. Instead, just type the single command line

**>"NEWØ:DISK,AA"**

3. When the disk drive stops whirring, take a look at the front of the drive. If the red light is not on, it means everything is okay. Your disk is now formatted! You should not format this disk again or it will erase any programs you store on it!

4. If the error light on the drive is blinking, then there may be a problem with the disk. Take the disk out of the drive, turn off the C64, make sure the connector cable is firmly plugged into both the computer and the disk drive, and turn the computer back on. Go back to step 1 and try again. If it still doesn't work, try a new disk. If it still won't work, call your dealer for help.

## **Easy Disk Commands**

The easiest commands to give your disk drive are the same as the cassette tape commands: **SAVE** and **LOAD**. **SAVE** lets you permanently record your programs on disk, **LOAD** gets them back into your computer later.

The only difference between the disk and tape commands is a small addition. After typing the command, you type a comma (,) and then the device number, 8. The "8" tells your C64 that programs should be **SAVEd** on, or **LOADed** from, a disk instead of tape. The disk drive is device number 8, as far as the computer is concerned.

In order to use any of the disk commands, you have to have the disk drive hooked up, and a formatted disk in it. Always connect and turn on your disk drive before you turn on the C64, and leave it set up and ready while you are programming. Then when you want to save something you will be able to do so.

## **To Save On A Disk**

If you are mostly interested in using programs you borrow or buy, you may skip ahead to the next section, page 69. In this section we will learn how to save a program which you have typed into the Commodore.

When you have typed in a program, and are ready to save it on a formatted disk, follow these steps.

1. Decide what you want to name your program. Short titles are easier to type. Many programs can be saved on a single disk, so each program must have a different name. The title must not be more than 16 characters long, and should not contain punctuation marks.
2. Make sure there is a disk in your disk drive.

3. Type

**SAVE "name",8**

*The comma is **after** the quotation marks.* The number 8 tells the C64 to SAVE the program on *disk*.

4. The red “busy” light on the disk drive will go on, and you will hear some noises coming from the disk drive as it SAVES your program.
5. When it has finished, the disk drive will automatically stop and the “busy” light will go off. On the screen, the computer will say OK and READY.
6. If the “error” light starts blinking and keeps blinking, you have a problem. The three most common errors are
  - forgetting to close the disk drive door.
  - putting the disk in upside down.
  - trying to name a program with a name that’s already used on the disk.

Make sure the disk is right side up, close the disk drive door, then try again. Use a name for the program which has not been used on that disk before.

Remember, saving a program on a disk doesn't remove it from the computer's memory. The disk drive only copies the program. If you want to work on another program with your C64, you must first type NEW to erase the old program in the computer's memory.

By the way, here's something for when you become an expert: when you write a program, it may grow and develop over several days. You can type it in one day, SAVE it, then LOAD it back and work on it some more the next day. You will then want to SAVE your new and improved version of the program. If you try to use the same program name, you will find that the disk drive gives you an error. What can be done about this?

Well, you could use a slightly different name for each version of your program. This is the best thing to do if you want to keep old versions around. But if you want to erase the old version, and SAVE the new version under the same name, there is a simple way to do it. In step 3 above, instead of typing SAVE "name",8, type the command

```
SAVE "@0:name",8
```

The "@0:" in front of the program name (but within the quotation marks) tells the Commodore disk drive to erase the old copy of the program, and save the new copy under the same name.

The VERIFY command may be used with disk in the same way as with tape (see page 54), by specifying device number 8:

```
VERIFY "name",8
```

However, the disk drive is so dependable that verification is unnecessary if there is no error light blinking.

## Disk Directory

The biggest advantage of using a disk drive is that the computer is able to give you a list of all the programs it has stored on the disk. This list is called a "directory," and is a little like a phone directory. It tells you all the names it has stored on the disk. This is much nicer than a cassette tape. With a cassette, you have to keep a written record of the program names you use and the beginning and ending positions on the tape. The disk drive does all this bookkeeping for you.

There are two ways to make the computer give you the disk directory. If you have already set up the DOS Support Program, just type the following command with the disk in the disk drive:

**>\$**

Funny-looking command, isn't it? But this simple command will make the computer print the directory of the disk on the screen.

If you don't have the DOS Support, you can give the following pair of instructions. The problem with this method of showing the disk directory is that it erases whatever is in your Commodore 64's program memory. In order to check the directory *without* erasing your current program, you need to have the DOS Support Program.

With the disk in the disk drive,

type the command

**LOAD "\$",8**

the computer will print

**SEARCHING FOR \$  
LOADING  
READY.**

Now type the command

**LIST**

>⌘ **RETURN**

Either of these two methods will make the directory of the disk flow across the screen. If you want it to stop midway, you can press the **STOP** key. You may **LIST** it again, or command >⌘ again, if you wish. To slow down the listing, hold down the **CTRL** key after the listing has started.

Here is an example of a disk directory.

```
⌀ "PET DISK # 5" ".05 2A
8 "UNIVERSAL DOS" PRG
18 "MENU" PRG
22 "HEARTS'DESIRE" PRG
15 "GRIBBET" PRG
26 "RACETRACK" PRG
17 "METEOR" PRG
26 "KALAH" PRG
21 "ZONE X" PRG
17 "NUMBER-TOE" PRG
12 "NIM" PRG
482 BLOCKS FREE.
READY.
■
```

You cannot get a directory of a cassette tape.

If you own a printer, you can also make the computer print out a "hard copy" of the directory. Instructions for using the printer are on page 74.



## To Load a Program from a Disk

Now that you've got your program stored on disk, you need to know how to get it back into the Commodore.

SAVE-ing a program does not erase it from the C64, but just copies it onto the disk. Likewise, recalling it does not erase it from the disk, but just copies it into the C64's program memory.

To get a SAVED program back into the C64's memory, you use the LOAD command. Here is how:

1. Make sure the disk is ready in your disk drive.
2. You must know the name of the program you are trying to load. If you forget, just get a directory of the disk (see the previous section, starting on page 69). With many game programs or professional programs purchased from computer stores, you can just use "\*" as the program name, and the Commodore will load the first thing on the disk.
3. To load your program from the disk into the C64, type

```
LOAD "name",8
```

As with the SAVE command, the comma and number 8 tell the computer to load from the disk drive instead of from the tape recorder. If you have the DOS Support set up, you may instead type the shorter command

```
/"name"
```

The slash (/) is an unshifted question mark key.

4. When the C64 prints READY, the program will be in memory. Type the command RUN to start it going. You may also use the LIST command to see a listing of the program.
5. If the red light on the disk drive starts blinking, an error has occurred. The most common error message from the LOAD command is ?FILE NOT FOUND ERROR. This means the C64 couldn't find any program on the disk with the name you gave it. Get a directory of the disk. Try the LOAD command again, and make sure you spell the program name exactly the same way as it is spelled in the directory. If you get a ?SYNTAX ERROR message, then you made a typing mistake or forgot the quotes or comma.

## **Disk Channel Commands**

Now you know how to use the easy disk commands: to SAVE a program onto a disk, get a "directory" of the names of all the programs stored on the disk, and LOAD old programs back into the computer from the disk. These are all you really have to be able to do, to use the disk drive for external memory.

There are several more commands you can use with the disk drive. For instance, you can erase a program you don't need any more. These extra commands are not quite as easy to use as the simple SAVE, LOAD, and directory commands. The next few sections will discuss how to use these extra commands, which involve the use of a "disk command channel."

If you have the DOS Support set up on your Commodore, the disk channel commands are just as easy to use as any of the others. However, to send any of these commands without the DOS Support requires three lines for each command. We will show you exactly what to type for each.

Every disk channel command word **MUST** be followed by a **0:** (zero, colon). This doesn't mean much, it's just left over from older models of Commodore disk drives. Also, if you are using the long (three line) form of the commands, every disk channel command uses a **PRINT#15**. There must be no spaces in this command, and it must be followed by a comma. Otherwise you will get a syntax error message.

In general, follow these rules:

- Type the commands exactly as they appear.
- Do not type any spaces in the commands.
- Type all the commas, quotes, and colons where they belong.

## **SCRATCH**

How do you erase an old useless program from your disk? Use the **SCRATCH** command. Of course, you need to know the name of the program you wish to erase. Type:

```
OPEN 15,8,15  
PRINT#15,"SCRATCH0:name"  
CLOSE 15
```

If you have the DOS Support, just type

```
>"SCRATCH0:name"
```

## INITIALIZE

Sometimes you will get an error condition on your disk drive (blinking red light) that won't go away. Now, you could just turn off your computer and turn it on again. This will reset your disk drive. However, it will also erase your program memory! Instead, you may use the INITIALIZE command. It is shortened to just the letter "I." Type:

```
OPEN 15,8,15
PRINT#15,"I"
CLOSE 15
```

If you have DOS Support, type

```
>"I"
```

This simple command will reset the disk drive without disturbing your computer's program memory.

## OTHERS

There are also commands able to rename a program stored on disk, reorganize the disk memory when needed, copy files, and work with entire sets of programs all at once. You are not likely to need these advanced commands for a while, but when you do they are described in chapter 4 of the 1541 User's Manual.

## The Commodore Printer

The last several sections showed you how to use one of the most useful, convenient ways to permanently store programs: the disk drive. However, we are now ready to move on to a new topic.

The Commodore printer can do many different tasks, but only one is immediately useful to a beginning programmer. As soon as you try to write a program longer than about 25 lines, you will find that when you LIST it, your program does not all fit on the screen. This can make it quite difficult to change your program, fix mistakes, and add to it.

The best way to get around this problem is to print out a listing of your program on paper. If you own a Commodore printer, this is easily done.

The printer must, of course, be hooked up to your C64. Remember that all machines must be off before making any electrical connections, otherwise your computer could be seriously damaged. The printer is connected to the “serial port” of the C64, just like the disk drive. If you already have a disk drive hooked up, just plug the printer into the second socket at the back of the disk drive. If you do not own a disk drive, connect the printer to the “serial port” on the back of the computer (see diagram, page 6), with the short connector cable which came with your printer. Plug the printer into a wall socket also, then turn the machines on, remembering:

*Turn the Commodore 64 on last of all.*

Now your printer will be ready whenever you want to use it.

In order to print a listing of a program, the program must first be in the Commodore 64. You may type it in fresh, or LOAD an old program from disk or tape. It is a very good idea to take the time to SAVE a new program on disk or tape before trying to print it, in case something goes wrong and you have to turn the computer off.

Once the program is ready to LIST, getting a printout is easy. Type:

```
OPEN 4,4  
CMD 4  
LIST
```

When the listing is all done, close the channel of communication between the printer and the computer by typing:

```
PRINT#4  
CLOSE 4
```

There should be no spaces in PRINT#4.

The Commodore printer can do much more than just give you typed LISTings of programs. It can do different kinds of fancy printing, and some models can do pictures. Most of these abilities can only be used from within sophisticated programs. When you get to this stage, please refer to the User's Manual for your printer.

## **Summary**

In this chapter you've learned some useful things about the Commodore and its more common peripherals. We had the most to say about the disk drive, because it is the most useful and complex piece of equipment that you could add to your Commodore 64.

Commodore and other manufacturers will continue to invent new, interesting peripherals for the C64. Modems, which allow the computer to use the telephone, and speech synthesizers, which allow the computer to actually talk, are now commonplace. Soon, we will be using even more of the computer's potential. The peripherals we plug into the computer expand its abilities and increase the amount of fun we can have with it, in work and play.

## **Vocabulary**

peripheral

cartridge

joystick

game paddles

external memory

Datasette recorder

cassette

disk drive

disk

DOS Support Program (DOS Wedge)

printer

## QUICK COMMAND REFERENCE PAGE

### Cassette Recorder

SAVE "name"  
VERIFY "name"  
LOAD "name"

Disk Drive	regular command	with DOS Support
save	SAVE "name",8	SAVE "name",8
replace old copy	SAVE "@0:name",8	SAVE "@0:name",8
directory	LOAD "\$",8 LIST	>\$
load	LOAD "name",8	/"name"
erase	OPEN 15,8,15  PRINT#15, "SCRATCH0:name" CLOSE 15	>"SCRATCH0: name"
format blank disk	OPEN 15,8,15 PRINT#15, "NEW0 DISK,AA" CLOSE 15	>"NEW0:DISK,AA"
reset drive (initialize)	OPEN 15,8,15 PRINT#15, "I" CLOSE 15	>"I"

### Printer (to list a program or directory on paper)

OPEN 4,4  
CMD 4  
LIST  
PRINT#4  
CLOSE 4



# PROGRAMMING

In Chapters 4 and 5 you will be typing programs into the Commodore's memory. This memory is erased when you type NEW or when the computer is turned off. This is no problem for short programs that you can easily retype if you want them again. However, it is inconvenient to retype long programs or programs you wish to use often.

So, if you have a program recorder or disk drive, you may wish to SAVE some of the following examples for later reference or experiments. It is *not* necessary for the use of this book. The decision to SAVE a program or not is up to you. If you want to SAVE any of the programs you do in the coming chapters, just turn back to the proper section in Chapter 3, and follow the instructions there.

## Key Words in BASIC

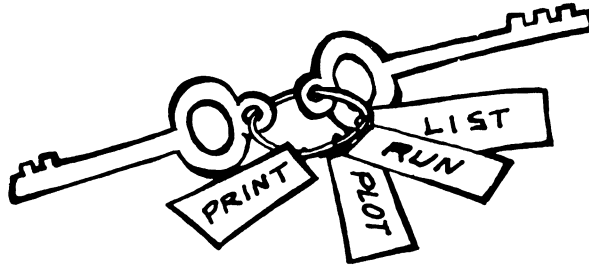
To communicate with the Commodore 64, you must "speak" the same "language." Commodore 64 understands the CBM version of the BASIC language.

Both English and BASIC are communication tools.

- In English, you combine words to make a sentence.
- In BASIC, you combine words, numbers, and symbols to make a statement.
  - Each statement in BASIC is a separate instruction to the computer.
  - Each statement in BASIC begins with a line number.
  - Each statement in BASIC must contain a *key word*.

A key word always follows the line number. It tells the Commodore 64 what to do with the information on that line.


Several key words have been introduced so far, such as PRINT and POKE. In this book you will use many of the common key words in the BASIC language.



## How to Change a Program

The instructions you type into the Commodore 64 are placed in the computer's memory. Once a program is in memory, you can change it by adding lines, erasing lines, or changing lines. This process is sometimes called "editing" a program.

First, type in this program. Remember that you get the  symbol by pressing **SHIFT CLR**.

```
NEW
10 PRINT "
20 REM-WAYS TO MODIFY A PROGRAM
30 PRINT "I'M LEARNING TO USE MY COMPUTER"
40 END
LIST
RUN
```

*Remember:*

- NEW cleans out the Commodore's "memory," and makes it ready for a new program. You should always type NEW to clear the memory so lines of an old program do not get mixed up in the new one.
- REM in line 20 stands for the key word REMARK. The Commodore 64 does not do anything with a REM statement. REMs allow you to put explanations and labels in your program.

Let's add a line.

To add another line, decide where the new instruction should go in the program. Look at the line number before and the line number after the place where you want the new line to go. Choose a number somewhere in between them. Type that number and the new instruction into the Commodore 64 and press RETURN.

Here's an example:

```
35 PRINT "WATCH MY PROGRAM CHANGE"  
LIST  
RUN
```

Now, let's change a line.

To change an instruction, type a new instruction with the same line number as the instruction you wish to change. The new line will replace the old one.

```
30 PRINT "I'M LEARNING TO USE MY COMMODORE 64"  
LIST  
RUN
```

Finally, let's erase a line.

To erase an instruction, type only the line number of the instruction you wish to remove, then press RETURN.

```
35
LIST
RUN
```

## Poems and Things

As you know, the key word PRINT instructs the Commodore 64 to print a message that is enclosed in quotation marks. Any set of characters (numbers, letters, and symbols) can be used between the quotation marks.

```
NEW
10 PRINT "☺"
20 REM - A POEM
30 PRINT "ROSES ARE RED"
40 PRINT "VIOLETS ARE BLUE"
50 PRINT "I LOVE MY COMMODORE"
60 PRINT "HOW ABOUT YOU"
70 END
LIST
RUN
```

Experiment with the poem program. Retype lines 50 and 60, changing the messages between the quotation marks.

The key word PRINT is one you will use often. There is a shortcut to use when you want to type PRINT. If you type a question mark, the Commodore 64 translates it into the word PRINT.

Practice using the ? with the following program. Put a friend's name and address in the computer.

```
NEW
10 PRINT "☺"
20 REM - MY FRIEND'S ADDRESS
30 ? "NAME"
40 ? "STREET ADDRESS"
50 ? "CITY, STATE ZIP"
60 END
LIST
RUN
```

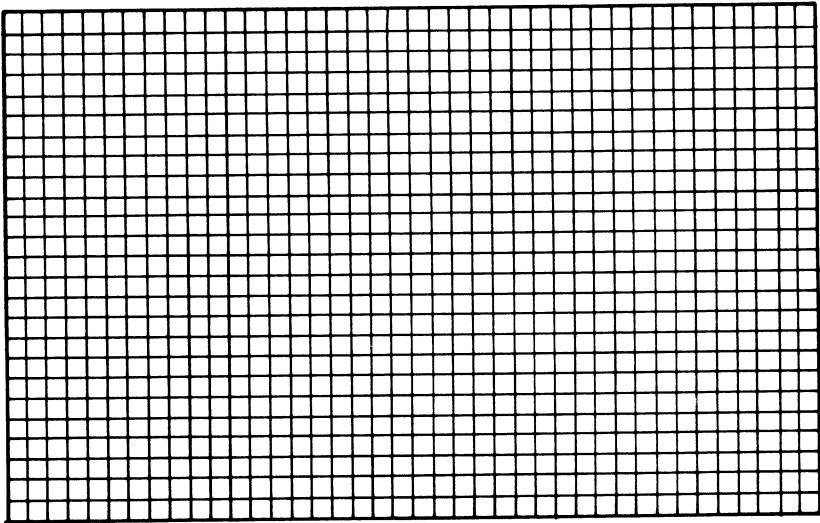
When you type LIST, the Commodore 64 displays the word PRINT on the screen rather than the ?.



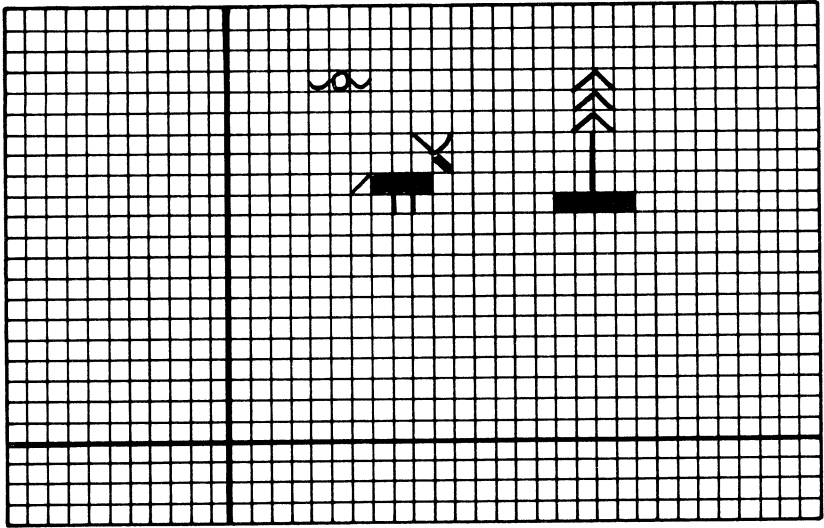
## Design a Picture for the Commodore 64

PRINT statements can be used to draw elaborate pictures, or “graphics,” on the Commodore screen. Designing such a picture can be done by trial-and-error. (This method can be fun!) On the other hand, if you know just what you want your picture to look like, you can design your work of art exactly with very little trouble. Let’s see how to do this.

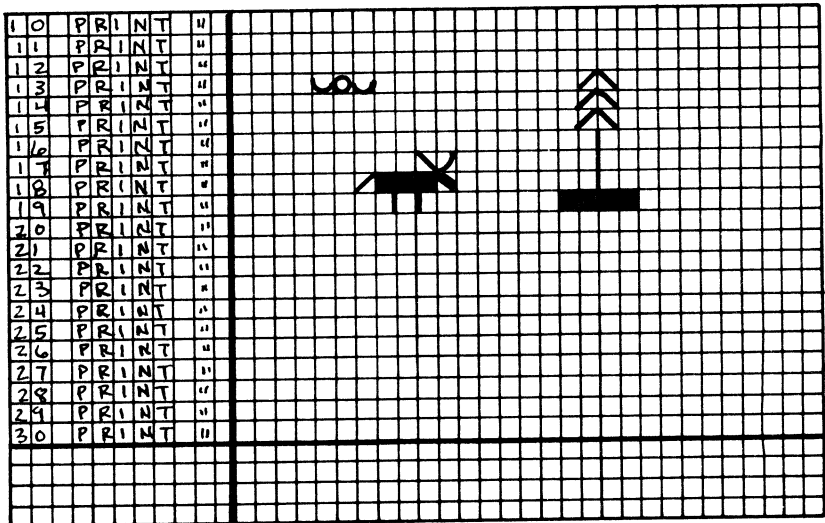
Although it isn’t visible, the Commodore 64 screen has 25 rows and 40 columns. The intersection of these rows and columns creates 1000 squares. You can use the 1000 squares to design a picture for the Commodore 64. First, start with an empty grid, a model of Commodore’s screen.



Draw a vertical line on the grid 10 squares from the left. Then draw a picture to the right of the line by filling in some squares with graphics characters. Put no more than one symbol in each square of the grid. Stop your picture at least 4 lines from the bottom.



Number the lines with consecutive two-digit numbers. Leave a space after each number, then write the word PRINT. After the word PRINT, put another space followed by a quotation mark.







This is how line 13 should look on the screen.

```
13 PRINT "  ◡      ^"
```

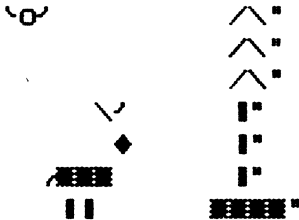
If a line has no graphics characters on it type: the line number, space, PRINT, and press **RETURN**.

If the line *does* have graphics characters type: the line number, space, PRINT, space, quote, and then everything on the line exactly as it appears including spaces. Put the final quote after the last graphics character on the line.

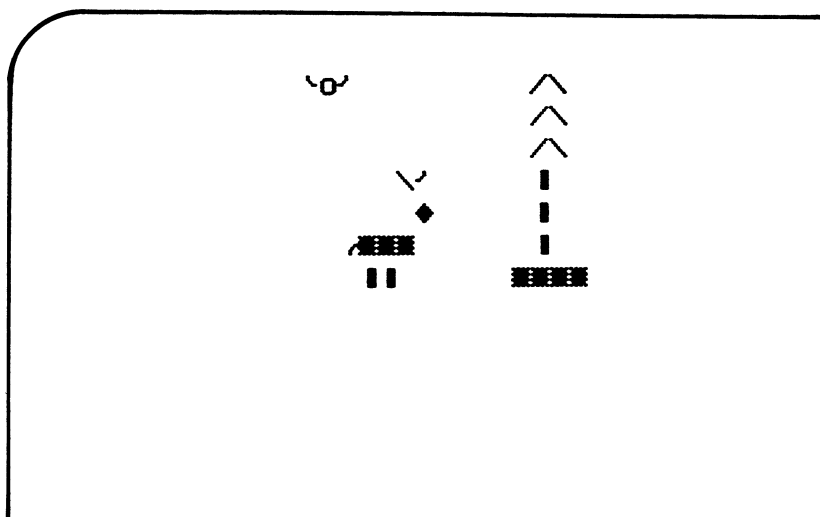
Remember, you can fix an error by using the **DEL** key if you find the error before you press **RETURN**. Otherwise, just retype the line.

Finally, here is the completed program for our picture.

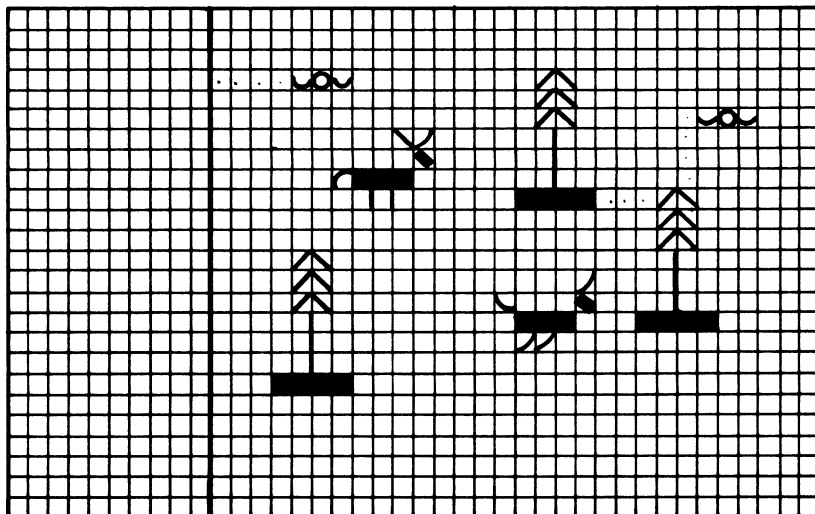
```
5 PRINT "◡"
10 PRINT
11 PRINT
12 PRINT
13 PRINT "  ◡      ^"
14 PRINT "    ◡      ^"
15 PRINT "      ◡      ^"
16 PRINT "        ◡      ^"
17 PRINT "          ◡      ^"
18 PRINT "            ◡      ^"
19 PRINT "              ◡      ^"
31 END
```



And, how the picture looks on the screen.



Add a running deer and some trees and birds if you wish.



## Variables

Inside the Commodore 64's memory, places or addresses can be set up to store information. Imagine these places as a very large set of file drawers or boxes. These locations are called "variables." Some variables store numbers and some store strings. (Remember that a string is any group of characters enclosed in quotation marks.)

Locations with variable names like X, Y, and Z store numbers and are called "numeric variables."

Locations with variable names ending in a \$, like X\$, Y\$, and Z\$, store strings and are called "string variables."

The Commodore 64 handles numeric variables and string variables differently. You can see this by asking the Commodore 64 to print the contents of two variable locations.

Print what is currently stored in variable location A. Type:

**PRINT A**

Print what is currently stored in variable location A\$. Type:

**PRINT A\$**

When the Commodore 64 is first turned on, zero (0) is stored in each numeric variable. Nothing is stored in each string variable. This is also true after you type NEW.

## Using String Variables

You can store any string in a string variable by using the key word LET followed by a letter and a dollar sign. Type:

```
LET F$ = "HI"
```

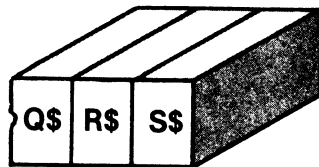
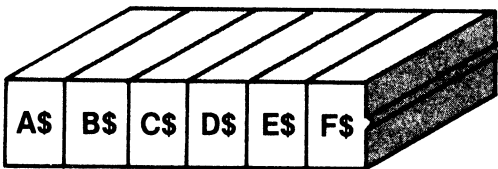
F\$ is a string variable. It is the location where the string "HI" is stored. Now type:

```
PRINT F$
```

When a string variable follows the key word PRINT, the Commodore 64 prints the contents of the string variable location, in this case HI.

```
NEW
```

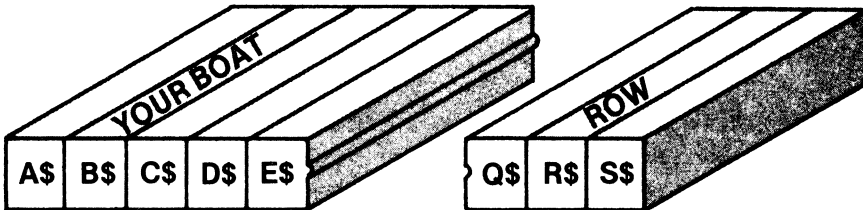
After typing NEW, all the variable locations are empty. Try printing F\$ now.



Now, type in this program:

```
NEW
10 PRINT "Q"
20 LET R$ = "ROW"
30 LET B$ = "YOUR BOAT"
40 PRINT R$,R$,R$,B$
50 END
LIST
RUN
```

Some locations now contain strings.



Information stored in variables is sometimes called “data.” Line 20 stores the data, “ROW” in variable location R\$. Data is also stored in B\$ in line 30. Line 40 PRINTs out the information stored in R\$ three times, then PRINTs out the information stored in B\$. Line 50 causes the program to stop.

## Using Numeric Variables

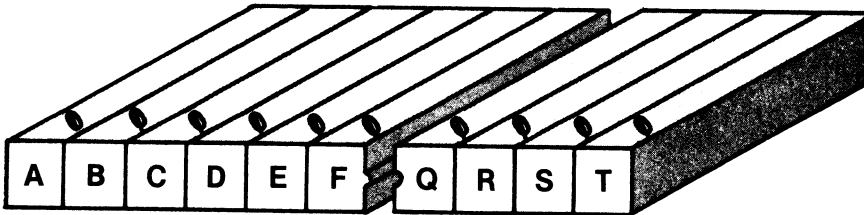
You can store a number in a numeric variable by using the key word LET followed by a letter.

```
LET Y = 7
PRINT Y
```

The letter after the key word LET is a numeric variable. In this case, Y is the name of a location where 7 is stored. The key word PRINT followed by a numeric variable tells the Commodore 64 to print the value of the variable location.

## NEW

After typing NEW, all numeric variable locations contain zero.



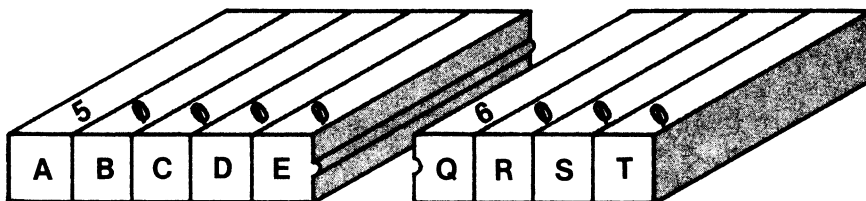
The computer can do arithmetic with the values stored in numeric variables. It cannot do arithmetic with string variables. This is the main difference between numeric and string variables.

Now type in this program:

```
NEW
10 PRINT "Q"
20 LET A = 5
30 LET Q = 2*3
40 PRINT "A=" A
50 PRINT "Q=" Q
60 END
LIST
RUN
```

The information that appears on the screen after you type RUN is called "output." To understand the Commodore 64's output look at the picture below. Then read the program again pretending you are the computer.

Line 20 says to store the number 5 in location A. Line 30 says to multiply  $2*3$  and store the answer in location Q.



The Commodore 64 printed:

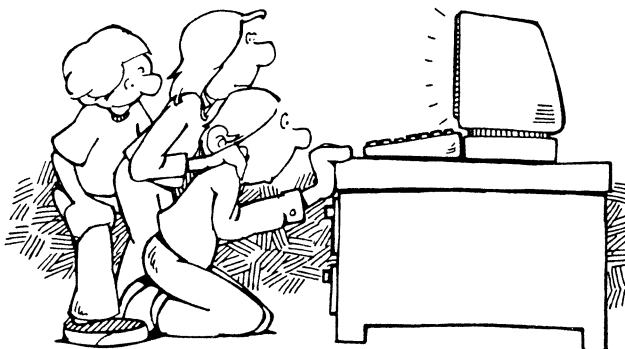
**A= 5** since 5 is stored in A

**Q= 6** since the result of the numeric expression  $2*3$  is stored in Q

*Things to Try:*

```
LET Y = 7
PRINT S;P;Y
```

Can you figure out how you got this famous agent's number?



## Playing with Variables

Type in this program:

```
NEW
10 PRINT "☺"
20 REM - ADD TWO NUMBERS
30 LET A = 7
40 LET M = 12
50 LET S = A + M
60 PRINT "MY LUCKY NUMBERS ARE" A "AND" M
70 PRINT "THAT ADDS UP TO" S
80 END
LIST
RUN
```

In line 50, the Commodore 64 added the value of the variable stored in A to the value stored in M and put the result in S. In line 70, the Commodore 64 printed the value, 19, now stored in S.

You can retype lines 30 and 40 to use your own lucky numbers.

Now try this:

```
NEW
10 PRINT "☺"
20 REM - A NURSERY RHYME
30 LET P$ = "POTATO"
40 PRINT 1,P$
50 PRINT 2,P$
60 PRINT 3,P$
70 PRINT "FOUR"
80 END
LIST
RUN
```



Can you add some lines to this program to complete the nursery rhyme?

*HINT:* Continue the nursery rhyme by starting at line 80. If you begin with line number 90, the computer will never get there because of the END statement in 80, which will tell the Commodore 64 to stop.

Now let's use a graphics character, color, and variables to design a flag. Use the chart to help find the symbol and colors to use.



SYMBOL	MEANS
␣	SHIFT CLR
␣	CTRL 7
␣	C = +
␣	CTRL 3
␣	CTRL 2



```

NEW
100 REM - BACKGROUND COLOR
110 POKE 53281,15
120 :
130 REM - CLEAR SCREEN
140 PRINT "␣"
150 :
160 REM - ASSIGN PATTERN TO VARIABLES
170 LET A$ = "▣██████████▣██████████████████"
180 LET B$ = "▣██████████▣██████████████████"
190 LET C$ = "▣██████████████████████████████"
200 LET D$ = "▣██████████████████████████████"
210 :
220 REM - DRAW DESIGN
230 PRINT A$
240 PRINT B$
250 PRINT A$
260 PRINT B$
270 PRINT C$
280 PRINT D$
290 PRINT C$
300 PRINT D$
310 PRINT C$
320 PRINT D$
330 END
LIST
RUN

```

*Line 170 says: After the quotation mark, press CTRL 7 to get the symbol for blue, then press  + seven (7) times to get 7 graphics characters. Follow that with CTRL 3 for red and  + for 12 more graphics characters.*

*Line 180 is the same as line 170 except use white, CTRL 2, instead of red.*

*Lines 190 and 200 use 19 graphics characters following CTRL 3 in line 190 and CTRL 2 in line 200.*

The Commodore 64 interprets the colon (: ) as though you pressed the RETURN key. The colons in lines 120, 150, and 210 are not necessary and do not change your output. However, they do allow you to see more easily the separate parts that make up the final program.

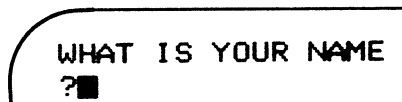
## Having a Conversation with the Commodore 64

Until now, all you could do was type in a program, type RUN, and then sit back and watch the screen. By using the key word INPUT, you can "talk" with the program in the Commodore's memory while it is running.

When you type this program be careful to copy it exactly as it appears. Note that spaces have been left inside the quotation marks in lines 70 and 80.

```
NEW
10 PRINT "?"
20 REM - PROGRAM TO ANSWER COMMODORE 64
30 PRINT "WHAT IS YOUR NAME"
40 INPUT N$
50 PRINT "HOW OLD ARE YOU"
60 INPUT N
70 PRINT N$ " IS " N " YEARS OLD."
80 PRINT "NEXT YEAR " N$ " WILL BE " N+1 "."
90 END
LIST
RUN
```

The screen looks like this:



```
WHAT IS YOUR NAME
?■
```

You might wonder where the question mark came from since it isn't in your program. The question mark shows up because of the key word INPUT.

The cursor continues to flash after the question mark. The Commodore 64 is waiting for you to answer the question and press the RETURN key.

After you answer with your name, the screen looks similar to this:

```
WHAT IS YOUR NAME
?SUE
HOW OLD ARE YOU
?█
```

The Commodore 64 is now asking for your age. You should type a number and press RETURN.

In addition to placing a question mark on the screen, the INPUT statement waits for you to type something in, then takes your response and stores it in the variable location named after the key word INPUT.

In this “conversation” program, the PRINT statements in lines 30 and 50 ask the questions. The INPUT statements in lines 40 and 60 wait for the answers and store them in the computer’s memory. Lines 70 and 80 cause the Commodore 64 to “talk back” to you using the information you put in.

#### *Things to Try:*

- RUN again and type your age in words (for example: “sixteen” instead of “16”). The Commodore will give you a REDO FROM START message because it is looking for a number, not a string, because line 60 says INPUT N, not INPUT N\$. You will also get an error message if you answer “16 years.” To satisfy the Commodore 64, type in a number and press RETURN.

## Using the Commodore 64 to Tell Your Fortune

A FILL-IN is a fun story. You can make it different every time you use it.

Type this program. Don't be concerned if one statement of the program is longer than one line on the screen and spills over to the next line. Type the complete statement before you press RETURN.

```
NEW
100 PRINT "☺"
110 REM - FORTUNE TELLING PROGRAM
120 :
130 REM - QUESTIONS AND ANSWERS
140 PRINT "WHAT IS YOUR FAVORITE COLOR";
150 INPUT C$
160 PRINT "WHAT IS YOUR FAVORITE CAR";
170 INPUT R$
180 PRINT "WHAT IS YOUR FAVORITE SPORT";
190 INPUT S$
200 PRINT "WHAT IS YOUR FAVORITE ANIMAL";
210 INPUT A$
220 PRINT "WHAT IS YOUR NAME";
230 INPUT N$
240 :
250 REM - FILL IN STORY
260 PRINT "☺"
270 PRINT "HERE IS YOUR FORTUNE"
280 PRINT
290 PRINT "THE CRYSTAL BALL SEES " N$
300 PRINT "ARRIVING AT A CHAMPIONSHIP
    " S$ " GAME"
310 PRINT "IN A " C$ " " R$
320 PRINT "ACCOMPANIED BY A PET " A$
330 END
LIST
RUN
```

If the output doesn't look the way you think it should, read "Things to Notice" below.

Just for fun, RUN the program several times. Put in different responses each time.

*Things to Notice:*

- The semicolon (;) at the end of the PRINT lines (140, 160, 180, 200, 220) in the QUESTION AND ANSWER section of your program means something special to the Commodore 64. When you use a semicolon after a PRINT statement, the cursor does not move to the next line. It stays where it is after PRINTing the line and then does the next instruction. Therefore the question and your answer will appear on the same line.
- Lines 290 through 320 contain typed spaces inside quotation marks. If these spaces are omitted, words will run together on the screen. Retyping a line to include the spaces will correct this problem.

## **Writing Your Own FILL-IN**

Follow these steps to write a FILL-IN program of your own. It may be useful to read all the steps before you begin.

STEP 1: Write a short story or poem on a piece of paper.

Example: MARY HAD A PURPLE TRUCK  
ITS WHEELS WERE. . . . .

STEP 2: Erase some of the words.  
Replace them with string variables.

Example: N\$ HAD A C\$ T\$

STEP 3: Make a list on paper of each string variable and what it represents. This list will be used later as a guide.

Example: N\$ = Name  
C\$ = Color  
T\$ = A favorite toy

STEP 4: Write the program in two parts.

*PART 1:* This part of the program collects data to be used later in the story. It consists of a group of statements using the key words PRINT and INPUT. Using the guide you developed in STEP 3, write a program similar to the following:

```
NEW
10 PRINT " "
20 REM - MY OWN FILL-IN
30 PRINT "WHAT IS YOUR NAME";
40 INPUT N$
50 PRINT "NAME YOUR FAVORITE COLOR";
60 INPUT C$
70 PRINT "NAME YOUR FAVORITE TOY";
80 INPUT T$
```

*PART 2:* The last part of the program PRINTs your story. It consists of several PRINT statements, including one to clear the screen. Use the version of your story that you wrote in STEP 2 that contains your string variables.

```
90 PRINT " "
100 PRINT N$ "HAD A" C$ " " T$
110 PRINT "ITS ..."
```

STEP 5: RUN your program. If your first attempt results in a SYNTAX ERROR, don't be discouraged. Very few programs work the first time. LIST and look for mistyped words, missing quotation marks, or other mistakes. These mistakes are known as "bugs." When your program is fixed ("debugged"), share your success with a friend.

## Using the Commodore 64 to Compute Information

Here are a few examples to demonstrate how a computer can be used to perform some simple calculations.

Type in the following program. Using the formula in line 130, the Commodore 64 will compute the number of miles per gallon that a car gets when gas is purchased in liters.

```
NEW
10 PRINT " "
20 REM - MILES PER GALLON
30 REM - GATHER INFORMATION
40 PRINT "HOW MANY MILES DID YOU TRAVEL "
50 INPUT M
60 PRINT "HOW MANY LITERS OF GAS DID YOU BUY"
70 INPUT L
80 PRINT: PRINT
90 REM - PRINT OUT INFORMATION
100 PRINT "IF YOU WENT " M " MILES"
110 PRINT "AND BOUGHT " L " LITERS OF GAS"
120 PRINT "YOU ARE GETTING ";
130 PRINT M/(L * 1.06/4) " MILES PER GALLON."
140 END
LIST
RUN
```

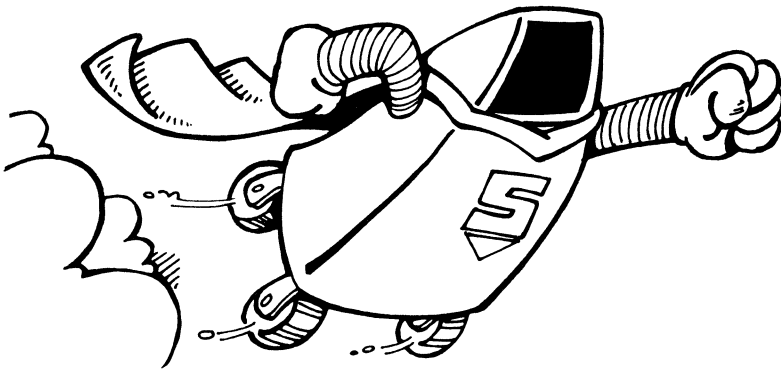


PRINT: PRINT in line 80 creates two blank lines when you RUN the program. Remember, the Commodore 64 interprets the colon (: ) as though you pressed the RETURN key. Another way to write this would be:

```
80 PRINT  
85 PRINT
```

The next program instructs the Commodore 64 to convert how much you weigh on earth to what your weight would be on the moon.

```
NEW  
10 PRINT "☺"  
20 REM - WEIGHT CONVERSION  
30 PRINT "HOW MUCH DO YOU WEIGH";  
40 INPUT W  
50 PRINT: PRINT  
60 PRINT "ON THE MOON YOU WOULD WEIGH " W/6  
70 END  
LIST  
RUN
```



## Teach the Commodore 64 Your Name

In this section, you will use the key word GOTO to create an "infinite loop." An infinite loop occurs when the same instructions are done over and over again.

The Commodore 64 usually follows instructions in order, beginning at the line with the smallest number. It progresses through the program, always going to the line with the next larger number unless instructed to do otherwise. The key word GOTO allows you to instruct the computer to change that order.

```
NEW
10 PRINT " "
20 REM - NAME FOREVER
30 PRINT "SUSIE.."
40 GOTO 20
50 END
LIST
RUN
```

To interrupt this infinite loop press the **STOP** key, since the Commodore 64 never gets to line 50, END.

To understand what happened in the Susie program, pretend you are the computer while reading the program instructions.

Line 10 clears the screen and line 20 explains the purpose of the program. In line 30, the Commodore 64 PRINTed SUSIE.. on the screen. In line 40, the computer was instructed to GOTO line 20 again, which it did. After another explanation, line 30 again PRINTed SUSIE... Line 40 sent the Commodore 64 back to line 20 again...over and over until you interrupted the RUN.

When you pressed **STOP**, a message like **BREAK IN 30** appeared. This is Commodore 64's way of saying it was interrupted while performing line 30.

Now let's give the program a different name to print. To make the change in line 30, retype it. The Commodore 64 will discard the old instruction and replace it with the new one. Make sure the semicolon is outside the quotation marks.

```
30 PRINT "MICHELE..";  
LIST  
RUN
```

To interrupt press **STOP**.

The difference between the Susie and Michele programs is the semicolon. Remember, the cursor does not move to the next line when you use a semicolon at the end of a **PRINT** statement. It instructed the Commodore 64 to print one name right after the last, filling the screen with the name.





```

30 PRINT "███████████▣███████████▣███████████▣███████████";
LIST
RUN

```

To interrupt press STOP.

Now let's explore using more graphics symbols with an infinite loop. Use this chart to help you with the next program.

SYMBOL	MEANS
▣	SHIFT CLR
∖	SHIFT J
0	SHIFT W
∕	SHIFT I
∕	SHIFT U
∕	SHIFT K

```

NEW
10 PRINT "▣"
20 REM - PICTURE FOREVER
30 PRINT "...∖0∕";
40 GOTO 20
50 END
LIST
RUN

```

You can slow down the output if you wish by holding down the CTRL key.

When you are ready to go on press STOP.

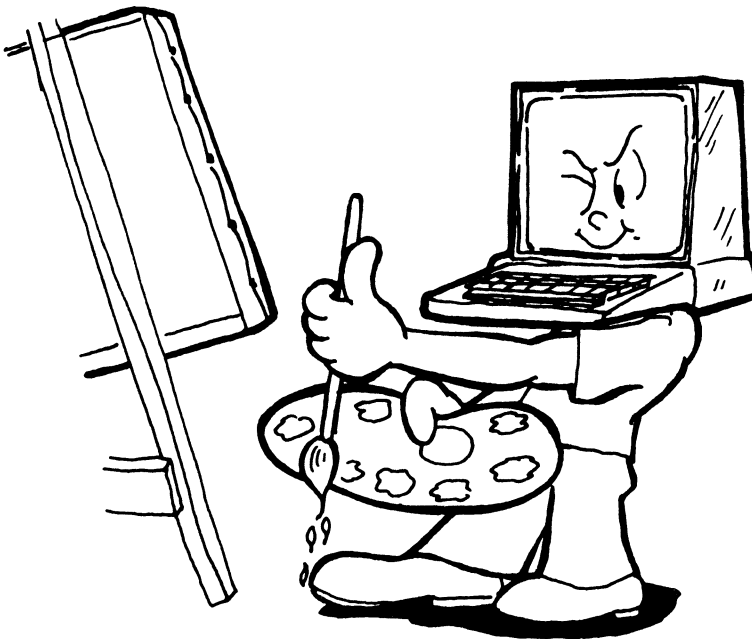
*Things to Try:*

- Add another line to your design.

```
35 PRINT "...:0'";  
LIST  
RUN
```

Press STOP to interrupt.

- Make up a design of your own using both color and graphics.



## Teach Commodore 64 to Count

As you already discovered, a computer loop repeats a series of instructions over and over very quickly. This is one of the abilities that makes a computer such a powerful tool. After a while, a human being would become bored or tired, or make errors. A computer, however, will accurately and tirelessly repeat a task as long as necessary.

You can write a program using a loop to teach the Commodore 64 to count.

```
NEW
10 PRINT "0"
20 LET C = 1
30 REM - COUNT FOREVER
40 PRINT C
50 LET C = C+1
60 GOTO 30
70 END
LIST
RUN
```

To interrupt, press STOP.

Line 20 starts the counting by storing the value 1 in variable location C. Line 30 explains that this is the beginning of the loop and line 40 tells the computer to PRINT the contents of C. The counting takes place in line 50. Each time through the loop the value stored in C increases by 1. Line 60 sends the Commodore 64 back to the beginning of the loop.

By adding a comma to the PRINT statement, the Commodore 64 prints the numbers across the screen.

```
40 PRINT C,  
LIST  
RUN
```

*Things to Try:*

- How long do you think it would take the Commodore 64 to count to one million (1000000)? If you have time, try it.
- With only one change, the counting program can instruct the Commodore 64 to count by steps different than one.

```
50 LET C = C + 2  
LIST
```

A listing of your program should look like this:

```
10 PRINT "0"  
20 LET C = 1  
30 REM - COUNT FOREVER  
40 PRINT C,  
50 LET C = C + 2  
60 GOTO 30  
70 END  
LIST  
RUN
```

To interrupt, press STOP.

*More Things to Try:*

- Teach the Commodore 64 to count by even numbers. (HINT: In line 20, let C = 2.)
- To count by 5's make changes in both line 20 and 50.



## Another Way to Count

So far you have looked only at infinite loops, which go on forever unless interrupted. These are valuable under certain circumstances. Sometimes, however, you want a job done an exact number of times, no more, no less. Let's say you want a job done exactly 10 times and you want the computer to keep track of how many times it has been done.

One way to control the counting is with a FOR-NEXT loop. The key words, FOR and NEXT, are used in separate statements.

Look at this example. Then, type it in and RUN.

```
NEW
10 PRINT "C"
20 REM - COUNT TO TEN
30 FOR C = 1 TO 10
40 PRINT C
50 NEXT C
60 END
LIST
RUN
```

Line 30 begins the loop with the key word FOR, and tells how many times the Commodore 64 goes through the loop. C will be counting from 1 to 10. Line 50 uses the key word NEXT to show the end of the loop and to instruct the computer to return to the beginning of the loop.

Change the program to count to 20.

```
20 REM - COUNT TO TWENTY
30 FOR C = 1 TO 20
LIST
RUN
```

Modify the program to print a name 20 times.

```
20 REM - PRINT NAME 20 TIMES
40 PRINT "COMMODORE 64"
LIST
RUN
```

*Things to Try:*

- Can you modify this program to count to 50?
- Can you modify this program to print your name 5 times?

You have used the FOR-NEXT loop to count by ones. The key word STEP allows you to count in steps other than by ones.

```
NEW
10 PRINT "5"
20 REM - COUNT BY FIVES
30 FOR X = 5 TO 100 STEP 5
40 PRINT X
50 NEXT X
60 END
LIST
RUN
```

To count by 5's you must also start at 5. Notice that in line 30 the value of X begins at 5 and steps by 5's until the computer reaches 100.

Let's change the program to count by tens.

```
20 REM - COUNT BY TENS
30 FOR X = 10 TO 100 STEP 10
LIST
RUN
```

Experiment. Can you modify the program to count by twos or fours?

## Slowing Down the Action

The FOR statement followed immediately by the NEXT statement can be used to slow down the printing of messages on the screen. This is called a delay loop.

```
NEW
10 PRINT "Q"
20 REM - MAKING A DELAY LOOP
30 PRINT "HOLD YOUR BREATH TIL I COUNT TO 4000"
40 FOR D = 1 TO 4000
50 NEXT D
60 PRINT "I'M DONE!"
70 END
LIST
RUN
```

Lines 40 and 50 are the delay loop. It does nothing 4000 times. There is about a one second delay for every 800 times the Commodore 64 goes through the loop. Did you feel as if you held your breath for 5 seconds? RUN the program again and count the seconds.

The higher the numbers used in the delay loop, the longer the pause. Change line 40 to increase the length of the delay loop.

```
30 PRINT "HOLD YOUR BREATH TIL I COUNT TO 8000"
40 FOR D = 1 TO 8000
LIST
RUN
```

Did the delay seem more like 10 seconds? If you wish, experiment with other numbers in the delay loop.

Use this chart to help find the correct symbol to use in the next program.

SYMBOL	MEANS
␣	SHIFT CLR
␣	CRSR↓
␣	SHIFT J
␣	SHIFT W
␣	SHIFT I

The delay loop can be used to slow down the action. First, look at this program without the delay loop.

```
NEW
10 PRINT "␣"
20 REM - DESIGN FOREVER
30 PRINT "␣␣";
40 GOTO 20
50 END
LIST
RUN
```

Now add the delay. The entire delay loop can be typed on the same line with the FOR and the NEXT statements separated by a colon (:).

```
35 FOR D = 1 TO 200: NEXT D
LIST
RUN
```

Experiment for a while if you wish.

Slow down the action with FOR D = 1 TO 1000: NEXT D  
Speed up the action with FOR D = 1 TO 50: NEXT D

Then try this program.

```
NEW
10 PRINT "Q"
20 REM - COUNTDOWN
30 PRINT "ALL SYSTEMS GO"
40 PRINT "READY FOR TAKEOFF"
50 FOR A = 10 TO 1 STEP -1
60 PRINT A
70 FOR D = 1 TO 800
80 NEXT D
90 NEXT A
100 PRINT "BLAST OFF"
110 END
LIST
RUN
```








*Things to Notice:*

- The counting loop in the “Countdown” program starts at 10 and subtracts one number, STEP -1, each time through the loop.
- Inside the “Countdown” loop is another loop in line 70 that causes the delay. This is known as a “nested loop.”

## An Electronic Message

This program will let you send an electronic greeting to a friend by using some of the special features of the Commodore 64.

Use the chart to find the correct symbol to use with the programs in this section.

SYMBOL	MEANS
	SHIFT CLR
	CTRL 2
	CTRL 3
	CTRL 9
	CTRL 0
	CRSR↓
	CRSR→

To send your message, type:

```
NEW
10 REM - NAME FOREVER
20 PRINT "☐"
30 FOR D = 1 TO 500
40 NEXT D
50 PRINT "☐TO MY FRIEND JODY"
60 FOR D = 1 TO 500
70 NEXT D
80 GOTO 10
90 END
LIST
RUN
```

Press the STOP key to interrupt.



Line 20 says to clear the screen and move the cursor 10 lines down from the home position and 10 columns over. Finally the semicolon tells the computer to print the message at that place rather than going to the beginning of the next line.

To stop the infinite loop, press STOP.

You can control the number of times a message appears on the screen by using the FOR-NEXT loop.

Add this to your program:

```
10 REM - YOUR MESSAGE IN LIGHTS
15 FOR X = 1 TO 10
80 NEXT X
LIST
RUN
```

When you are ready, share your greeting with a friend.





## Summary

You've now seen many examples of BASIC programs, and along the way you learned some new BASIC key words. You have also seen some very *useful* ideas, like using a graphics grid to plan your computer drawings.

If you liked the idea of being able to plan your drawings, there is a grid in the back of this book which looks very much like the invisible grid on the Commodore 64 screen. If you photocopy this grid, then you will always be able to use a new grid to plan new drawings. The empty grid is on page 167. Further examples of grid art are in Appendix 6, page 168.

## Vocabulary

key words

REM—remark statement

? to stand for PRINT

edit

graphics grid

numeric variable

string variable

LET

INPUT

GOTO

infinite loop

FOR-NEXT loop

delay loop

nested loops

# LIGHTS, SOUND, ACTION!

In this chapter, most of the programs are longer and more complicated than those in earlier chapters. You can extend your enjoyment of the programs after they are up and running by making changes in PRINT statements, color, variable values, GOTO statements, FOR-NEXT statements, etc. You may get some interesting results or some disasters. If you get an error message you can accept the challenge to find it, or you can erase the program with NEW and go on.

## Using Random Numbers

The Commodore 64 can produce a "surprise" number between 0 and 1. One way is with a PRINT statement.

```
PRINT RND(1)
```

Another way is with a LET statement.

```
LET C = RND(1)  
PRINT C
```

Producing a random decimal number may not seem impressive. However, it opens the door to lots of computer applications. Random color allows you to create amazing graphic displays. An element of chance enlivens any computer game.

You can get a range of numbers by combining the random number function, RND, with some arithmetic.

The next program will choose a number in the range 0 to 14 to represent a color for the cursor. We won't use light grey, because it is the color the background will be and light grey print will not show up. (See page 32 if you've forgotten how the background color is changed.)

SYMBOL	MEANS
♥	SHIFT S

**NEW**

```
10 POKE 53281,15: REM - LIGHT GREY BACKGROUND
20 LET C = INT(RND(1)*15): REM - GIVES NUMBER
  0 TO 14
30 POKE 646, C
40 PRINT "I ♥ WRITING IN COLOR"; C
50 END
LIST
RUN
```

Run the program again to get a randomly different result. Line 20 gives you a random value for C. C will then represent the COLOR in line 30.

You can also use random numbers to create a variety of backgrounds on the screen.

SYMBOL	MEANS
☺	SHIFT CLR

In this program lines 40 and 50 give random numbers between 0 and 15. These random numbers are POKEd into the address 53281, which controls screen background color, and into the address 53280 which controls border color. You can press STOP and RESTORE to put the screen back to normal.

```
NEW
10 REM - RANDOM SCREEN AND BORDER
20 PRINT "☐"
30 FOR T = 1 TO 25
40 POKE 53281, INT(RND(1)*16)
50 POKE 53280, INT(RND(1)*16)
60 FOR D = 1 TO 500: NEXT D
70 NEXT T
80 END
LIST
RUN
```

Try this program to simulate the roll of a pair of dice. To appreciate the element of chance, RUN this program several times. (Note: Lines 30 and 40 each produce a number between 1 and 6.)

```
NEW
10 PRINT "☐"
20 REM - DICE ROLL
30 LET D1 = INT(RND(1)*6)+1
40 LET D2 = INT(RND(1)*6)+1
50 PRINT "DIE 1 = " D1
60 PRINT "DIE 2 = " D2
70 PRINT "TOTAL ROLL = " D1+D2
80 END
LIST
RUN
```

Use this program to choose a date in March:

```
NEW
10 PRINT "☺"
20 REM - LUCKY DAY
30 LET D = INT(RND(1)*31)+1 : REM - GIVES
   NUMBER 1 TO 31
40 PRINT "YOUR LUCKY DAY IS MARCH " D
50 END
LIST
RUN
```

Change the program to find a lucky day in your birth month.

- Use  $\text{INT}(\text{RND}(1)*30)+1$  for April, June, September, or November.
- Use  $\text{INT}(\text{RND}(1)*31)+1$  for January, March, May, July, August, October, or December.
- Use  $\text{INT}(\text{RND}(1)*28)+1$  if you were born in February. Use  $\text{INT}(\text{RND}(1)*29)+1$  for leap years, like 1984.

## Making Decisions

The Commodore 64 can be told to make a type of “decision” or “choice.” There is a pair of BASIC key words IF . . . THEN.

We humans use “IF . . . THEN” very often. IF it snows, THEN we can go skiing. IF we get tickets, THEN we will see a movie.

This program shows what the Commodore does when it comes to a line with an IF . . . THEN statement:

```
NEW
10 PRINT "☺"
20 PRINT "HOW OLD ARE YOU?"
30 PRINT "GIVE ME THE NUMBER AND PRESS RETURN"
40 INPUT A
50 IF A < 16 THEN PRINT "YOU ARE TOO YOUNG
   TO DRIVE"
60 IF A > 15 THEN PRINT "YOU ARE OLD ENOUGH
   TO DRIVE"
70 END
LIST
RUN
```

When the Commodore 64 comes to line 50, it makes a choice. IF the age the person input, is less than 16, THEN the computer does the PRINT instruction. Otherwise it skips to the next line *without* doing the PRINT instruction.

When it gets to line 60, it has another choice. IF the number is greater than 15, THEN it prints line 60's message. Otherwise it skips printing the message and goes immediately to the next line.

Because of the IF...THEN instruction, the Commodore 64 can "choose" which message to PRINT, and which message to skip over.

Any BASIC instruction can be put after the word "THEN." The most common thing to put after the "THEN" is either a PRINT instruction or a GOTO instruction. The instruction will be obeyed *only* if the "IF" condition is true. The next program gives another example of this.

## The Game of Guess

Have you ever played the Game of Guess? It is usually a two-person game. One person thinks of a number from 1 to 100. The other person attempts to guess the number in the fewest tries possible. The player who thought of the number gives clues of "higher" or "lower" after each guess until the correct number is guessed.

Here is a program that teaches the Commodore 64 to play the Game of Guess. Notice three things as you read through the program.

1. The Commodore 64 picks a random number in the range of 1 to 100 in line 20.
2. The key words IF...THEN are used in lines 70, 80, and 90 to compare your Guess with the secret number.

3. Line 110 keeps count of the number of guesses.

```
NEW
10 PRINT "N"
20 LET N = INT(RND(1)*100)+1
30 LET C = 1
40 PRINT "GUESS MY NUMBER"
50 REM - GUESS AND CHECK GUESS
60 INPUT G
70 IF G = N THEN GOTO 130
80 IF G < N THEN PRINT "GUESS HIGHER"
90 IF G > N THEN PRINT "GUESS LOWER"
100 PRINT
110 LET C = C+1
120 GOTO 50
130 PRINT "YOU GUESSED MY NUMBER!!"
140 PRINT "IT TOOK YOU "; C; " TRIES"
150 END
LIST
RUN
```

Do you have a strategy for guessing? It is always possible to get the number within 7 guesses.

Play the game again by typing RUN or add this routine to allow the program user to make the choice.

```
141 PRINT
142 PRINT "DO YOU WANT TO PLAY AGAIN (TYPE Y
OR N)"
143 INPUT A$
144 PRINT
145 IF A$ = "Y" THEN GOTO 10
146 PRINT "BYE NOW"
LIST
RUN
```

## Flash Cards for Multiplication Drill

You can use the Commodore 64 to help you learn the multiplication tables by turning the screen into an electronic flashcard.

This program will give you ten problems at a time. It will also keep track of the number you have correct and give you a percentage score at the end.

Here is a list of the variables you will use:

P = Problem Number  
N1 = First Flashcard Number (0-9)  
N2 = Second Flashcard Number (0-9)  
RA = Real Answer  
YA = Your Answer  
C = Number of Correct Answers

NEW

```
100 PRINT "☺"  
110 PRINT "10 PROBLEM MULTIPLICATION DRILL"  
120 FOR P = 1 TO 10  
130 LET N1 = INT(RND(1)*10)  
140 LET N2 = INT(RND(1)*10)  
150 LET RA = N1 * N2  
160 PRINT "WHAT IS THE ANSWER TO " N1 "*" N2;  
170 INPUT YA  
180 IF RA = YA THEN GOTO 220 : REM - TEST YOUR  
ANSWER  
190 REM - WRONG ANSWER  
200 PRINT "THE CORRECT ANSWER IS " RA  
210 GOTO 250  
220 REM - CORRECT ANSWER  
230 PRINT "YOU ARE RIGHT!!!"  
240 LET C = C+1  
250 REM - TIME DELAY TO SEE RESPONSE  
260 FOR D = 1 TO 1000: NEXT D  
270 NEXT P  
280 PRINT "YOU GOT " C " RIGHT FOR " C*10 "%"  
290 END
```

LIST

RUN



Once you are able to get 100%, add an extra challenge to your program. The lines below will erase each problem before asking for the answer.

```
165 FOR D = 1 TO 300: NEXT D
166 PRINT "☹"
LIST
RUN
```

If you wish the problem to remain on the screen a shorter length of time, change line 165:

```
165 FOR D = 1 TO 200: NEXT D
```



## POKE-ing the Screen Memory

Everyone knows that computers have “memory,” but most people don’t realize that every piece of memory has an “address.” The Commodore 64’s memory is separated into little pieces called “bytes” (pronounced “bites!”). A “byte” is just enough memory to hold ONE typed character. When you type the program line

```
10 PRINT "HELLO"
```

it takes up about 16 bytes or pieces of memory, because there are sixteen letters, spaces, and characters in that line.

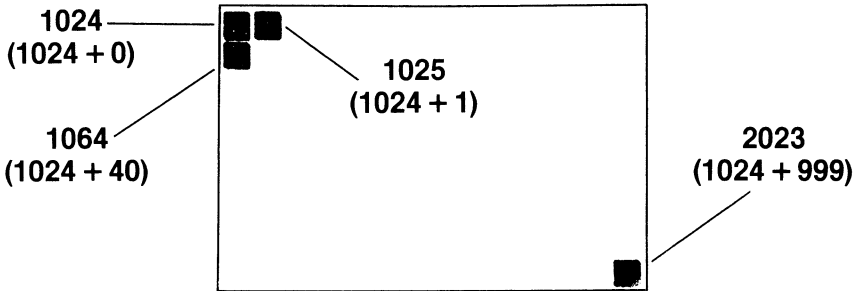
Every single byte in the memory has an *address* so the computer can find it. The Commodore 64 has about 64000 (sixty-four thousand) bytes of memory, numbered 0 to 63999.\*

The key word POKE is used to put information into a memory address. In this section, we will work with the pieces of memory which control the TV display screen.

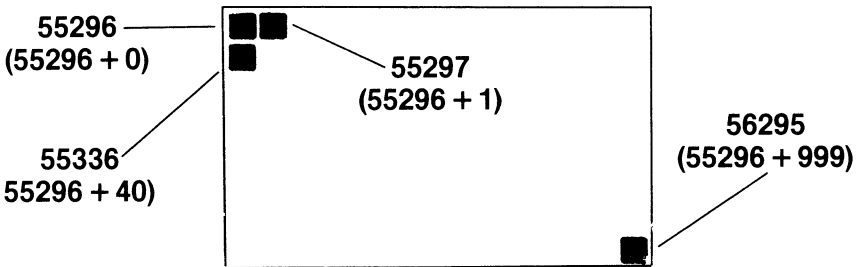
The Commodore 64 screen is 40 characters wide and 25 lines high. Therefore, there are 1000 positions on the screen in which to display characters. Each of these positions has an address. The first position, in the top left of the screen, is numbered 1024. The space next to it has address 1025. If you go ten spaces over, the address will be  $1024 + 10$ , or 1034. If you go forty spaces over, to address  $1024 + 40$  or 1064, you come to the end of the line and pop down to the second line. The last space on the screen, in the lower right corner, is  $1024 + 999$  or 2023.

---

\*Actually, the addresses go all the way up to 65535, but the higher addresses aren’t usually used in programs.



You already know how to use PRINT to put a character on the screen. But there is another way: You can POKE a code for a character into a memory address which controls a position on the screen. Now you must *also* give the character a color. This means using another set of memory addresses labeled 55296 to 56295.



Thus, each screen position is controlled by two sets of addresses. One set controls the placement of a character on the screen and the other set controls that character's color. Let's test this out.

First begin with a clear screen:

**PRINT " "**

Now, let's poke a ●, code 81, into address 1524 (1024 + 500). Type this line just as it looks:

**POKE 1024+500, 81**

Nothing seems to have happened. The ● is there, but since the screen location has not been given a color, you cannot see the character. Put the color yellow, code number 8, into the corresponding address 55796, (55296 + 500):

**POKE 55296+500,8**

Now you should see a yellow dot near the middle of your screen.

Now, try this:

**NEW**

```
10 POKE 53281,0 : REM - BLACK BACKGROUND
20 FOR X = 1 TO 25
30 PRINT "C"
40 READ - LETTER C IN YELLOW
50 POKE 1024+619, 3 : REM - CODE FOR C
60 POKE 55296+619, 7 : REM - CODE FOR YELLOW
70 REM - NUMBER 6 IN ORANGE
80 POKE 1024+620, 54 : REM - CODE FOR DIGIT 6
90 POKE 55296+620, 8 : REM - CODE FOR ORANGE
100 REM - NUMBER 4 IN GREEN
110 POKE 1024+621, 52
120 POKE 55296+621, 5
130 FOR D = 1 TO 300: NEXT D
140 NEXT X
150 END
LIST
RUN
```

*Things to Try:*

- Write a program to place your initials on the screen with POKE statements. You can find the POKE codes for the letters in the appendix (page 165). Also remember to give each letter a color.

## Twinkling Lights

This section combines random screen locations, random characters, and random colors to produce twinkling lights.

NEW

```
10 PRINT "☐"  
20 POKE 53281,15: REM-LIGHT GREY BACKGROUND  
30 LET L = INT(RND(1)*1000): REM-RANDOM  
  SCREEN LOCATION 0-999  
40 LET S = INT(RND(1)*256): REM-RANDOM  
  CHARACTER SHAPE 0-255  
50 LET C = INT(RND(1)*15): REM-RANDOM COLOR  
  EXCEPT LIGHT GREY 0-14  
60 POKE 1024+L, S  
70 POKE 55296+L, C  
80 END  
LIST  
RUN
```



Line 60 POKEs a random shape into the memory location for characters while line 70 POKEs random color into the corresponding memory location for color.

Now, look on the screen to see where the point of light is located. In all probability, this program will produce different results the next time it is run. RUN it several times and see.

Well, that is hardly twinkling, so let's add some more to the program.

```
5 FOR T = 1 TO 25
75 NEXT T

LIST

RUN
```

That's a little better, but the lights twinkle so quickly that it's impossible to enjoy the screen. You can fix that by adding a delay loop.

```
72 FOR D = 1 TO 200: NEXT D

LIST

RUN
```

RUN this again if you wish. You can vary the number of lights that appear on the screen by changing line 5. You can control how long before the next light appears on the screen by changing the delay loop in line 72.

## Design a Quilt

You can make a quilt-like design on the Commodore 64 if you fill each screen location with a character. Since there are 1000 positions on the screen, the FOR-NEXT loop should be done 1000 times.

**NEW**

```
10 PRINT "☐"  
20 POKE 53281,15  
30 FOR T = 1 TO 1000  
40 LET L = INT(RND(1)*1000): REM - RANDOM  
   SCREEN LOCATION 0-999  
50 LET S = INT(RND(1)*128)+128: REM - RANDOM  
   SHAPE 128-255  
60 LET C = INT(RND(1)*15): REM - RANDOM COLOR  
   0-14  
70 POKE 1024+L, S  
80 POKE 55296+L, S  
90 NEXT T  
100 END
```

**LIST**

**RUN**

Oops!! Some screen positions were picked more than once so there are some holes in the quilt.

PEEK is the key word that lets you look at the number stored in a memory address. If you PEEK into a location and IF it is not empty THEN you should choose another location. (32 is the code for a space, or "empty," and "<>" means "not equal to.")

SYMBOL	MEANS
☐	SHIFT CLR
<	SHIFT,
>	SHIFT.

Clear the screen then add:

```
65 LET F = PEEK(1024 + L)    Check screen  
                             location L  
66 IF F <> 32 THEN GOTO 40  Get another location  
LIST                          number  
RUN                            if L is not empty
```

It will take a while to fill the entire screen so relax and wait for the READY to appear on the screen. The computer may choose a location in line 40 more than once. But before it fills a location in lines 70 and 80, the computer PEEKs into the chosen location to see that it contains a space. If the computer doesn't find it empty, it returns to line 40 to choose another location. The program will stop when all 1000 empty locations have been filled.

*Things to Try:*

- Choose a color you like and make your quilt all the same color. For example:

```
60 LET C = 8  
LIST  
RUN
```

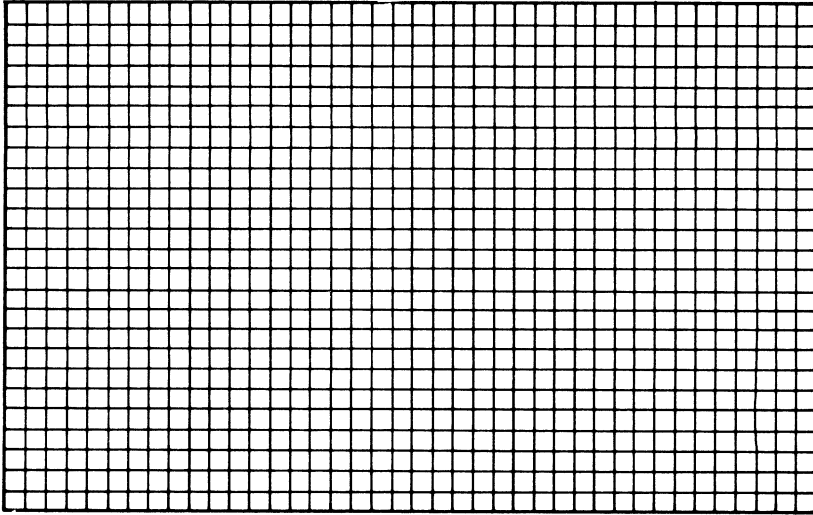
- Make your quilt pieces all have the same pattern. Choose a design to use for line 50 from the Poke Code Character Chart in the appendix. For example, LET S = 200 will make our quilt pieces all have graphics character number 200 on them.

```
50 LET S = 200  
60 LET C = INT(RND(1)*15)  
LIST  
RUN
```

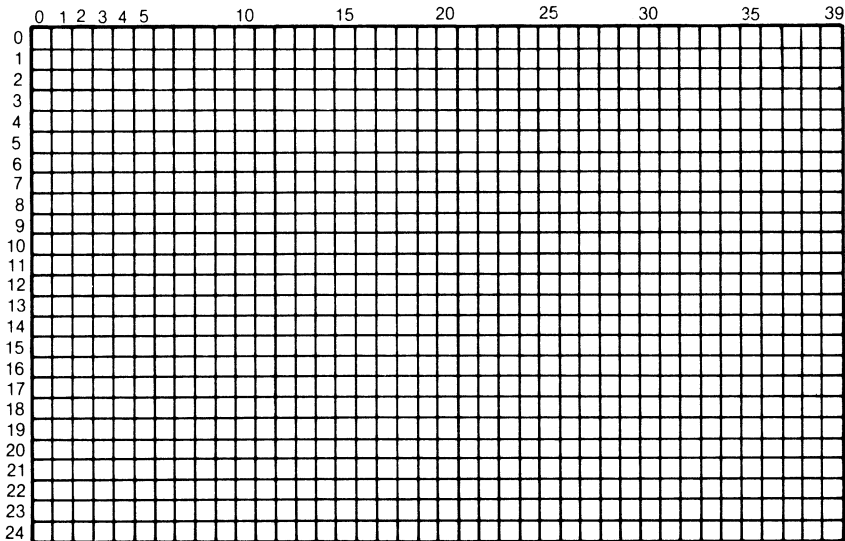


## Hearts and More Hearts

You have been locating the 1000 positions on the Commodore 64 screen by adding the numbers 0 through 999 to the uppermost left screen position, 1024.



There is another way to think of the screen. It has 25 rows of 40 characters each. Number the rows 0 through 24; the columns, 0 through 39.

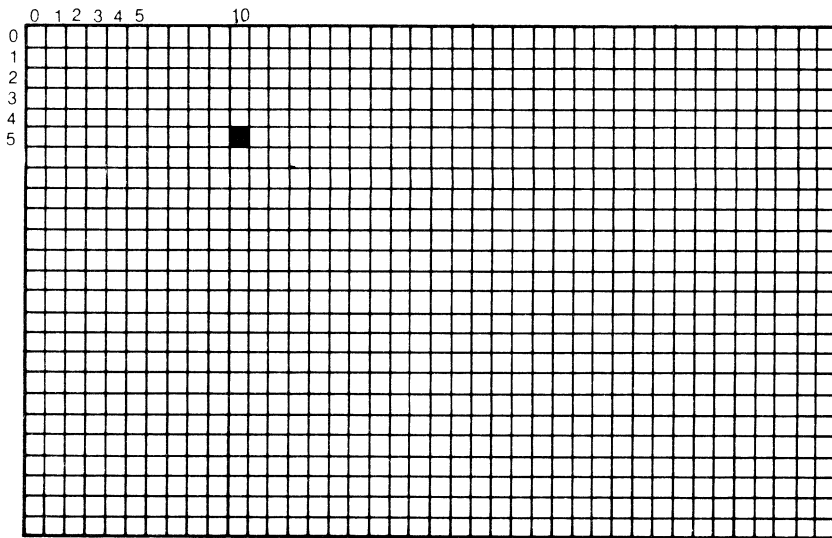


To find any location on the screen, use the following formula:

$$\text{Position} = \text{Start of Screen} + (40 * \text{Row} + \text{Column})$$

$$\text{or: } P = 1024 + (40 * R + C)$$

For example:



The location shaded is in row 5 and column 10. Using the formula, you will get:

$$P = 1024 + (40 * 5 + 10)$$

$$P = 1024 + (200 + 10)$$

$$P = 1024 + (210)$$

Count the number of squares from the upper left corner starting with zero (0). The shaded square will indeed be 210 units beyond the starting position.

We can use the same method to calculate the POKE location of a position on the screen:

$$55296 + (40 * R + C)$$

Now, let's use our new method to fill the screen with hearts (POKE code 83).

**NEW**

```
10 PRINT "♥"  
20 POKE 53281,0: REM-BLACK BACKGROUND  
30 FOR R = 0 TO 24  
40 FOR C = 0 TO 39  
50 POKE 1024 + (40*R + C),83: REM-CODE FOR HEART  
60 POKE 55296 + (40*R + C),2:REM-CODE FOR RED  
70 NEXT C  
80 NEXT R  
90 END
```

**LIST**

**RUN**

Line 30 tells the computer to start at Row 0. The loop created by line 40 through line 70 fills each Column in that row. Line 80 sends the Commodore 64 back to line 30 to repeat the process with the remainder of the rows.

We can also start with Column 0 and fill each Row in that column before going to the next. You may wish to clear the screen by pressing SHIFT CLR before you continue.

```

NEW
10 PRINT "☺"
20 POKE 53281,0
30 FOR C = 0 TO 39
40 FOR R = 0 TO 24
50 POKE 1024 + (40*R + C),83
60 POKE 55296 + (40*R + C),2
70 NEXT R
80 NEXT C
90 END
LIST
RUN

```

You can make each column a different color by adding:

```

35 LET V = INT(RND(1)*15)+1: REM-RANDOM COLOR
   CODE 1-15
60 POKE 55296 + (40*R + C),V
LIST
RUN

```

*Things to Notice:*

- If you get a ?NEXT WITHOUT FOR ERROR check lines 70 and 80 to see if you've reversed R and C.
- If you change line 90 to: 90 GOTO 90, the picture will stay on the screen without a READY message. However, your program will never end, so press the STOP key to interrupt.

## Vertical Bars

In this section, you will display columns or vertical bars on the Commodore 64 screen in a variety of ways.

Try this program first:

```
NEW
10 PRINT "□"
20 POKE 53281,15: REM-LIGHT GRAY BACKGROUND
30 REM - VERTICAL BARS FROM THE LEFT
40 FOR C = 0 TO 39 STEP 3
50 FOR R = 0 TO 24
60 POKE 1024 + (40*R + C),200: REM-CODE FOR
  BAR CHARACTER
70 POKE 55296 + (40*R + C),10:REM-CODE FOR
  LIGHT RED
80 NEXT R
90 NEXT C
100 END
LIST
RUN
```

The loop created by lines 50 and 80 determines how long each bar will be. The loop formed by lines 40 and 90 sets the distance between the bars. "STEP 3" in line 40 tells the computer to move over three columns before drawing the next bar.

Experiment with STEP in line 40. The larger the STEP, the greater the distance between the bars.

```
40 FOR C = 0 TO 39 STEP 5
RUN
```

Change the program once more.

```
40 FOR C = 0 TO 39 STEP 2
RUN
```

Now, let's fill the empty spaces starting at the right side of the screen. Clear the screen, LIST, and add the following lines to your program.

```
100 REM - VERTICAL BARS FROM THE RIGHT
110 FOR C = 39 TO 0 STEP -2
120 FOR R = 0 TO 24
130 POKE 1024 + (40*R + C),196: REM-ANOTHER
    BAR CHARACTER
140 POKE 55296 + (40*R + C),12:REM-COLOR DARK
    GRAY
150 NEXT R
160 NEXT C
180 END
LIST
RUN
```

Line 110 tells the computer to start at column 39 and subtract 2 (STEP -2) each time through the loop. Therefore, the next bar will be drawn 2 columns to the left of the last one. This continues until the left side of the screen is reached.

Clear the screen by using the SHIFT and CLR keys; clear memory by typing NEW. You are now ready to vary the color and location of the bars by another method.

```

NEW
10 PRINT " "
20 POKE 53281,15: REM-LIGHT GRAY BACKGROUND
30 LET Y = 2
40 REM - MORE VERTICAL BARS
50 FOR C = Y TO 39 STEP Y
60 FOR R = 0 TO 24
70 POKE 1024 + (40*R + C),200
80 POKE 55296 + (40*R + C),Y
90 NEXT R
100 NEXT C
110 IF Y = 10 THEN 140
120 LET Y = Y + 1
130 GOTO 40
140 REM - END OF PROGRAM
150 GOTO 150
160 END
LIST
RUN

```

Once the screen is filled and the action is complete, press the STOP key.

*Things to Notice:*

- The Y in line 50 is a variable that tells the computer where to place the first bar and the distance between the bars. The first time through the loops, Y is equal to 2 so the first bar begins at 2 and the next one is placed 2 columns away. Line 120 adds 1 to the variable so the next time through the loops the first bar starts at 3 with a distance of 3 between each bar, then 4, and so on. Line 100 stops this when Y reaches 10.
- Line 150 creates a loop so the READY message will not appear on the screen when the program is complete. Therefore, it is necessary to press the STOP key when you are ready to continue.

## Colorful Corners

Let's fill the screen another way. By alternating a row with a column, you can achieve a different effect. In this design both the row and column start from the same screen location.

```
NEW
10 PRINT "☺"
20 POKE 53281,15
30 FOR X = 0 TO 24
40 LET V = INT(RND(1)*15): REM-RANDOM
   COLOR 0-14
50 REM - MAKE A ROW
60 FOR C = X TO 39
70 POKE 1024 + (X*40 + C),127
80 POKE 55296 + (X*40+C),V
90 NEXT C
100 REM - MAKE A COLUMN
110 FOR R = X TO 24
120 POKE 1024 + (R*40 + X),127
130 POKE 55296 + (R*40+X),V
140 NEXT R
150 NEXT X
160 GOTO 160
170 END
LIST
RUN
```

To interrupt, press the STOP key.

To choose a random symbol, add and change the following lines:

```
35 LET Q = INT(RND(1)*256): REM-RANDOM
   CHARACTER 0-255
70 POKE 1024 + (X*40 + C),Q
120 POKE 1024 + (R*40 + X),Q
LIST
RUN
```

When you are ready to go on, press the STOP key.



## Kaleidoscope

A kaleidoscope delights and entertains the viewer with a constantly changing pattern. The design for this kaleidoscope originates in the upper right section of the screen and is then copied to the other three quadrants of the screen.

NEW

```
10 REM - KALEIDOSCOPE
20 PRINT "☺"
30 POKE 53281,0: REM-BLACK BACKGROUND
40 POKE 53280,0: REM-BLACK BORDER
50 FOR C = 1 TO 19
60 FOR R = 0 TO 12
70 LET V = INT(RND(1)*16): REM-RANDOM COLOR
  0-15
80 REM - UPPER RIGHT
90 POKE 1024+20 + (R*40+C),160
100 POKE 55296+20 + (R*40+C),V
110 REM - UPPER LEFT
120 POKE 1024+20 + (R*40-C),160
130 POKE 55296+20 + (R*40-C),V
140 REM - LOWER RIGHT
150 POKE 1024+20 + ((24-R)*40+C),160
160 POKE 55296+20 + ((24-R)*40+C),V
170 REM - LOWER LEFT
180 POKE 1024+20 + ((24-R)*40-C),160
190 POKE 55296+20 + ((24-R)*40-C),V
200 NEXT R
210 NEXT C
220 GOTO 50
230 END
LIST
RUN
```

Press the STOP key to interrupt.

## Elements of Sound

Sound can add another dimension of entertainment to your computer. The Commodore 64 contains the Sound Interface Device (SID) chip which can be made to simulate a variety of sounds.

Before you can hear any sound, the Commodore 64 needs to be hooked to a television set or a monitor that has an audio outlet with the proper cable plugged into it. If you are not equipped for sound, you may wish to skip to the program "Jumping Jacks," page 151.

Most sounds build to a peak (Attack), drop to an intermediate level (Decay), stabilize for a while (Sustain) and finally fade away (Release). This cycle called the ADSR can be programmed with the SID chip. The Attack and Decay rate combine by a formula into a single number which is then stored in a memory location. The Sustain and Decay rates share another. Other memory locations are used for the volume, the waveform, and the pitch of the tone.

Twenty-five memory locations, 54272 through 54296, have been set aside to control the sound for three separate tones, or "voices." We will use the six locations necessary to control voice #1.

### *MEMORY LOCATION*

### *FUNCTION*

54296	Volume
54273	High frequency of a note
54272	Low frequency of a note
54277	Attack and delay rate
54278	Sustain and release rate
54276	Waveform or vibration type

**VOLUME:**

You can control the volume by POKEing a number from 0 (off) to 15 (loudest) into the memory location 54296. The sound should always be turned off, POKE 54296,0, before ending a program that has sound. In case you forget, you can turn off the sound by using the **STOP** key with the **RESTORE** key.

**HIGH AND LOW FREQUENCIES OF A NOTE:**

The frequency of all the notes of the scale from eight octaves is included in the appendix of the *Commodore 64 User's Guide* which came with your Commodore 64. Here is a table for the scale at octave 5. These numbers should be POKEd into memory locations 54273 and 54272 for the high and low frequencies.

<i>NOTE</i>	<i>POKE VALUE FOR HIGH FREQUENCY</i>	<i>POKE VALUE FOR LOW FREQUENCY</i>
C	34	75
C# D <sup>b</sup>	36	85
D	38	126
D# E <sup>b</sup>	40	200
E	43	52
F	45	198
F# G <sup>b</sup>	48	127
G	51	97
G# A <sup>b</sup>	54	111
A	57	172
A# B <sup>b</sup>	61	126
B	64	188

**ATTACK AND DECAY RATE:**

The time it takes for a note to reach its full volume or to fade away is determined by a number from 0 (fastest rate of change) to 15 (slowest rate of change). The formula used to produce the number to be POKEd into memory location 54277 is:

$$\begin{aligned} & (\text{Attack Rate}) * 16 + (\text{Decay rate}) \text{ or} \\ & (0 \text{ to } 15) * 16 + (0 \text{ to } 15) \end{aligned}$$

**SUSTAIN AND RELEASE RATE:**

The rate to extend a note at a certain level before releasing it is determined by a number from 0 (fastest) to 15 (slowest). The number produced by the following formula is POKEd into memory location 54278.

$$\begin{aligned} & (\text{Sustain rate}) * 16 + (\text{Release rate}) \text{ or} \\ & (0 \text{ to } 15) * 16 + (0 \text{ to } 15) \end{aligned}$$

**WAVEFORM:**

The SID chip can produce four different vibrations or sound waves: triangle, sawtooth, pulse, and noise. The triangle produces a mellow sound; the sawtooth a brassy sound; the pulse, a range of sounds from sharp to nasal; and noise, a low rumbling to a hissing sound. The following is a table of waveform values to be POKEd into memory location 54276:

<i>WAVEFORM</i>	<i>POKE VALUE</i>
Triangle	17
Sawtooth	33
Pulse	65
Noise	129

Now let's write a program to hear how each waveform sounds.

```
NEW
10 REM - WAVEFORM EXPLORATIONS
20 REM - SET VOLUME TO 15
21 LET V = 15
25 POKE 54296,V
30 REM - SET HIGH FREQUENCY, C AT OCTAVE FIVE
31 LET HF = 34
35 POKE 54273,HF
40 REM - SET LOW FREQUENCY, C AT OCTAVE FIVE
41 LET LF = 75
45 POKE 54272,LF
50 REM - SET ATTACK/DECAY RATE
51 LET A = 15
52 LET D = 8
55 POKE 54277, A*16+D
60 REM - SET SUSTAIN/RELEASE RATE
61 LET S = 8
62 LET R = 8
65 POKE 54278, S*16+R
70 REM - WAVEFORM
71 LET WF = 17: REM - TRIANGLE TYPE OF
    WAVEFORM
75 POKE 54276,WF
80 REM - DURATION OF SOUND
85 FOR DU = T TO 5000: NEXT DU
90 REM - TURN OFF SOUND BY SETTING VOLUME TO
    ZERO
95 POKE 54296,0
100 END
LIST
RUN
```

In the program above, we used the triangle waveform. To change to the others:

```
71 LET WF = 33: REM - SAWTOOTH TYPE OF WAVEFORM  
LIST  
RUN
```

```
71 LET WF = 65: REM - SAWTOOTH TYPE OF WAVEFORM  
LIST  
RUN
```

```
71 LET WF = 129: REM - SAWTOOTH TYPE OF WAVEFORM  
LIST  
RUN
```

To change the duration of the sound, change line 85. The larger the number the longer the sound remains.



## Exploring Sound

The program below will give you an opportunity to more easily experiment with all the variables that are used to compose sound on the Commodore 64.

Type in the following program, and after typing RUN, answer the questions.

```
NEW
10 PRINT "Q"
20 REM - SOUND EXPLORATIONS
30 PRINT "VOLUME 1 TO 15";
40 INPUT V
50 PRINT "HIGH FREQUENCY 0 TO 255";
60 INPUT HF
70 PRINT "LOW FREQUENCY 0 TO 255";
80 INPUT LF
90 PRINT "ATTACK RATE 0 TO 15";
100 INPUT A
110 PRINT "DECAY RATE 0 TO 15";
120 INPUT D
130 PRINT "SUSTAIN LEVEL 0 TO 15";
140 INPUT S
150 PRINT "RELEASE RATE 0 TO 15";
160 INPUT R
170 PRINT "WAVEFORM TYPE 17,33,65, OR 129";
180 INPUT WV
190 PRINT "HOW MANY SECONDS TO HEAR SOUND";
200 INPUT T
210 REM - POKE LOCATIONS & PLAY SOUND
220 POKE 54296,V
230 POKE 54273,HF
240 POKE 54272,LF
250 POKE 54277,A*16+D
260 POKE 54278,S*16+R
270 POKE 54276,WV
280 FOR DU = 1 TO T*800: NEXT DU
290 POKE 54296,0
300 END
LIST
RUN
```

If you answer a question with a number outside the allowed range, you will get a ?ILLEGAL QUANTITY ERROR and the sound will not turn off. To turn off the sound, type STOP/RESTORE, or type:

**POKE 54296,0**

Run this program several times and try different numbers to get different results.

## Jumping Jacks

Cartoon artists use tricks to create the illusion of motion. Some of the same techniques can be used with the computer. To animate our figure we will draw it, erase it, and draw a slightly different figure in a new location.

To draw it in a new location we will use the TAB function. TAB moves the cursor in from the left margin of the screen before it prints its message starting in a specific column. For example, PRINT TAB(12) starts printing at column 12.

Use the following chart to help you design your figure.

SYMBOL	MEANS
▣	CTRL 8
◡	SHIFT CLR
\	SHIFT M
o	SHIFT W
/	SHIFT N
⊗	⌘ +
	⌘ G



```

NEW
10 REM - ANIMATION
20 POKE 53281,0: REM -BLACK BACKGROUND
30 PRINT "■": REM - CTRL 8 -- YELLOW
40 FOR X = 1 TO 30
50 REM - PICTURE 1
60 PRINT "□"
70 PRINT TAB(X); " 0"      Move in X
80 PRINT TAB(X); " /■\"    spaces before
90 PRINT TAB(X); " 11 "    printing.
100 REM - DELAY
110 FOR D = 1 TO 100: NEXT D
120 :
130 REM - PICTURE 2
140 PRINT "□"
150 PRINT TAB(X); " \0/"
160 PRINT TAB(X); " ■"
170 PRINT TAB(X); " / \"
180 REM - DELAY
190 FOR D = 1 TO 100: NEXT D
200 NEXT X
210 END
LIST
RUN

```

There should not be space left between the word TAB and the (X) in your program. If a space is present, a 0 will appear to the left of your figure and you eventually will get a ?BAD SUBSCRIPT ERROR.

Can you design a car or an animal and have it move across the screen?

## Draw

This program puts a dot near the center of the screen. Starting at that dot, you can draw a picture with dots by pressing the cursor keys. The line of dots will extend whichever way the arrow on the key points. To get the dot to go up or left, remember to use the SHIFT Key with the proper cursor key.

Like the key word INPUT, GET allows you to respond or interact with the computer. GET is different than INPUT, in that it does not place a question mark on the screen; it accepts only one character at a time from the keyboard; and, instead of waiting for your response, it checks the keyboard and continues to the next instruction immediately whether a response has been given or not.

SYMBOL	MEANS
␣	CRSR↓
␣	CRSR↑
␣	CRSR→
␣	CRSR←

NEW

```

10 PRINT "␣"
20 REM - DRAW
30 POKE 53281,0: REM - BACKGROUND
40 POKE 53280,0: REM - BORDER
50 LET R = 10: REM - START AT ROW 10
60 LET C = 20: REM - START AT COLUMN 20
70 LET A = 81: REM - CHARACTER IA A INPUT
80 POKE 1024+(R*40+C), A
90 POKE 55296+(R*40+C), 8
100 GET D$←
110 IF D$ = "" THEN 100←
120 REM - MOVE DOWN
130 IF D$= "␣" THEN LET R=R+1
140 IF R>19 THEN LET R=19
150 REM - MOVE UP
160 IF D$= "␣" THEN LET R=R-1
170 IF R<0 THEN LET R=0
180 REM - MOVE RIGHT
190 IF D$= "␣" THEN LET C=C+1
200 IF C>39 THEN LET C=39
210 REM - MOVE LEFT
220 IF D$= "␣" THEN LET C=C-1
230 IF C<0 THEN LET C=0
240 GOTO 80
250 END
LIST
RUN

```

Look for a key pressed at the keyboard.

No key pressed go look again.

These are cursor control keys.

To interrupt press the **STOP** key with **RESTORE**.

Can you draw your initials in block letters on the screen?

Line 100 looks at the keyboard to see if a key has been pressed. If none has, then line 110 sends the computer back to line 100 to check again. Once a response has been given, the computer checks to see what the response was and acts accordingly. In this program, the Commodore 64 is waiting for you to press **CRSR↓**, **CRSR↑**, **CRSR→** or **CRSR←**. Any other character will be ignored.

Line 130, 160, 190, and 220 increase or decrease the row or column location of the ball. Then the computer checks to see if the edge of the screen has been reached. Lines 140, 170, 200, and 230 will not let you go off the screen.

*Things to Try:*

- Change the character from a ball to a block, by changing 70 to:

```
70 LET A = 160
LIST
RUN
```

- With the following addition to your program, you can leave empty spaces. To leave spaces, press the **S** key before you use the cursor keys. To get the block back again, press the **B** key and continue as before. (Remember in line 250 that 32 is the POKE code for a blank.)

```
240 REM - DRAW WITH A BLANK SPACE
250 IF D$ = "S" THEN LET A=32
260 REM - DRAW WITH A BLOCK
270 IF D$ = "B" THEN LET A=160
280 GOTO 80
290 END
LIST
RUN
```

Experiment and draw with horizontal or vertical lines. Check the appendix for the poke codes.

## Animation with Poke Characters

Great possibilities exist for creating animation using poke characters. This section gives some idea of how it is done by making a ball fall down stairs.

First, let's create the stairs with horizontal lines making each line longer than the previous one.

```
NEW
10 PRINT "┌"
20 POKE 53281, 15
30 REM - STEPS
40 FOR R = 3 TO 24
50 FOR C = 0 TO R+5
60 POKE 1024+(R*40+C), 160
70 POKE 55296+(R*40+C), 4
80 NEXT C
90 NEXT R
LIST
RUN
```

To drop a ball down the stairs, add the following instructions to your program. Poke code 81, the ball, is POKEd onto the screen to the right of the stairs.

```
100 REM - BALL
110 LET B = 3
120 FOR C = 9 TO 30
130 POKE 1024+(B*40+C), 81
140 POKE 55296+(B*40+C), 0
150 LET B = B+1
160 NEXT C
LIST
RUN
```

Slow down the ball by adding a delay loop.

```
145 FOR D = 1 TO 200: NEXT D
RUN
```

The previous RUN left a trail. Printing a space over the “last ball” will erase it and give the illusion of motion.

```
146 POKE 1024+(B*40+C), 32
RUN
```

Roll the ball across the top of the stairs before it starts its trip to the bottom.

```
91 REM - ROLL BALL ACROSS TOP
92 FOR C = 0 TO 8
93 POKE 1024+(2*40+C), 81
94 POKE 55296+(2*40+C), 0
95 FOR D = 1 TO 200: NEXT D
96 POKE 1024+(2*40+C), 32
97 NEXT C

LIST

RUN
```

Why not speed up the ball as it goes down the stairs? Change or eliminate the delay loop in line 145.

## Bouncing Ball

One of the earliest video games featured a ball bouncing across the screen. Here is one way you can do that with your Commodore 64.

SYMBOL	MEANS
◻	SHIFT CLR
◼	CRSR↓
◻	CRSR↑
◼	CRSR←
◼	CRSR→

NEW

```
10 PRINT "◻"  
20 POKE 53281,0  
30 PRINT "◼◼BOUNCING BALL"  
40 LET R = 10  
50 LET C = 1  
60 LET DR = 1  
70 LET DC = 1  
80 FOR T = 1 TO 300  
90 POKE 1024+(R*40+C), 81  
100 POKE 55296+(R*40+C), 7  
110 FOR D = 1 TO 10: NEXT D  
120 POKE 1024+(R*40+C), 32  
130 LET R = R + DR  
140 IF R = 0 OR R = 24 THEN LET DR = -DR  
150 LET C = C + DC  
160 IF C = 0 OR C = 39 THEN LET DC = -DC  
170 NEXT T  
180 END  
LIST  
RUN
```

Lines 40 and 50 tell the beginning location of the ball; Row 10, Column 1. Lines 60 and 70 say that the direction of the ball's movement in a row, DR, and the direction in the column, DC, will be 1. With this information, lines 130 and 150 set the location for the next ball. Line 140 says that if the ball reaches the top or the bottom of the screen to reverse direction, LET DR = - DR. Line 160 changes direction if the ball reaches the extreme right or left of the screen.

You can make this program into a simple game. Notice that some of the letters in the title are erased as the ball goes over them.

The game is to figure out where you can start the ball so that the most letters will be knocked out from the title, BOUNCING BALL, by the end of 300 moves.

To do that you will need to change the initial position of the ball. Line 40 tells what row you begin in and line 50 tells the column. For example:

```
40 LET R = 15
50 LET C = 10
LIST
RUN
```

Try several different starting positions to find one that will knock out the most letters.

You can also increase the number of times the ball moves in the game by changing line 80.

```
80 FOR T = 1 TO 500
LIST
RUN
```

Once you have discovered a winning strategy, use this program with a friend.

## Critter

The last program in this chapter gets a critter moving across the screen.

```
NEW
10 PRINT "  "
20 POKE 53281,0
30 REM - CRITTER
40 LET R = 12
50 FOR C = 0 TO 39
60 POKE 1024+(R*40+C), 90
70 POKE 55296+(R*40+C), 5
80 FOR D = 1 TO 150: NEXT D
90 REM - LEAVE A TRAIL
100 POKE 1024+(R*40+C), 160
110 POKE 55296+(R*40+C), 7
120 NEXT C
130 END
LIST
RUN
```

Erase the critter's trail.

```
100 POKE 1024+(R*40+C), 32
LIST
RUN
```

Now, let's make that critter really move.

```
80 FOR D = 1 TO 25: NEXT D
LIST
RUN
```

Best wishes as you move along to your next computing adventure!



## Where Do You Go From Here?

You are now familiar with the Commodore 64. You have been introduced to many of the key words in BASIC, and to loops, random numbers, and how to work with designs, color and sound. Now, what can you do with your knowledge?

If you enjoyed this book, you will want to keep on learning. There are several ways to learn more about your Commodore 64.

1. Write your own programs. It can be a lot of fun to discover how to do new things with knowledge you already have. **EXPERIMENT!**
2. Talk to other people who like computers. There are groups for people who use the Commodore 64, where they can share their experiences and knowledge. Magazines for the Commodore 64 often have a list of Users' Groups around the country. There is probably one in your area.
3. If you like games, there are many books and magazines that contain BASIC game programs. Borrow or buy one and look through it. If you see a game you think would be fun to play, try it out. But don't just type it in. Try to figure out why and how it works. This way you will learn more about the BASIC language and more ways to use the statements you already know.
4. There are several magazines that are devoted to Commodore computers. They contain many articles as well as useful programs that can be typed into your C64.
5. When you received your computer, the *Commodore 64 User's Guide* came with it. Use the guide to learn new information about Sprites, High Resolution (HI-RES) Graphics, and Three-Part Sound. These take quite a bit of exploration and work, but are worth the effort.

6. Finally, use and enjoy programs written by other people. You can buy many useful programs for your Commodore 64 and there are many programs available FREE from Users' Groups. For more information about Users' Groups and their libraries of free software, you might try "*Free Software for Your Commodore*" by Dave and Dorothy Heller (ENRICH/OHAUS, 1984).

There are many more adventures ahead using your Commodore 64. HAPPY COMPUTING!!

## **Vocabulary**

random numbers

IF...THEN

POKE

address

screen memory

color memory

column

row

PEEK

sound

animation

GET

Users' Groups

# APPENDIX

## 1. How To Use The Background/Foreground Combination Chart

When most Commodore 64's are first turned on, the TV displays light blue printing on a dark background. Chapter 2 explains how the colors of the background, the letters, and the border around the background, can all be changed very easily by using special POKE commands. So, if you don't like light blue print on a dark background, or if it seems hard to read on your TV set, you can try other color combinations.

You can make a chart of which color combinations look nice on your TV, by following these instructions. Remember that you can always go back to "normal" just by pressing STOP and RESTORE at the same time.

1. Type a POKE instruction from the side of the chart on page 162 into the Commodore 64.
2. Type a POKE instruction from the bottom of the chart.
3. Type **PRINT "YOUR NAME"**
4. Is YOUR NAME easy to read? Do you like the way it looks? Put a "Y" or a "N" in the grid at the intersection of the 2 statements used.

















For example:

```
POKE 53281,0
POKE 646,7
PRINT "YOUR NAME"
```

Put a "Y" in the appropriate location on the grid. (This one has already been recorded for you.)









## 2. Color Chart Using CTRL or ⌘ Keys

COLOR	SYMBOL	KEY
BLACK		CTRL 1
WHITE		CTRL 2
RED		CTRL 3
CYAN		CTRL 4
PURPLE		CTRL 5
GREEN		CTRL 6
BLUE		CTRL 7
YELLOW		CTRL 8
ORANGE		⌘ 1
BROWN		⌘ 2
LIGHT RED		⌘ 3
GRAY 1		⌘ 4
GRAY 2		⌘ 5
LIGHT GREEN		⌘ 6
LIGHT BLUE		⌘ 7
GRAY 3		⌘ 8

### 3. Color Chart Using POKE

<u>COLOR</u>	<u>POKE CODE</u>
BLACK	0
WHITE	1
RED	2
CYAN	3
PURPLE	4
GREEN	5
BLUE	6
YELLOW	7
ORANGE	8
PINK	9
LIGHT RED	10
GRAY 1	11
GRAY 2	12
LIGHT GREEN	13
LIGHT BLUE	14
GRAY 3	15

### 4. Key to Special Symbols

<u>SYMBOL</u>	<u>MEANS</u>	<u>KEY</u>
	HOME	<b>CLR/HOME</b>
	CLEAR HOME	<b>SHIFT CLR/HOME</b>
	CRSR↓	<b>CRSR↑↓</b>
	CRSR↑	<b>SHIFT CRSR↑↓</b>
	CRSR→	<b>CRSR←→</b>
	CRSR←	<b>SHIFT CRSR←→</b>

## 5. Screen Poke Code (Upper Case Graphics Mode)

This chart shows all the characters available on the Commodore 64. Any character may also be displayed in REVERSE.

The first column of numbers is the code necessary to use in a POKE statement to get the desired character. The second column of numbers is obtained by adding 128 to the first column and is the POKE code for the reverse character.

*	**	***	*	**	***
@	0	128	S	19	147
A	1	129	T	20	148
B	2	130	U	21	149
C	3	131	V	22	150
D	4	132	W	23	151
E	5	133	X	24	152
F	6	134	Y	25	153
G	7	135	Z	26	154
H	8	136	[	27	155
I	9	137	£	28	156
J	10	138	]	29	157
K	11	139	↑	30	158
L	12	140	←	31	159
M	13	141	SPACE	32	160
N	14	142	!	33	161
O	15	143	"	34	162
P	16	144	#	35	163
Q	17	145	\$	36	164
R	18	146	%	37	165

- \* CHARACTER
- \*\* POKE NUMBER
- \*\*\* REVERSE POKE NUMBER

## Screen Poke Code (Upper Case Graphics Mode)

*	**	***	*	**	***	*	**	***
&	38	166		68	196		98	226
'	39	167		69	197		99	227
(	40	168		70	198		100	228
)	41	169		71	199		101	229
*	42	170		72	200		102	230
+	43	171		73	201		103	231
,	44	172		74	202		104	232
-	45	173		75	203		105	233
.	46	174		76	204		106	234
/	47	175		77	205		107	235
0	48	176		78	206		108	236
1	49	177		79	207		109	237
2	50	178		80	208		110	238
3	51	179		81	209		111	239
4	52	180		82	210		112	240
5	53	181		83	211		113	241
6	54	182		84	212		114	242
7	55	183		85	213		115	243
8	56	184		86	214		116	244
9	57	185		87	215		117	245
:	58	186		88	216		118	246
;	59	187		89	217		119	247
<	60	188		90	218		120	248
=	61	189		91	219		121	249
>	62	190		92	220		122	250
?	63	191		93	221		123	251
	64	192		94	222		124	252
	65	193		95	223		125	253
	66	194		96	224		126	254
	67	195		97	225		127	255

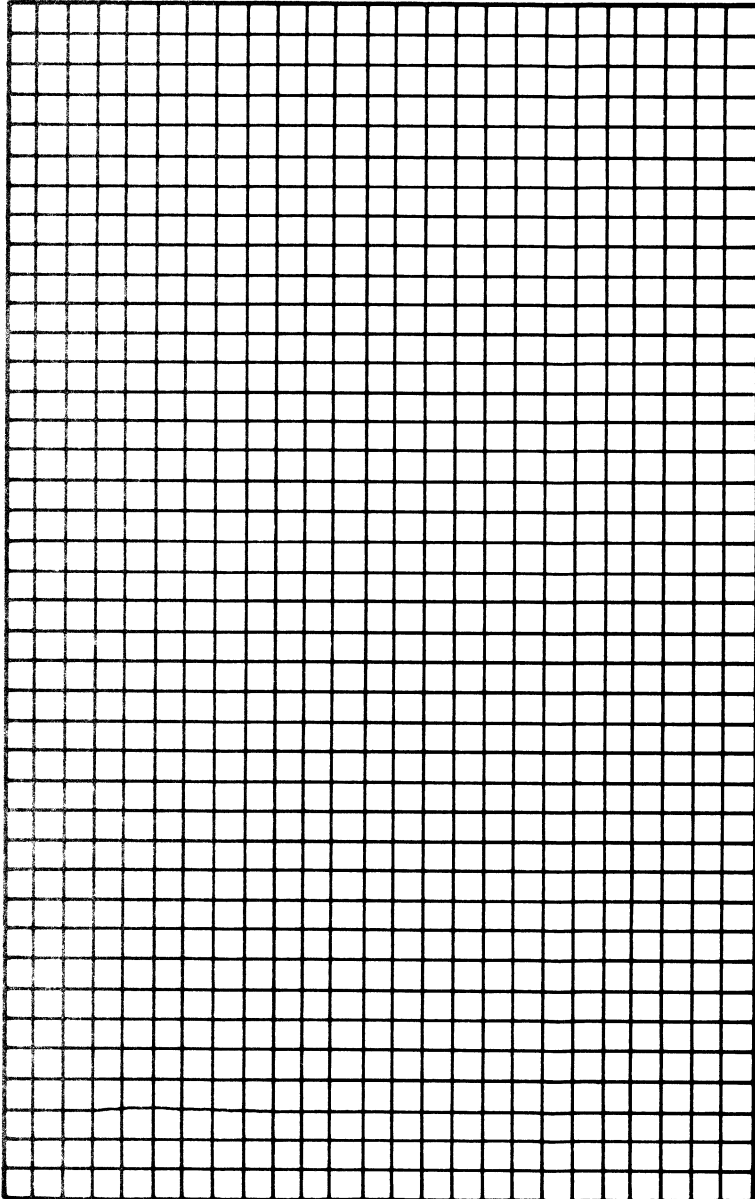
\* CHARACTER

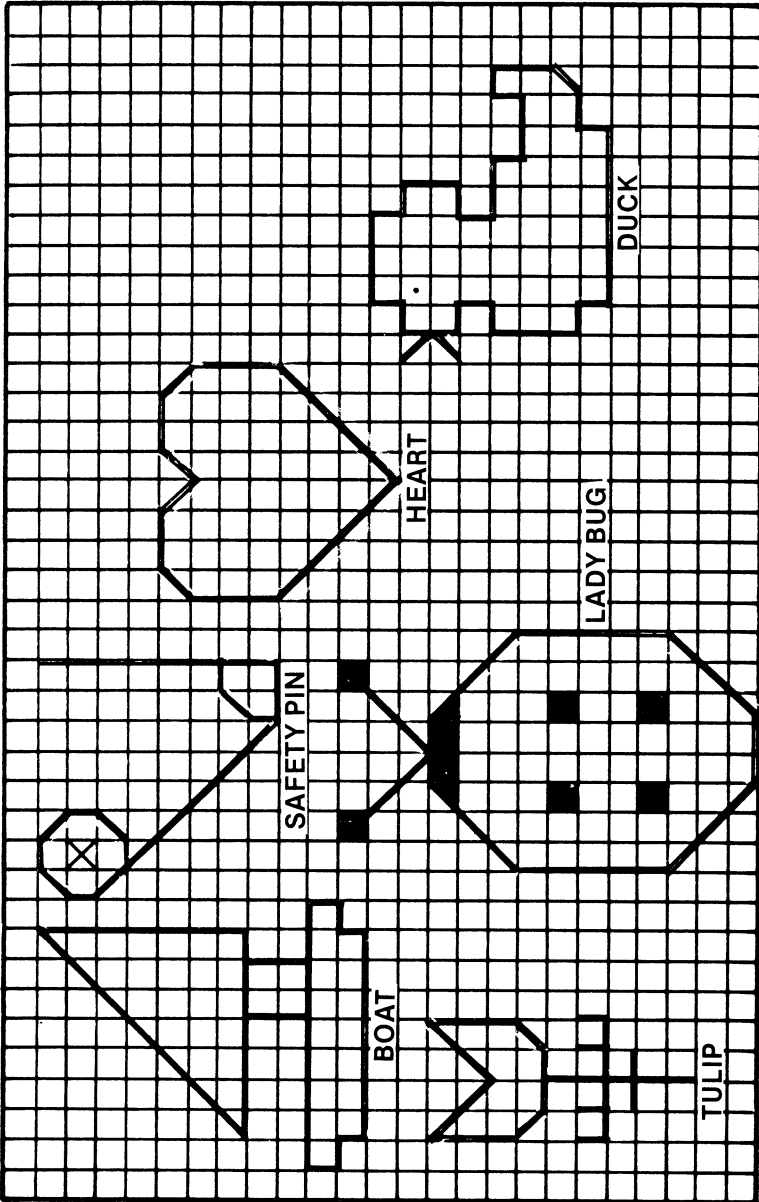
\*\* POKE NUMBER

\*\*\* REVERSE POKE NUMBER



**6. DESIGN A PICTURE FOR THE  
COMMODORE 64**



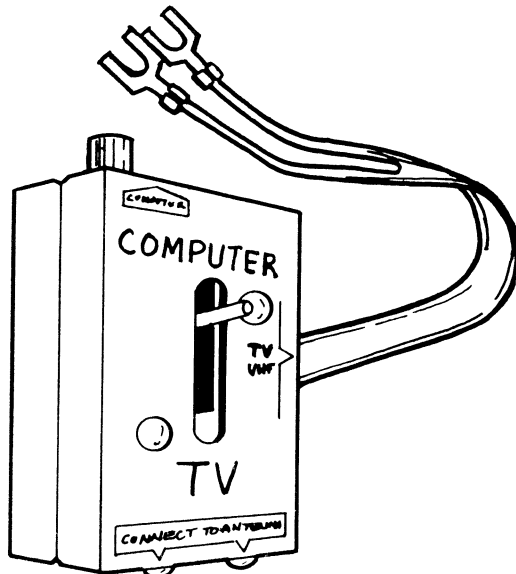


## 7. Connecting the TV

Your Commodore 64 must be connected either to a television set or to a monitor.

A “monitor” is a special, high-quality TV screen. If you bought a monitor for your C64, you should also have received a special connecting cable for it. Plug one end into the socket on the back of the Commodore which is marked “video connector” in the diagram. Plug the other end into the monitor.

If you are using an ordinary television set, you will need a screwdriver to connect your computer. Find the little silver box with a switch sticking up from it that came with your Commodore 64. This switchbox also has a short wire coming out of it. Most video games and home computers come with a switchbox like this. If you have never seen one attached to a TV, it might be a good idea for you to ask your dealer or a friend to show you how it is connected. If you already have a video game, you can probably use the same switchbox for your C64.



Use the screwdriver to connect the switchbox to your TV. It is attached the same place a pair of "rabbit ear" antennae would be. Many televisions have two pairs of antenna screws, labeled VHF and UHF. Make sure the switchbox gets connected to the ones labeled VHF. If you have an antenna or rabbit ears already connected, remove them, and attach their wires to the side of the switchbox instead. (If you have a circular loop antenna attached to the UHF screws you can leave it alone.) If you have any difficulty, please talk to your dealer. He or she should be glad to help you get your Commodore 64 up and running.

That was the hard part! The rest is easy. Now that you have the switchbox connected to your TV, find the "video cable" that came with the C64. It is just a long piece of wire with a small metal rod sticking out of each end. Plug one end firmly into the switchbox on the TV. Of course, the other end plugs into the computer. At the back of the computer, one of the small round sockets is the right size. On page 6 this is labeled "TV connector."

It takes a bit of effort to connect your Commodore 64 to your TV the first time. However, once you've hooked up the switchbox, you can easily switch between watching television and using your computer. When the switch is in "TELEVISION" position, your TV can use its antenna normally. When the switch is in "COMPUTER" position, the TV is getting its signal from the Commodore 64.

# GLOSSARY

**algorithm:** a sequence of actions which solve a problem or result in a decision.

**argument:** the variable or expression that a function needs as information. For example, in the function  $ABS(X)$ ,  $X$  is the argument of the function.

**assignment:** the storing of a value in a variable, using the assignment instruction (variable = value).

**auto-repeat:** the ability of certain keys on the keyboard to repeat themselves quickly when held down.

**BASIC:** Beginner's All-purpose Symbolic Instruction Code. A programming language, used by most personal computers.

**bug:** an error in a computer program.

**cartridge:** a small plastic box containing a program in the form of a memory chip. When the cartridge is plugged into a computer, its program automatically starts running.

**character:** any symbol that can be found on the computer keyboard such as a letter, digit or punctuation mark.

**color memory:** the part of the computer that stores information about the color of each character on the screen.

**CPU:** Central Processing Unit. The part of the computer that interprets instructions and carries them out; the "core" chip of the computer.

**CRT:** Cathode Ray Tube. A TV picture tube used to show information.

**cursor:** the blinking square ( ■ ) that shows where the next character to be typed will appear on the screen.

**data:** information processed by the computer.

**debugging:** the process of finding and correcting the mistakes in a computer program (see “bug”).

**directory:** the list of all programs stored on a disk.

**disk, diskette:** a flexible, plastic disk with a magnetic surface used to save programs and files.

**disk drive:** a machine that reads and records on disks and diskettes; one kind of input or output equipment.

**DOS:** Disk Operating System. A set of computer programs that transfer information to and from a disk.

**editing:** the process of changing or correcting a program.

**execute:** to RUN a computer program.

**friendly:** see user-friendly.

**floppy disk:** see disk.

**function:** a small built-in routine that allows special actions to be performed during a BASIC program.

**function keys:** special keys on the keyboard. They do not type any particular character, but instead are reserved for use in programs for special purposes.

**graphics:** designs and patterns. A way of showing information by pictures rather than by words.

**graphics characters:** special picture-like characters provided on some computers. They can be combined into full graphics displays.

**graphics mode:** ordinary letters printed as upper-case, and shifted letters printed as graphics characters. Opposite of “text mode”. The Commodore switches between graphics and text modes when you press the **C** and **SHIFT** keys at the same time.

**hardware:** the computer and its electrical parts, as opposed to computer programs (software).

**HI-RES:** High Resolution. A picture drawn with very tiny dots of light, so as to appear smooth and photograph-like.

**increment:** to increase the current value of a numeric variable (usually by one).

**infinite loop:** a loop that has no way of ending.

**input:** information put into the computer.

**I/O:** input and/or output.

**instructions:** directions for the computer to follow.

**jump:** a change in the usual order in which instructions are followed; a branch. In BASIC this is signalled by a GOTO statement. It may also be implied in an IF . . . THEN statement.

**keyboard:** the part of the computer that operates like a typewriter to give input to the computer.

**key word:** a BASIC vocabulary word.

**line number:** the number in front of the key word in a program statement. The computer will normally follow instructions in order of line numbers.

**LIST:** the command that displays the program lines stored in memory.

**LOAD:** the command to the computer to read information from an outside peripheral, such as a disk drive or tape player. The information is put into the computer's memory unit.

**logic error:** a mistake in constructing a program, so that it does not operate as intended.

**loop:** a set of instructions that repeats.

**LO-RES:** Low Resolution. A picture drawn with small colored rectangles, which appears blockish and of lower quality than a high resolution picture. (see HI-RES.)

**memory:** the part of the computer that stores information.

**mode:** an operating state of a computer, a method of working. Examples are program mode and immediate mode, also text mode and graphics mode.

**modem:** a machine used to allow a computer to receive or send information over a telephone line.

**monitor:** a very high-quality TV screen for computer use.

**nesting (loop nesting):** the method of placing one program loop inside another.

**NEW:** the command that erases the computer's RAM memory.

**output:** the information or results coming out of the computer.



**peripheral:** any machine which plugs into a computer to make it more fun, useful, or powerful. Examples are disk drives, printers, joysticks, speakers, and voltmeters.

**PRINT:** a key word that tells the computer to put something on the video screen.

**program:** the set of instructions that tell the computer what to do step by step; also, to write the instructions that tell the computer what to do.

**prompt:** a symbol or question on the screen which requests input.

**RAM:** Random Access Memory. The part of the computer's memory that stores information temporarily. The computer user puts information into this memory.

**REM:** a BASIC key word short for remark.

**ROM:** the part of the computer's memory that contains built-in instructions. ROM is permanent memory and cannot be erased or changed.

**routine:** a portion of a program. It usually executes a single algorithm.

**RUN:** the command that tells the computer to follow the program instructions stored in memory.

**SAVE:** the command to the computer to record information on disk or tape. SAVED information can be used later.

**screen memory:** the part of the computer that stores information about each character in each position on the screen.

**software:** computer programs, as opposed to computer machinery (hardware).

**sprite graphics:** an advanced method of making graphics pictures on the TV screen. Each “sprite”, or picture, can be made to move around independently, and they can pass in front of and behind each other.


**string:** a series of characters enclosed in quotation marks.

**subroutine:** a self-contained set of instructions used outside of the main program.

**syntax error message:** the way the computer responds when it does not understand an instruction. It is usually caused by an error in spelling or punctuation.

**tape:** cassette tape used to store programs and file magnetically.

**text characters:** the digits and letters in a program, as opposed to the graphics characters.

**text mode:** ordinary letters printed as lower-case, and shifted letters printed as upper-case. Opposite of “graphics mode”. The Commodore switches between graphics and text modes when you press the  and **SHIFT** keys at the same time.

**user-friendly:** a term describing certain programs and computers which are easy and enjoyable for non-experts to use. Also, a goal for good programs.

**variable:** a location in memory that is changeable or may contain different values.

**VERIFY:** to confirm that a program has been correctly **SAVED** on tape or disk.





## Computer Books

\$8.95

from

**ENRICH/OHAUS®**  
The Good Idea Company

Finally, microcomputer handbooks you can understand. Written in everyday language for the beginning beginner, these handbooks...

- Start at the absolute beginning
- Provide the information to make you "Computer Literate"
- Guide you as you discover what your computer can *really* do
- Give you "hands-on" experiences so that you can write your own programs
- Put you in complete control

**FUNCTIONAL TOO!** The built-in easel allows these books to stand up at your computer for easy use!

---

*Other computer books from ENRICH:*

The Illustrated Computer Dictionary & Handbook \$9.95

Free Software for Your ATARI \$8.95

Free Software for Your Commodore \$8.95

Free Software for Your Apple \$8.95

Free Software for Your TI \$8.95

Apple for the Beginning Beginner \$8.95

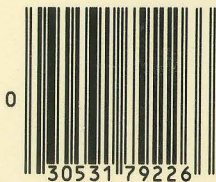
ATARI for the Beginning Beginner \$8.95

Commodore 64 for the Beginning Beginner \$8.95

PET for the Beginning Beginner \$8.95

TRS-80 for the Beginning Beginner \$8.95

TI for the Beginning Beginner \$8.95



**Ask for these books wherever  
good books are sold!**

Printed in the U.S.A.  
ISBN 0-86582-162-3