Commodore CBM Bi-Directional Printer

User's Manual

MPP-1361



commodore COMPUTER

"WARNING – THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTIONS MANUAL, MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS. IT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR A CLASS A COMPUTING DEVICE PURSUANT TO SUBPART J OF PART 15 OF FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCE WHEN OPERATED IN A COMMERCIAL ENVIRONMENT. OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE IN WHICH CASE THE USER AT HIS OWN EXPENSE WILL REQUIRED TO TAKE WHATEVER MEASURES MAY BE REQUIRED TO CORRECT THE INTERFERENCE."

Dieses Handbuch wurde gescannt, bearbeitet und ins PDF-Format konvertiert von

Rüdiger Schuldes

schuldes@itsm.uni-stuttgart.de

(c) 2003

Commodore CBM Bi-Directional Printer

User's Manual

MPP-1361

Part Number 251198

April 1983

COMMODORE BUSINESS MACHINES, INC.

TABLE OF CONTENTS

Page No.

Section 1: GENERAL INFORMATION 1
DESCRIPTION 1
Printing Characteristics 1
IEEE–488 Interface 3
Paper feed Mechanisms 3
Feed Button 3
Printer Ribbon 3
Internal Microprocessor System 3
Specifications 3
UNPACKING YOUR PRINTER 5

Section 2: PREPARING TO USE YOUR PRINTER 6
CONNECTING THE PRINTER TO YOUR COMPUTER 7
PERFORMING THE POWER–ON TEST 8
Printer Self–Test 9
Error Indicator 9
INSTALLING THE PAPER 9
REMOVING THE PAPER 10
PERFORMING THE PRINT HEAD TEST 10

Section 3: USING YOUR PRINTER 11
SPECIAL PRINTER–ASSOCIATED COMMANDS 12
The OPEN Command 12
The CMD Command 12
The PRINT Command 13
The CLOSE Command 13
PRINTING IN THE DIRECT MODE 14
PRINTING UNDER PROGRAM CONTROL 14
FORMAT CONTROL 19
Printing Data Exactly As Received: **sa=0** 20
Printing Data according to a Previously Defined Format : **sa=1** 20
Storing the Formatting Data: **sa=2** 21
The Formatting Characters 21
Numeric 21
Alpha (String Data) 24
Skip (Blank) 24
Literals in Format Strings 24
Setting the Number of Lines Per Page: **sa=3** 25
Enabling the Printer Diagnostic Messages: **sa=4** 26
Diagnostic Messages 26
Defining a Programmable Character: **sa=5** 27
Setting Spacing Between Lines: **sa=6** 29
Setting Upper or Lower Case: **sa=7** 29
Selecting ASCII/Graphics: **sa=8** 30
Suppress Diagnostic Message Printing: **sa=9** 30

Resetting the Printer: **sa = 10** 30
Setting Uni-Direction Mode: **sa = 11** 30
Resetting Uni-Direction Mode: **sa = 12** 31
Setting Condense Mode: **sa = 13** 31
Resetting Condense Mode: **sa = 14** 31
Setting Pseudo Letter Quality Mode: **sa = 15** 31
Resetting Pseudo Letter Quality Mode: **sa = 21** 32
Storing Bit Image Data: **sa = 17** 32
Printing Bit Data Previously Written **sa = 18** 33
SPECIAL CHARACTER FUNCTIONS 33
Enhanced Characters 33
Paging 34
Specifying Individual Upper- and Lower-case Characters 34
Reversing a Field 36
The Carriage Return 36
Quotation Marks 36

LIST OF TABLES

Table

1 MPP-1361 Printer Specifications 4
2 Formatting Examples 23
3 Special Control Character Summary 35

LIST OF ILLUSTRATIONS

Figure

1 Feed Mechanism and Ribbon Cartridge 2
2 The Printer Cartridge 6
3 Printer to Computer Hookup 7
4 Multiple Hookup 8

APPENDIX A A1
Simple String Formatting A2
Leading Zeroes Forced A4
Signed Numbers A5
Decimal Fraction Specification A6
Fixed and Floating $ Fields A7
Mixed Format Field Types A8
Formatting With Literals A9

APPENDIX B B1
ASCII Character Codes B1

APPENDIX C C1
SAMPLE PROGRAM OF BIT IMAGE C1

Section 1

GENERAL INFORMATION

With the purchase of a Commodore CBM Printer, you have added a great deal of versatility and convenience to the use of your Commodore Computer. You have purchased the Tractor Feed, Friction Feed Bi-Directional MPP-1361. While this manual contains all the information you need to check out, connect, and operate your printer, you should also refer to other Commodore manuals to get the most out of your computer system.

> NOTE TO BUSINESS KEYBOARD USERS ONLY: If you wish to duplicate the examples in this manual and you are using the CBM Business Keyboard computer, you should type the BASIC command POKE 59468, 12 at the beginning of each session with your computer and printer.

DESCRIPTION

Your CBM printer is designed to operate through software control. It prints upper- and lower-case alphabetic characters, all the graphic characters available from your Commodore computer, and even custom user-defined characters. Additionally, your printer has considerable formatting capability owing to its own internal microprocessor system.

PRINTING CHARACTERISTICS

Your printer employs a Brother print head. The eight-wire heavy-duty jeweled head has a life expectancy of 100 million characters. If a print head malfunction should occur, obtain authorized technical assistance. Failure to do this may void your warranty.

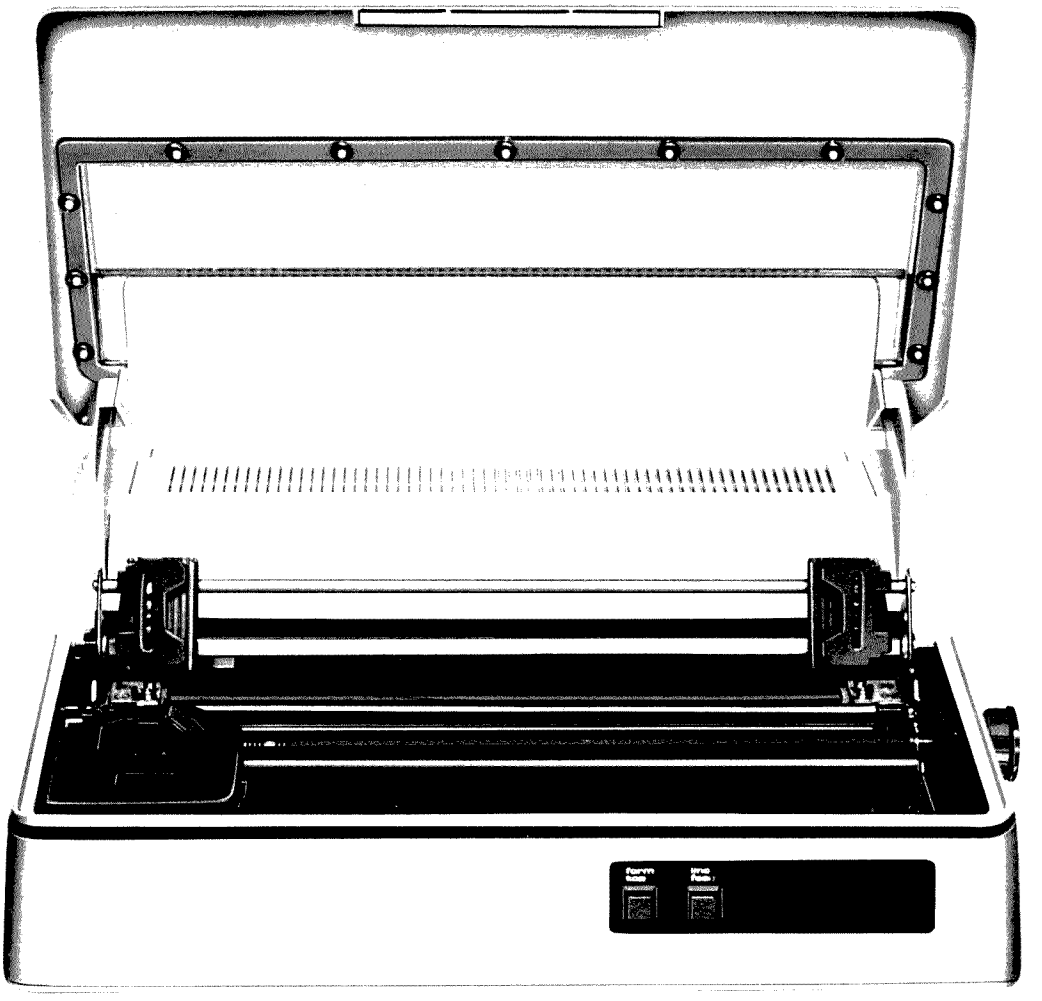


Figure1. Feed Mechanism and Ribbon Cartridge

IEEE-488 Interface

The MPP-1361 Printer conforms to IEEE-488 interface requirements and may be connected directly to your Commodore computer. The printer is designed to be used with the Commodore CBM Floppy Disk Drives and may be daisy-chained with other IEEE-488 devices.

Paper Feed Mechanisms

The MPP-1361 Printer employs both a tracter feed and a friction feed mechanism. You can use either pin-fold paper or no pin-fold paper. To select a tracter feed or a friction feed, the Feed Select Lever (on the left side of mechanism) is used.

FEED Button

The MPP-1361 Printer has two FEED BUTTON. One is the FORM TOP button, the other is the LINE FEED button. FORM TOP button feeds one page of paper, and LINE FEED button feeds one line per one push.

Printer Ribbon

The printer ribbon (shown in Figure 2) is an easily-loaded cartridge.

Internal Microprocessor System

Your printer contains a microprocessor system that resets and executes a diagnostic and initialization sequence when power is applied. In addition, it contains a random access memory (RAM) in which you can store formatting data. Because your printer is an "intelligent" peripheral, it uses none of your computer's memory.

Specifications

Table 1 shows the specifications for the MPP-1361 Printer.

TABLE 1

MPP-1361 Printer Specifications

Printing Method	Serial Impact Dot Matrix
Print Rate	150 character per second 40 lpm with 136 columns printed 75 lpm with 68 columns printed 110 lpm with 34 columns printed
Print Direction	Bi-directional with logic seek
Column Capacity	normal 136 characters condense 250 characters
Character Font	5 × 8 dot matrix
Column Spacing	1/10" – 10 characters per inch
Line Spacing	Programmable (72 step per inch)
Character Size	0.116″ high, 0.08″ wide
Copies	3, including original
Ribbon Type	Cartridge
Ribbon Life	1 million characters
Paper Width	15″ computer folded paper
Forms	5″ ~ 15″ + .5 × 2 (sprocked margins) Pin to pin distance: 5" longitudinally 9.5" Laterally 5/32" diameter

UNPACKING YOUR PRINTER

Before you unpack your printer, inspect the shipping carton for signs of damage. If it appears to be damaged, be especially careful when you inspect its contents. DON'T throw away any of the packaging material until you have located all the contents of the carton! The package should contain:

1. Commodore Printer MPP-1361
2. User Manual (P/N 251198)
3. Warranty Card
4. Printer ribbon cartridge (2)
5. Power code

If any of the first five items are missing or damaged, notify your Commodore dealer immediately.

Additionally, you must obtain the appropriate cable from your Commodore dealer to connect your printer to your computer. See Section 2, page 7, for information about which IEEE cable to get.

Section 2

PREPARING TO USE YOUR PRINTER

Before using your printer, you should make sure that it is in good working condition. The following procedure includes checking for obstructions in the path of the print head or paper feed and insuring that the printer ribbon is in place and properly set:

1. Open the printer lid
2. Carefully remove any foreign material that may have strayed into the mechanism or electronics
3. Install the printer ribbon cartridge
4. Close the printer lid.

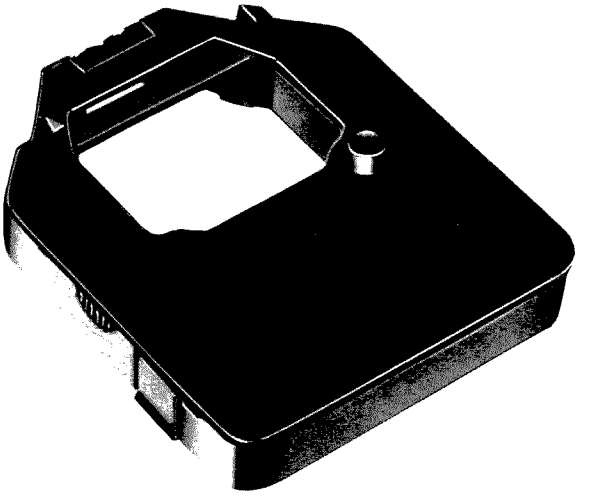


Figure 2. The Printer Cartridge

CONNECTING THE PRINTER TO YOUR COMPUTER

One of two connector cables is required to connect the printer to the computer:

1. PET-to-IEEE cable, P/N 320101
   Use this cable if the printer is to be the only IEEE device connected to your computer.

2. IEEE-to-IEEE cable, P/N 905080
   Use this cable if you will be using your printer in conjunction with the Dual Drive Floppy Disk.

Follow these steps to connect your printer to your computer:

1. Turn off the AC power to your computer.

2. Place the printer in a convenient location close to the computer. DO NOT connect the printer to an AC outlet at this time.

3. If you are using the PET-to-IEEE cable, connect the cable between the IEEE-488 interface connector on your computer and the connector on the printer as shown in Figure 3. If you are using the IEEE-to-IEEE cable, connect the printer in a configuration similar to that shown in Figure 4.

4. Connect the printer's power cord to an AC outlet. DO NOT turn on the power at this time.

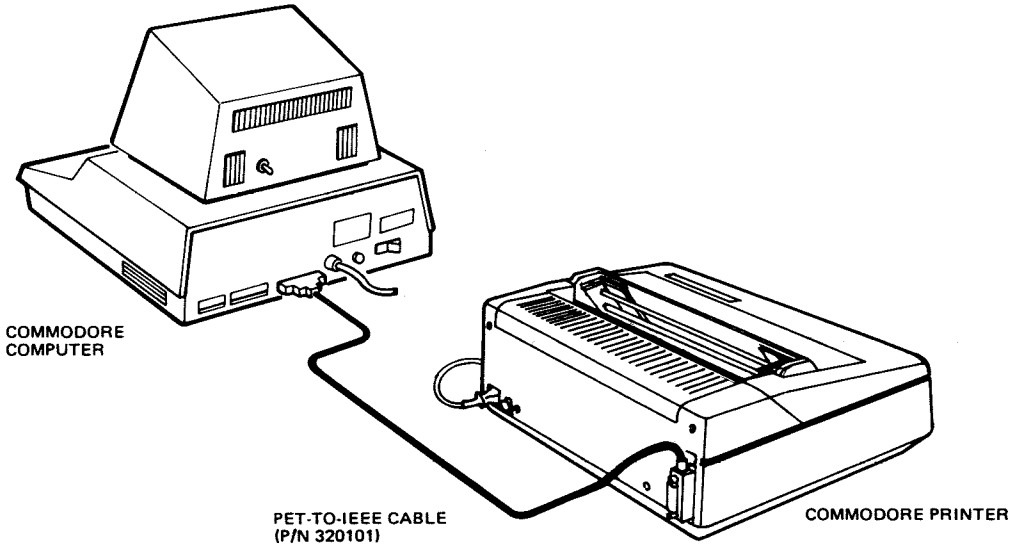


Figure 3. Printer to Computer Hookup

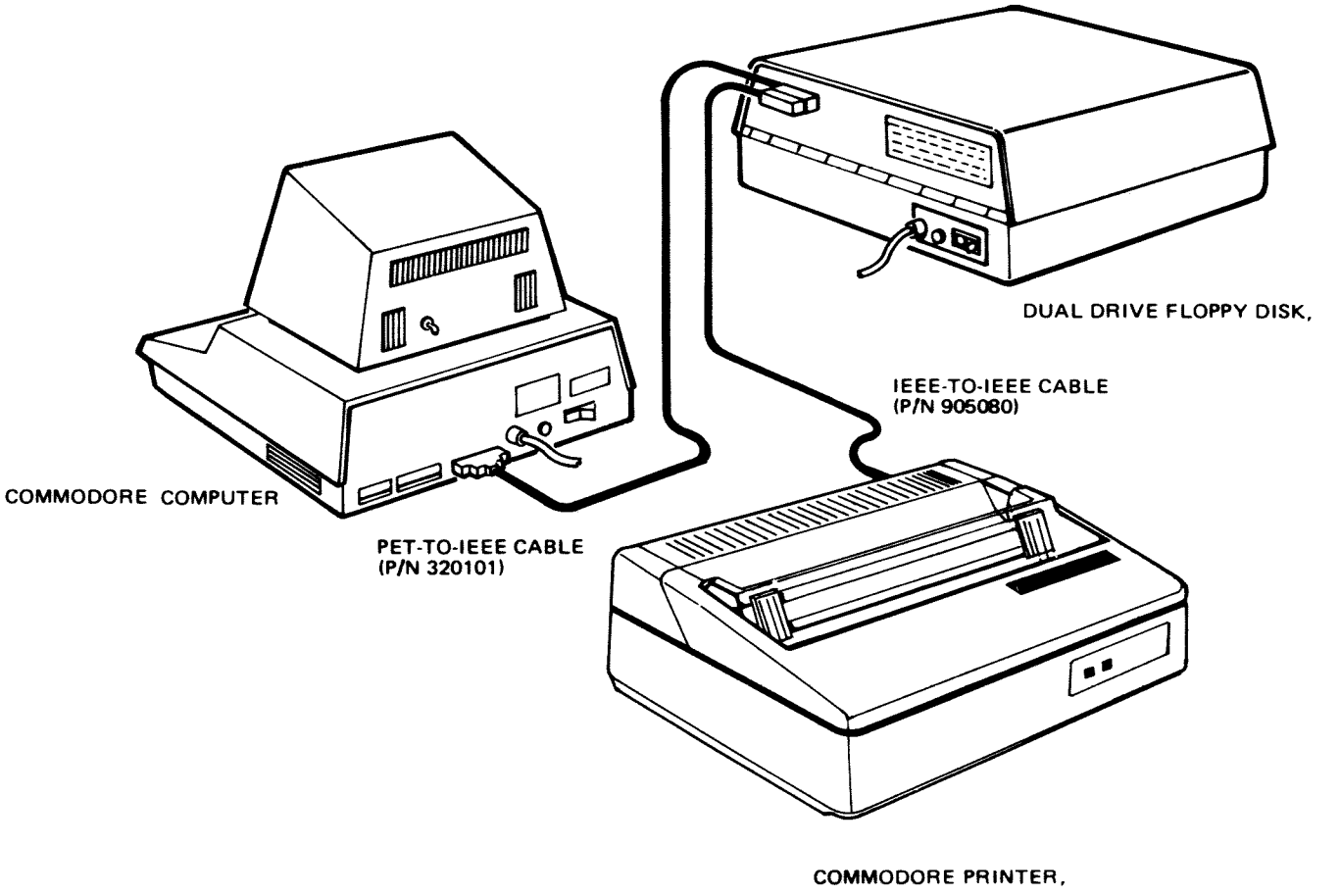


Figure 4. Multiple Hookup

PERFORMING THE POWER-ON TEST

You are now ready to proceed with the power-on part of the checkout:

1. Turn on the AC power to your computer and verify that it is working correctly.

2. To turn on the power to the printer, press the rocker switch (located toward the rear on the right side of the printer). In response to the application of AC power, the printer's microprocessor should move the print head all the way to the right, then to its home position at the far left. If this does not happen (and that's highly unlikely), turn off both machines, check all connections and try again. If you still get no response, contact your Commodore dealer.

Printer Self-Test

The printer initialization process provides a test on the RAM, ROM, and the timers. If a failure is detected, the LED will blink a specific number of times to indicate the fault. The flash code is repeated continuously with a one second pause separating each group of flashes. The following lists the flash codes:

Number of Flashes	Resource
1	Zero-page RAM
2	8K ROM
3	6522 Timer
4	6532 Timer
5	6532 Timer

Error Indicator

After the printer passes the initialization self-test, the LED is used as a power-on and failure indicator. If the LED flashes at a high rate (approx. 8/sec.), the carriage has failed to reach the home position or the carriage has been blocked. If the LED flashes at a slow rate (approx. 2/sec.), it is an indication that the printer is out of paper, the plastic cover is lifted. In either case, normal printing can be resumed after the problem has been solved and the top-of-form button is depressed.

INSTALLING THE PAPER

Now that you have positive responses from all of the previous steps, you can load the paper.

The MPP-1361 Printer accepts both pin-fold paper and the no pin-fold paper. To load the paper, follow the procedure below.

1. Raise the Printer lid.
2. Pull the Feed Selection Lever (on the left side of mecanisms) forward.
3. Push the paper into the insertion slot from rearside.

When you use the pin-fold paper, follow the procedure below.

1. Raise the two sprocket lock levers to loosen and adjust the sprocket pin position to accommodate the paper width.
2. Engage the paper feed holes onto the feed pins, and adjust the paper tension. Push the paper holding covers and two sprocket lock levers down.

When you use the no pin-fold paper, follow the procedure below.

1. Raise the two sprocket lock levers to loosen and adjust the sprocket to the paper width.
2. Push the Feed Selection Lever to rearward.

REMOVING THE PAPER

To remove the fanfold paper, follow either of the two methods described below:

1. To disengage the paper from the paper holding mechanism, pull it forward out of the printer.

> NOTE: Do not attempt to pull the paper out in the backward direction.

2. Feed the paper out of the printer by pressing the TOP OF FORM (details are described later).

PERFORMING THE PRINT HEAD TEST

You can test the print head (and the ribbon cartridge as well) after you have inserted the paper. NEVER allow any printing to occur when there is no paper in the printer. To do so may result in damage to the print head. To perform this test, simply turn off the printer, then turn it back on while pressing the TOP OF FORM button. It is necessary to hold the TOP OF FORM down until the head starts to move. The printout will be as follows:



You know how to connect your printer to your computer, how to check it out and how to insert the ribbon and paper. You are now ready for the next step: putting your Commodore printer to work.

Section 3

USING YOUR PRINTER

Your printer does much more than just give you clean, fast copy. Since it has its own internal microprocessor system, it is very versatile. In this section you will learn how to use your printer to print out listings, program results, and graphic displays. You will learn how to use it to format data and enhance character size. You will even be able to create your own custom characters.

Before you attempt to use your printer, make sure you know how to do the following:

1. Operate your Commodore computer.
2. Do elementary programming in BASIC.
3. Write files to and from a peripheral device such as the tape cassette recorder or the CBM Floppy Disk Drive.
4. Open and close files.

You should refer to your computer User Manual for this information.

This section uses certain conventions to indicate certain actions or requirements:

Example	Description
dn	Italicized lowercase letters in a syntax indicate that you should enter something (a variable) in place of letters.
[ ]	Brackets indicates optional usage.

NOTE: The commands described in this manual apply specifically to use with the printer. Certain of the commands may follow a slightly different general form or produce different results from those described here when they are used for the computer or other peripheral devices. Consult the appropriate manual for the exact usage of these commands in other contexts.

SPECIAL PRINTER-ASSOCIATED COMMANDS

When you want to print something on your printer, essentially what you are doing is transferring the video screen function to the printer. A few special BASIC commands allows you to make this transfer. Most other BASIC considerations and rules remain the same. Always remember to press the RETURN key after you type each command.

The OPEN Command

The syntax of the OPEN command is:

OPEN *lfn, dn, [sa]*

This command sets a correspondence between a file number and a physical device. The ***lfn*** or *logical file number* may be any number you choose to assign to your file from 1 to 255. It doesn't matter which number you choose as long as you remain consistent throughout your set of commands. The ***dn*** or *device number* (also known as the primary address) refers to the device to which you wish to send the file. In the case of the printer, the number must be 4 since that is the number assigned to it at the factory.

> NOTE: If you should want to change a printer number, you can have it changed by an authorized Commodore technician. See your Commodore dealer about this. (You might want to have a printer's device number changed if you have two printers attached to your computer. You need to refer to each of them - individually; hence, the need to change one of the device numbers.)

The ***sa*** or *secondary address* is somewhat of a unique concept. It alerts the printer's microprocessor system that formatting is to occur. Secondary addresses are thoroughly discussed starting on page 19, under FORMAT CONTROL.

The CMD Command

The syntax of this command is:

*CMD **lfn***

CMD transfers control from the computer to the printer. The ***lfn*** must be the same as that in the OPEN statement with which it is associated. Unlike a PRINT command, the line or bus to the - receiving device is left open. The line or bus to the receiving device (in this case, the printer) is said to be "listening." When you give the CMD command, the printer prints READY and is - waiting for further commands. If you follow a CMD command with a PRINT or a LIST command, the output is directed to the printer.

The PRINT # Command

The syntax of the PRINT# command is:

PRINT# *lfn, data*

PRINT# works just like PRINT except that it directs output to the printer instead of the video screen. The line to the printer is closed after printing the designated data. The line or bus to the printer is said to be "unlistened." Therefore, if you have used the CMD command, it is necessary to follow it with a PRINT# command in order to close down the connection between the printer and the computer.

> NOTE: In standard Commodore BASIC, the PRINT command can be abbreviated as a question mark (?). You may not do this with PRINT#. It must always be typed out as PRINT#.

The CLOSE Command

The syntax of this command is:

CLOSE *lfn*

You should always close a file after printing from it. You may not exceed ten open files so it is well to make a habit of closing files when you are finished with them. This way you will always have the maximum number of files available for use. As you shall see later, one file may be open under several logical file numbers at one time.

Remember, since CMD does not close the line to the printer, you must always precede a CLOSE command with PRINT# in order to properly close the file.

Example:

Right

```
OPEN 5,4
PRINT#5,"HELLO THERE"
CLOSE 5
```

```
OPEN 5,4
CMD 5,"HELLO THERE"
PRINT#5:CLOSE 5
```

not

Wrong

```
OPEN 5,4
CMD 5,"HELLO THERE"
CLOSE 5
```

```
OPEN 5,4
CMD 5,"HELLO THERE"
PRINT#5,"HELLO THERE"
CLOSE 5
```

not

```
OPEN 5,4
CMD 5,"HELLO THERE"
PRINT#5,"HELLO THERE"
PRINT#5:CLOSE5
```

```
OPEN 5,4
PRINT#5,"HELLO THERE"
CMD5,"HELLO THERE"
PRINT#5:CLOSE5
```

not

```
OPEN 5,4
PRINT#5,"HELLO THERE"
CMD5,"HELLO THERE"
CLOSE 5
```

Armed with these thumbnail descriptions of what the printer-related commands do, you can now proceed to the next part of this section, which tells you how to use these commands to control your printer. You can find more detailed information about these commands in your computer User Manual.

PRINTING IN THE DIRECT MODE

The direct mode of communication allows you to enter printing commands at the keyboard.

The following example shows the entire direct mode listing process of a short BASIC program. A file containing a single BASIC statement is typed into the computer's memory. The file is opened, and the file is listed. After the file is listed, the output channel is closed and the file is closed. The printer is now out of use and the computer is ready to accept new commands.

Example:

You type:	The screen displays:	The printer prints:	Comments:
10?"THIS IS A TEST"	10?"THIS IS A TEST		You place the data in the computer's memory
OPEN 3, 4	OPEN 3,4 READY		You open the file and give it a *lfn* of 3. The 4 makes the file available to the printer.
CMD 3	CMD 3	READY	The printer is "listening"
LIST	LIST	10 PRINT "THIS IS A TEST" READY.	The program is printed on the printer. The printer is still "listening."
PRINT#3	PRINT#3 READY		Use the PRINT# command to "unlisten" the printer.
CLOSE#3	CLOSE#3 READY		You close the file so that *lfn* 3 can be used for something else.

PRINTING UNDER PROGRAM CONTROL

As we have seen, you can control the printer directly from the keyboard. You can also control the printer from within a BASIC program. In the example below, this short BASIC program is placed in the computer's memory. (It could have been placed there from the keyboard, a cassette tape, or a floppy disk):

```
READY.

 10 OPEN 3,4
 20 CMD 3
 30 PRINT"THIS PROGRAM ILLUSTRATES PROGRAM CONTROL"
 40 LIST
```

> REMEMBER THIS: The LIST command within a program terminates program execution. Then, when you are finished running a program, you must type the PRINT# command to close the channel. Then type the CLOSE command to close the file.

The RUN command is given and the resulting printout is shown here:

```
THIS PROGRAM ILLUSTRATES PROGRAM CONTROL

 10 OPEN 3,4
 20 CMD 3
 30 PRINT"THIS PROGRAM ILLUSTRATES PROGRAM CONTROL"
 40 LIST
READY.
```

A more sophisticated example shows a BASIC program that takes advantage of some of your printer's special features. The program directs the printer to generate the following:

- An enhanced heading. You will find more about this feature later.
- The entire printer character set.
- A listing of the program.

Two special character functions are used in the program: the OFF/RVS key and the CRSR Down (Cursor Down) key. You may find out more about these starting later.

A printout of a program you place in your computer's memory looks like this:

```
 10 OPEN4,4
 20 FOR I=32  TO 95     :A$=A$+CHR$(I):NEXT
 30 FOR I=160 TO 223    :B$=B$+CHR$(I):NEXT
 40 C$="■"+A$
 50 D$="■"+B$
 60 E$="■"+A$
 70 F$="■"+B$
 80 G$="■"+C$
 90 H$="■"+D$
 100 PRINT#4,CHR$(1)"MPP-1361 PRINTER CHARACTER SET"
 110 PRINT#4:PRINT#4:PRINT#4
 120 PRINT#4,A$
 130 PRINT#4,B$
 140 PRINT#4,C$
 150 PRINT#4,D$
 160 PRINT#4,E$
 170 PRINT#4,F$
 180 PRINT#4,G$
 190 PRINT#4,H$
 200 CMD4
 210 PRINT:PRINT
 220 LIST
READY.
```

Here is an explanation of each statement in the program.

Line No.	Explanation
10	Opens the printer as a file for printing.
20	Sets A$ equal to a string containing all the unshifted characters from the keyboard.
30	Sets B$ equal to a string containing all the shifted characters from the keyboard.
40	Sets C$ equal to the set of all unshifted characters printed in reverse field.
50	Sets D$ equal to the set of all shifted characters in reverse field.
60-90	Each time a PRINT statement is encountered as in lines 120-190, a Carriage Return is executed (unless a semicolon is placed at the end of the preceding PRINT statement). Therefore, each string as in lines 160-190 must begin with a Cursor Down to reset the printer to lower case.

NOTE: The use of a Cursor Down to set to lowercase and Cursor Up to reset to uppercase enables you to use upper and lowercase letters on the same line.

Printing a Cursor Down has the same effect on the printer as a POKE 59648,14 has to the computer's screen in that it prints the characters following the Cursor Down in lowercase. However, after a Cursor Up or Carriage Return is encountered, the printer will resume printing in uppercase and graphics.

60 Sets E$ = to a Cursor Down plus A$.

70 Sets F$ = to a Cursor Down plus B$.

80 Sets G$ = to a Cursor Down plus C$.

90 Sets H$ = to a Cursor Down plus D$.

100-190 PRINT #4 indicates to the computer to print to device 4 (in this case, the printer).

100 CHR$(1) increases the width of the printed character to twice its original size. This line prints the title of the program in "enhanced characters."

110 Prints three Carriage Returns on the printer.

120-190 Prints each of the strings constructed in lines 20 through 90 on the printer.

200 CMD4 transfers all commands which would normally be displayed on the screen to the printer. However, if this mode is used, care must be taken to close the file to assure proper functioning of the computer after the printout is completed.

210 Prints 2 Carriage Returns on the printer.

220 Prints out a listing of the program.

NOTE: Printing a listing from within a program terminates the program. Therefore, LIST should be placed at the end of the program.

To execute this program, simply type:

RUN

This is the result:

```
MPP-1361 PRINTER CHARACTER SET

 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[£]↑←
 !"#$%&'()*+,-./0123456789:;<=>?@abcdefghijklmnopqrstuvwxyz┼ ▒
```

```
10 OPEN4,4
20 FOR I=32  TO 95     :A$=A$+CHR$(I):NEXT
30 FOR I=160 TO 223    :B$=B$+CHR$(I):NEXT
40 C$="■"+A$
50 D$="■"+B$
60 E$="■"+A$
70 F$="■"+B$
80 G$="■"+C$
90 H$="■"+D$
100 PRINT#4,CHR$(1)"MPP-1361 PRINTER CHARACTER SET"
110 PRINT#4:PRINT#4:PRINT#4
120 PRINT#4,A$
130 PRINT#4,B$
140 PRINT#4,C$
150 PRINT#4,D$
160 PRINT#4,E$
170 PRINT#4,F$
180 PRINT#4,G$
190 PRINT#4,H$
200 CMD4
210 PRINT:PRINT
220 LIST
```

Now, to close the channel and the file, type:

PRINT#4
CLOSE4

FORMAT CONTROL

Through the printer's format control option, you can control the interpretation of data sent to the printer. The format control option allows you to print numbers in columns, set the number of lines per page, and perform other useful formatting tasks. To implement the format control option, you use the third parameter of the OPEN command.

OPEN *lfn, dn, sa*

As before, ***lfn*** is the *logical file number,* ***dn*** is the *device number (also known as the primary address), and* ***sa*** is the *secondary address.* It is in the secondary address position that you enter the format control parameter.

You can enter any of the 19 secondary addresses in your OPEN command. These are:

0 Print data exactly as received

1 Print data according to a previously-defined format

2 Store the formatting data

3 Set the number of lines per page to be printed

4 Enable the printer format diagnostic messages

5 Define a programmable character

6 Set spacing between lines

7 Upper/Lower case

8 ASCII/Graphics

9 Suppress Diagnostic Message Printing

10 Reset Printer

11 Set Uni-Direction

12 Reset Uni-Direction

13 Set Condense mode

14 Reset Condense mode

15 Set pseudo letter quality

21 Reset pseudo letter quality

17 Storing bit image data

18 Printing bit data previously written

After the appropriate OPEN statement has been transmitted, a PRINT# statement is required to transmit the secondary address information to the specified device (in this case, your printer).

Remember, that it is possible to have as many as ten files open simultaneously. This allows you to perform several formatting functions at one time on the data in the computer's memory. There are examples of the formatting capabilities described in this section in Appendix A.

Printing Data Exactly As Received: sa=0

This secondary address is the default value. Whether or not you include it in your OPEN statement, the printer prints data exactly as received. Up to 136 characters are printed on each line and if the next character is not a Carriage Return, a Carriage Return is performed automatically and the overflow characters are printed on the next line.

Example:

```
10 OPEN 5,4
20 PRINT#5,"THIS IS A TEST!"
```

Results in:

```
THIS IS A TEST!
```

Printing Data According to a Previously Defined Format: sa=1

A secondary address of 1 invokes the formatting features of your printer. The data to be printed is arrayed according to a previously specified format using sa=2. If you should transmit a string of data when sa=1 is in effect and there is no formatting data in the printer's memory, then the data string is printed exactly as it is received.

When formatting string data from the computer, a skip, CHR$(29), must be sent to delimit the end of a string being edited to a field. Leading blanks are stripped off a string; therefore, to print a blank alpha field you must transmit a shifted blank, CHR$(160). The alpha field is then right padded with blanks as shown below.

Example:

```
10 OPEN2,4,2
20 OPEN1,4,1
30 PRINT#2,"AAA   AAA   AAA"
40 PRINT#1,"ABC"CHR$(29)CHR$(160)CHR$(29)"DEF"
50 CLOSE2:CLOSE1
```

Results in:

```
ABC          DEF
```

Storing the Formatting Data: sa=2

Perhaps the most significant feature of your printer is its ability to format data. Picture formatting generates a simple one-to-one correspondence between the column position of the print line and the symbol that is to appear there. Formatting allows you to left- or right-justify columns of data, or to align numeric data on its decimal point position.

Example:

```
10 OPEN 2,4,2
20 OPEN 1,4,1
30 PRINT#2,"$$$.99"
40 PRINT#1,.05
```

Results in:

```
$.05
```

The commands in the example above transfer formatting data to the printer's internal memory for future print editing use. Once completed, READY and the blinking cursor appear on the screen.

The Formatting Characters

Specification of a format is accomplished by a set of formatting characters which, when joined, define the rules of formatting data for the printer format interpreter. The formatting characters may be grouped into three classes:

Numeric	9,Z,$,¥,S,.,-
Alpha	A
Skip	(Blank)

Fields are specified by combining these formatting characters. Up to 136/250 characters are accepted in a format string. Excess (overflow) characters are printed onto the next print line.-Table 2 contains formatting examples.

Numeric

9 - Specifies a digit position in a numeric field. If there is no digit to print in this position, then a blank is substituted.

Z - Also specifies a digit position in a numeric field. Unlike the 9, this character forces a 0 to be printed if there is no digit available for this position. This is desirable if leading zeroes are required in a numeric field.

$ - If one $ is specified, then the field is treated as a dollar amount with a fixed-sign position as follows:

```
$123
$1234.00
```

If all digit positions to the left of the decimal point are $, then the number is printed as a floating dollar sign right justified before the most significant digit.

¥ - Above $ sign may be replaced with ¥ sign in JIS version. The function will be same as those described.
S-When preceding a numeric field, the sign of the number (+ or -) is to be printed in this fixed column position.

. - Defines the position of the decimal point and is printed literally in this position.

- - Specifies a trailing sign. If the number is +, a blank is printed. A numeric field cannot have both S and -. When this is the case, only the S will be honored.

Here are some examples of format fields:

a) Field width and decimal position specified-
   99 Two-digit integer with no sign
   99.999 Five-digit real type with 3 positions to right of decimal

b) Dollar sign insertion-
   $99.99 Fixed dollar sign
   $$$.99 Floating dollar sign

c) Leading zeros forced-
   ZZZZ No decimal
   ZZ.99 Decimal

d) Trailing minus-
   999- Blank is printed, if number is positive

e) Signed numbers-
   s99 + or - always printed

Table 2 contains more examples of sample data, format fields, and the resultant editing.

FORMAT FIELD	DATA	EDITED RESULT
AAAAA	ABC	ABC
AAAAA	ABCDEFG	ABCDE
$$$$	99	$99
$9999	99	$ 99
$99.99	77	$77.00
$99.99	−77	$77.00
$99.99−	−77	$77.00−
$99.99−	77	$77.00
S$99.99	77	+ $77.00
ZZZZ	77	0077
ZZ.999	77	77.000
ZZZ.99	77	077.00
999.99	77	77.00
.99	77	.**
.99	.001	.00
S.999	1.5E−02	+.015
Z.999−	1.5E−02	0.015
Z.999−	−1.5E−02	0.015−

Note: Above $ sign will be replaced with ¥ sign in JIS version.

TABLE 2. Formatting Examples

The printer software handles up to ten significant figures, and an exponent range ± 99 on numbers passed as data to be formatted. Exponential numbers must be normalized such that $0 \leq |x| < 10$.

Alpha (String Data):

The letter A represents one position of an alpha field. Within the field, leading blanks are truncated, the field is left-justified, and it is padded to the right with blanks. Shifted blanks CHR$(160), are not deleted if in the leading positions.

Example:

```
10 OPEN2,4,2
20 OPEN1,4,1
30 PRINT#2,"A  AA  AAA"
40 PRINT#1,"CBM"CHR$(29)"CBM"CHR$(29)"CBM"
50 CLOSE2:CLOSE1
```

Results in:

```
C  CB  CBM
```

Skip (Blank):

Simply use blanks where spaces are required:

Example:

```
10 OPEN2,4,2
20 OPEN1,4,1
30 PRINT#2,"AAA AAA    AAA"
40 PRINT#1,"PET"CHR$(29)"PET"CHR$(29)"PET"
50 CLOSE2:CLOSE1
```

Results in:

```
PET PET    PET
```

To right justify integers, define an alpha string the length of the longest integer. Then blank fill the left of the integer data string and right justify the number within that string.

Literals in Format Strings

Literals are characters that are to be printed exactly as they exit in the printer's memory rather than being used to format other data. Literals are flagged by preceding each literal in the format string with a reverse-field ON character.

One use of literals is to create a special form at the same time you are printing data. This is done by using the special graphics characters to form vertical, horizontal, or other lines between the fields. You can create a form while printing data, either by overprinting graphic characters or by inserting literal characters in a format string as shown below.

The only limitation on literal characters in format strings is that they cannot be printed in the reverse field mode. They are also restricted to the same character set in which data are currently being printed.

Example:

```
10 OPEN2,4,2
20 OPEN1,4,1
30 PRINT#2,"■[AAAAA■]"
40 PRINT#1,"MPP-1361"CHR$(29)
50 CLOSE2:CLOSE1
```

Results in:

```
[MPP-1]
```

Setting the Number of Lines Per Page: sa=3

This special secondary address allows you to vary the number of printed lines per page. In order for this paging option to take effect, you must turn paging on with the special paging character described on page 35. When paging is on and the paging secondary address is not implemented, the default number of lines per page is 66 including three blank lines at the top of the page.

Example:

```
10 OPEN4,4
20 OPEN1,4,1
30 OPEN2,4,2
40 OPEN3,4,3
50 A$="999    9999      99.99999999    99.99999999"
55 PRINT#2,A$
60 PRINT#3,CHR$(60)
70 PRINT#4,CHR$(147)
80 FORI=1TO99
90 PRINT#1,I;I*I;SQR(I);I↑(1/3)
100 NEXTI
110 PRINT#4,CHR$(19)
120 CLOSE4:CLOSE3:CLOSE2:CLOSE1
```

Enabling the Printer Diagnostic Messages: sa=*4*

When a secondary address of 4 is transmitted, the printer's diagnostic messages appear whenever an error occurs. When a formatting error occurs, a message is printed, the format is dumped, and a pointer is located at the offending field. If a secondary address of 4 has NOT been transmitted, then once an error occurs, data characters are dumped directly to the print line even if the last command was to format data.

Example:

```
*PE:C*


 10 OPEN4,4,4:PRINT#4:CLOSE4
 20 OPEN25,4,25
 30 PRINT#25
 40 CLOSE25
 50 OPEN1,4:CMD1:LIST
READY.
```

Whether or not diagnostic messages are enabled, overflow of numeric fields is always indicated by fields filled with asterisks(******). This is a nonfatal error; as in other cases, the secondary address option is reset to zero and all data retrieved by the printer are printed exactly as they are received.

Diagnostic Messages

PE:L - Lines per page out of range. An attempt was made to set the lines per page to a value outside the range <13<linesperpage<*128* via a print-to-SA 3. The command is ignored and the previous length remains in force.

PE:C - Bad command. You gave the printer an invalid SA outside the range. The command is ignored.

PE:M - Data-format mismatch. You gave the printer non-numeric data to print in a numeric field. The first character printed after the error message is the offending character.

*PE:E - Exponent error. The numeric data given to the printer to print in a numeric field had an invalid exponent. The expected form of a number is exponential form is : n.nnnn+ee or n.nnnn-ee. The exponent must contain a plus or minus sign with a two digit exponent.

PE:F - Bad format. The data, that was sent to SA 2 store, either contains illegal formatting characters, or was of an unrecognizable syntax.

PE:T - Terminator error. The SA changed before a terminator was detected. A terminator is defined as a carriage return (chr$(13)), a line feed (chr$(10)), or a carriage return line-feed sequence. Communication with the current SA must be terminated with a terminator before attempting to "talk" to another SA.

Defining a Programmable Character: sa=5

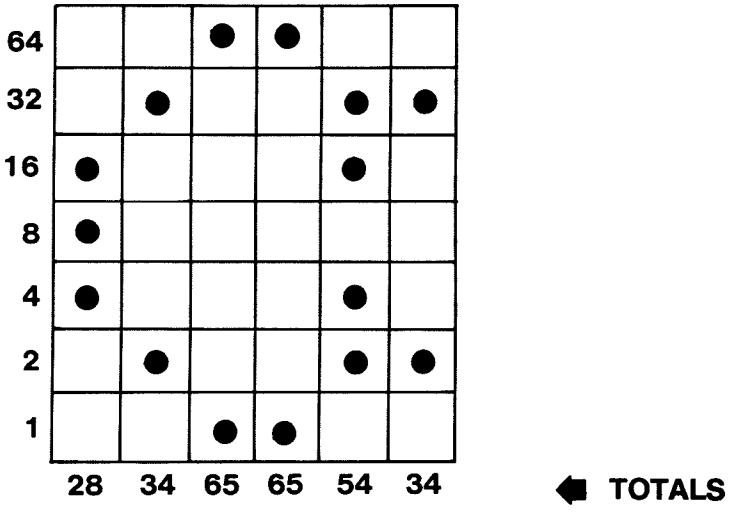
A secondary address of 5 allows you to create a custom character of your own. This programmable character is initialized with this secondary address.

Suppose you wanted to program the symbol in Commodore Business Machines' logo, which is:

C= commodore

Lay out a 7x6 matrix (the same matrix as characters). To the left of the matrix, write the binary bit value of each line. Use dots, one per square to create you character. Then add up the binary bits indicated by your dots to each column. These totals are used in the DATA statement in your program.

Example:



The DATA statement in your program will read:

DATA 28,34,65,65,54,34

The program shown in the printout in the next example writes the Commodore logo ten times. It creates a string with the CHR$ value of the column totals and passes the string to the printer with sa=5. To achieve upper- and lower-case characters, use the CRSR Up (Cursor Up) for upper-case characters, and CRSR Down (Cursor Down) for lower-case characters.

Example:

```
10 DATA28,34,65,65,54,34
20 OPEN5,4,5
30 FORI=1TO6:READA:A$=A$+CHR$(A):NEXT
40 PRINT#5,A$
50 OPEN4,4
60 FORI=1TO10
70 PRINT#4,CHR$(1)CHR$(254)" ■C■OMMODORE ■B■USINESS ■M■ACHINES"
80 NEXT
90 CLOSE5
100 CLOSE4
```

The following is a description of the program shown above:

Line No.

20 Open 5,4,5

- 5 — Communicates to the printer what function you want executed. A secondary address of 5 indicates that you want something stored in the printer's memory.
- 4 — The device number (the printer's)
- 5 — The logical file number

30 The FOR NEXT loop contained in this line reads the data contained in line 10 and constructs a string A$. A$ contains the necessary information to print the - Commodore logo.

40 Stores A$ in printer memory.

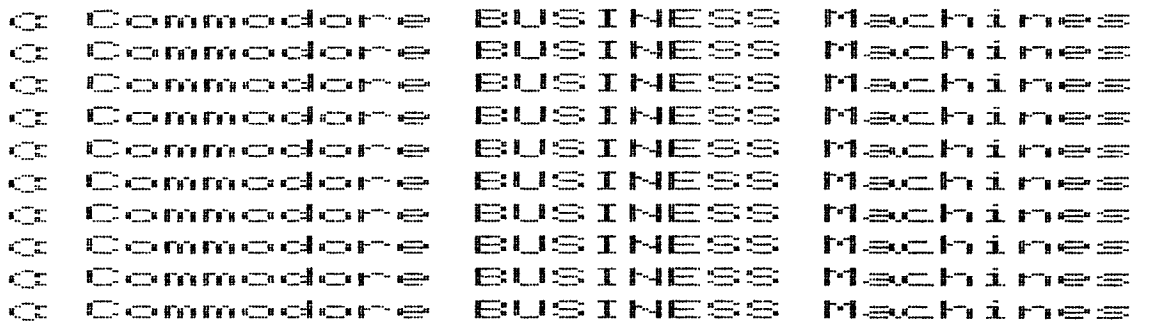
50 Opens the print file for printing.

60-80 Prints the Commodore logo and "Commodore Business Machines" ten times.

PRINT#4	Prints to the printer
CHR#(1)	Prints 'enhanced characters'
CHR$(254)	Prints the programmed character (the Commdore logo)
Cursor Up	Sets the printer to uppercase
Cursor Down	Sets the printer to lowercase

90-100 Closes the two files opened by the program.

After typing RUN, you get this result:



NOTE

Multiple programmable characters in the same line can only be made by overprinting. The programmable character cannot be changed when a line wraps around to the next line.

Setting Spacing Between Lines: sa=6

A secondary address assignment of 6 controls the number of steps between successive lines of print. There are 72 steps per inch, so a declared value (≦127) of 9 produces eight lines per inch. Declaired value of 72 produces lines spaced one inch apart. The default value is 12, which produced the standard 6 lines per inch.

Example:

```
10 OPEN4,4
20 OPEN6,4,6
30 FORI=0TO127STEP4
40 PRINT#6,CHR$(I)
50 PRINT#4,"HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH"
60 NEXT
```

Selecting Upper or Lower Case: sa=7

To change the computer and the printer to all upper-case characters, you must change location 59468 to 12 and condition the printer as indicated by the following program:

Example:

```
10 POKE59468,12
20 OPEN7,4,7:PRINT#7:CLOSE7
30 OPEN4,4
40 PRINT#4,"MPP-1361 PRINTER"
50 CLOSE4
```

Results in:

```
mpp-1361 printer
```

Selecting ASCII/Graphics: sa=8

To change the computer and printer back to ASCII/graphics, you must restore location 59468 to 14 and restore the printer back to its original condition as indicated by the following program:

Example:

```
10 poke59468,14
20 open8,4,8:print#8:close8
30 open4,4
40 print#4,"mpp-1361 printer"
50 close4
```

Results in:

```
MPP-1361 PRINTER
```

Suppress Diagnostic Message Printing: sa=9

To turn off the diagonostic message printing, you will send a secondary adress of 9.

Example:

```
*PE:M*
COMMODORE
```

```
COMMODORE
```

```
10 OPEN4,4,4
20 PRINT#4
30 CLOSE4
40 OPEN2,4,2
50 OPEN1,4,1
60 PRINT#2,"###.9"
70 PRINT#1,"COMMODORE"
80 CLOSE2:CLOSE1
90 OPEN1,4:CMD1:LIST
```

```
10 OPEN4,4,4:PRINT#4:CLOSE4
15 OPEN9,4,9:PRINT#9:CLOSE9
20 OPEN2,4,2
30 OPEN1,4,1
40 PRINT#2,"###.9"
50 PRINT#1,"COMMODORE"
60 CLOSE2:CLOSE1
70 OPEN1,4:CMD1:LIST
```

Resetting the Printer: sa=10

To reset the printer, you will send a secondary address of 10.

Example:

```
10 OPEN10,4,10
20 PRINT#10
30 CLOSE10
```

Setting Uni-Direction Mode: sa=11

To set the uni-directional print mode on normal and condense mode, you will send a secondary address of 11.

Example:

```
10 OPEN4,4
20 OPEN11,4,11:PRINT#11:CLOSE11
30 PRINT#4,"SETTING UNI DIRECTION MODE"
40 PRINT#4,"SETTING UNI DIRECTION MODE"
50 CLOSE4
```

Results in:

```
SETTING UNI DIRECTION MODE
SETTING UNI DIRECTION MODE
```

Resetting Uni-Direction Mode: sa = 12

To reset the uni-directional print mode on normal and condense mode, you will send a secondary address of 12.

Example:

```
10 OPEN4,4
20 OPEN11,4,11:PRINT#11:CLOSE11
30 PRINT#4,"SETTING UNI DIRECTION MODE"
40 PRINT#4,"SETTING UNI DIRECTION MODE"
50 OPEN12,4,12:PRINT#12:CLOSE12
60 PRINT#4,"RESETTING UNI DIRECTION MODE"
70 PRINT#4,"RESETTING UNI DIRECTION MODE"
80 CLOSE4
```

Results in:

```
SETTING UNI DIRECTION MODE
SETTING UNI DIRECTION MODE
RESETTING UNI DIRECTION MODE
RESETTING UNI DIRECTION MODE
```

Setting Condense Mode: sa = 13

To set the condense mode, 250 characters per line, you will send a secondary address of 13:

Example:

```
10 OPEN4,4
20 FORI=32TO95:A$=A$+CHR$(I):NEXT
30 PRINT#4,A$
40 OPEN13,4,13:PRINT#13:CLOSE13
50 PRINT#4,A$
60 CLOSE4
```

Results in:

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
```

Resetting Condense Mode: sa = 14

To reset the condense mode, you will send a secondary address of 14:

Example:

```
10 OPEN4,4
20 FORI=32TO95:A$=A$+CHR$(I):NEXT
30 OPEN13,4,13:PRINT#13:CLOSE13
40 PRINT#4,A$
50 OPEN14,4,14:PRINT#14:CLOSE14
60 PRINT#4,A$
70 CLOSE4
```

Results in:

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
```

Setting Pseudo Letter Quality Mode: sa = 15

A secondary address of 15 allows you to print in the pseudo letter quality mode. To print in the pseudo letter quality mode, the condense mode (Sa = 13) must be set prior to the pseudo letter quality mode.

Example:

```
10 OPEN4,4
20 PRINT#4,"SETTING QUASI-LETTER QUALITY MODE"
30 OPEN13,4,13:PRINT#13:CLOSE13
40 OPEN15,4,15:PRINT#15:CLOSE15
50 PRINT#4,"SETTING QUASI-LETTER QUALITY MODE"
60 CLOSE4
```

Results in:

```
SETTING QUASI-LETTER QUALITY MODE
SETTING QUASI-LETTER QUALITY MODE
```

Resetting Pseudo Letter Quality Mode: sa = 21

To reset the pseudo letter quality mode, you will send: sa = 21 at first, and then sa = 14.

Example:

```
10 OPEN4,4
30 OPEN13,4,13:PRINT#13:CLOSE13
40 OPEN15,4,15:PRINT#15:CLOSE15
50 PRINT#4,"SETTING QUASI-LETTER QUALITY MODE"
55 PRINT#4,"SETTING QUASI-LETTER QUALITY MODE"
60 OPEN21,4,21:PRINT#21:CLOSE21
70 OPEN14,4,14:PRINT#14:CLOSE14
80 PRINT#4,"RESETTING QUASI-LETTER QUALITY MODE"
85 PRINT#4,"RESETTING QUASI-LETTER QUALITY MODE"
90 CLOSE4
```

Results in:

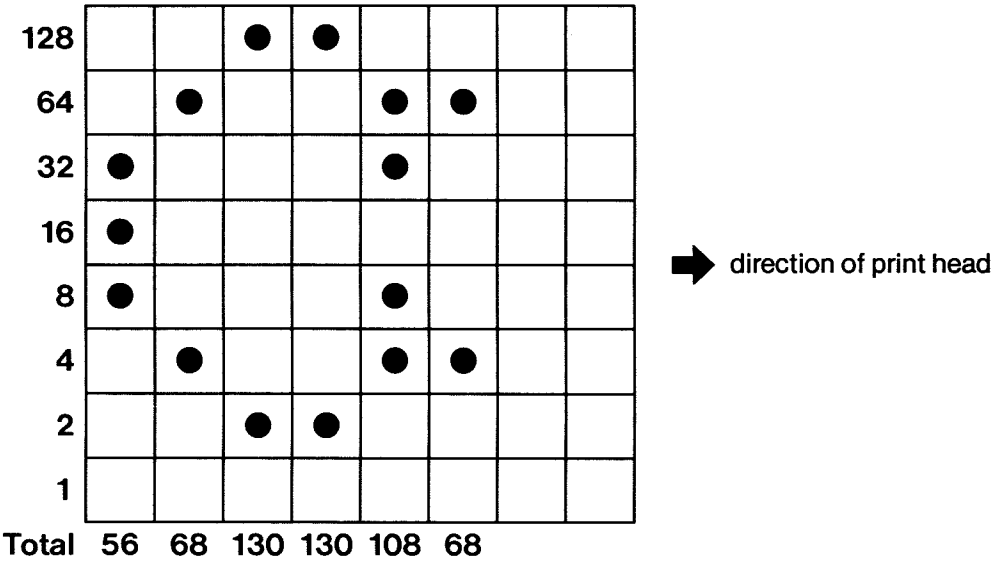
```
SETTING QUASI-LETTER QUALITY MODE
SETTING QUASI-LETTER QUALITY MODE

RESETTING QUASI-LETTER QUALITY MODE
RESETTING QUASI-LETTER QUALITY MODE
```

Storing Bit Image Data: sa = 17

A secondary address of 17 allows you to store bit image data into buffer. In normal mode, 8*816 binary bit data can be written and 8*1020 in condence mode. This data can be printed by sending a secondary address of 18 (sa = 18). But the printing starts automatically when the buffer overflows, and then the line feed will be made.
Suppose you wanted to program the Commodore logo similar to the one in the example of sa = 5.

Lay out 8*816 matrix (or 8*1020 matrix). To the left of the matrix, write the binary bit value of each row. Use dots, one per square to create your image pattern. Then add up the binary bits indicated by your dots in each column. These totals are used in the DATA statement in your program.



The DATA statement in your program will read:

DATA 56, 68, 130, 130, 108, 68

The following is the example program to store the data in the buffer.

```
10 DATA 56,68,130,130,108,68
20 OPEN17,4,17
30 FORI=1TO6
40 READA:A$=A$+CHR$(A)
50 NEXT
60 PRINT#17,A$;
70 CLOSE17
```

Printing Bit Data Previously Written: sa = 18

The bit image data stored by Sa = 17 can be printd by sending Sa = 18.

A program shown below prints the image pattern created as an example in Sa = 17.

Example:

```
10 DATA 56,68,130,130,108,68
20 OPEN17,4,17
30 FORI=1TO6
40 READA:A$=A$+CHR$(A)
50 NEXT
60 PRINT#17,A$;
70 CLOSE17
80 OPEN18,4,18
90 PRINT#18
100 CLOSE18
```

Results in:



SPECIAL CHARACTER FUNCTIONS

Special control characters can be used to change the mode of printing within a single line. Table 3 contains a summary of the special control characters. The table is followed by a brief description of each function.

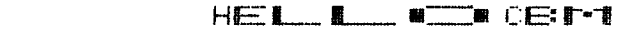
Enhanced Characters

You can print any character double its width. While characters are normally printed in a seven-row by six-column matrix, enhanced characters appear in a seven-row by twelve-column matrix. The example below shows how already enhanced characters can be enhanced, and how to unenhance:

Example:

```
10 OPEN4,4
20 ENHANCE$=CHR$(1):UNENHANCE$=CHR$(129)
30 PRINT#4,"H"EN$"E"EN$"L"EN$"L"EN$"O"UN$"C"EN$"B"EN$"M"
40 CLOSE4
```

Results in:



Paging

Paging must be turned on in your program by means of the keyboard-entry code as shown in Table 3. Otherwise, printing is continuous. The paging function provides for 66 lines per page - including three blank lines at the top of the page and three blank lines at the bottom. The number of lines-per-page may be altered by using the secondary address 3 option. When paging is in effect, the paging off character performs a top-of-form function.

Specifying Individual Upper- and Lower-case Characters

The printer default character set is that of the CBM Computer when power is first applied to the printer. Though it is not possible to display both graphics and lowercase simultaneously on the video display screen, the printer does allow shifting between character sets on the same line. This function is analogous to the letter/figures shift on Teletype™.

Example:

```
10 OPEN4,4
20 PRINT#4,"■C■OMMODORE"
30 CMD4
40 LIST
```

Results in:

```
Commodore

10 OPEN4,4
20 PRINT#4,"■C■OMMODORE"
30 CMD4
40 LIST
```

TABLE 3

Special Control Character Summary

Printer function	Code	ASCII	Keyboard
Enhance	CHR$(1)	SOH	NA
Unenhanced	CHR$(129)		NA
Paging on/reset	CHR$(147)		SHIFT & CLR HOME
Paging off	CHR$(19)	DC3	CLR HOME
RVS ON	CHR$(18)	DC2	OFF RVS
RVS OFF	CHR$(146)		SHIFT & OFF RVS
Carriage return with no line feed	CHR$(141)		NA
Line feed	CHR$(10)	LF	NA
Uppercase	CHR$(145)		CRSR Cursor Up
Lowercase	CHR$(17)		CRSR Cursor Down
Skip space	CHR$(29)		CRSR Cursor Right
Quote	CHR$(34)	"	" Quote

Reversing a Field

This function inverts the dot matrix to produce the effect of white on black. DO NOT use this mode of printing for more than five consecutive lines since extended printing in this mode will damage the print head.

Example:

```
10 OPEN5,4
20 PRINT#5,"■C■OMMODORE"
30 CMD5
40 LIST
```

Results in:

■commodore

```
10 OPEN5,4
20 PRINT#5,"■C■OMMODORE"
30 CMD5
40 LIST
```

The Carriage Return

If you attempt to print more than 136/250 (normal/condense) characters on a line, a Carriage Return with Line Feed will be forced and the overflow characters will be printed on the next line. If a line contains more than 136/250 (normal/condense) printable characters with not carriage return, the excess over 136/250 (normal/condense) characters will be printed on the next line. If you want to overprinting, you must set CHR$ (141), carriage return with no line feed. See following example.

```
10 OPEN4,4
20 FORI=1TO10
30 PRINT#4,"MPP-1361 PRINTER"CHR$(141);
40 NEXT
```

Results in:

```
MPP-1361 PRINTER
```

Quotation Marks

If an odd number of quotation marks have been transmitted, control characters are made visible. This can be particularly useful when you are making a listing of a BASIC program containing cursor control characters in quotation marks. If you want to print quotation marks, you must send CHR$ (34). See following example.

```
10 OPEN4,4
20 CMD4
30 PRINT#4,"10 OPEN4,4"
40 PRINT#4,"20 PRINT#4,"+CHR$(34)+"MPP-1361 PRINTER"+CHR$(34)
50 PRINT#4:CLOSE4
```

Results in:

```
10 OPEN4,4
20 PRINT#4,"MPP-1361 PRINTER"
```

APPENDIX A

The programs in this appendix illustrate the many formatting capabilities of your printer. The first program is completely annotated. In the following programs, new concepts are described.

SIMPLE STRING FORMATTING

This program opens four files to perform four different tasks. It prints three lines: the format string, the unformatted data string, and the formatted data string. It then lists the program.

Here is a printout from the program:

```
AA AA AA AA
ABC
AB AB AB AB
```

```
 10 OPEN1,4
 20 OPEN2,4,1
 30 OPEN3,4,2
 40 OPEN4,4,4
 50 PRINT#4            :REM ENABLE ERROR DIAGNOSTICS
 60 A$="AA AA AA AA"   :REM FORMAT STRING
 70 B$="ABC"           :REM STRING TO BE FORMATTED
 80 PRINT#1,A$
 90 PRINT#1,B$
 100 PRINT#3,A$
 110 C$=CHR$(29)
 120 PRINT#2,B$C$B$C$B$C$B$
 130 CMD1
 140 PRINT:PRINT:PRINT:PRINT
 150 LIST
```

An analysis of the program will help you understand how it works.

Line NO.	Explanation
10-40	Four logical files are opened to the printer so that data in the computer's memory can be used in four different ways:

Logical File No.	Secondary Address	Usage
1	0 (default)	Print data exactly as transmitted.
2	1	Format data before it is printed.
3	2	Transmit the format string.
4	4	Enable format error diagnostic messages to be printed (if any)

50	Prints error messages (if any)
60	Defines the format string
70	Defines the data string
80	Prints the format string
90	Prints the unformatted data string
100	Transmits the format string to the printer's memory
110	Skip a space
120	Prints the formatted data string
130	Transfers control from the computer to the printer
140	Prints four blank lines
150	Lists the program

LEADING ZEROES FORCED

This program prints ten rows of numbers, each row beginning with a number one higher than in the preceding row. In line 60, the formatting statement (which is prepared for in line 30, and transmitted in line 70) specifies leading zeroes in all fields and plus signs in the first two fields.

```
+0001.   +0002.   0003.   0004
+0002.   +0003.   0004.   0005
+0003.   +0004.   0005.   0006
+0004.   +0005.   0006.   0007
+0005.   +0006.   0007.   0008
+0006.   +0007.   0008.   0009
+0007.   +0008.   0009.   0010
+0008.   +0009.   0010.   0011
+0009.   +0010.   0011.   0012
+0010.   +0011.   0012.   0013
```

```
10 OPEN1,4
20 OPEN2,4,1
30 OPEN3,4,2
40 OPEN4,4,4
50 PRINT#4,     :REM ENABLE ERROR DIAGNOSTICS
60 F$="SZZZZ.   SZZZZ.   ZZZZ.   ZZZZ"
70 PRINT#3,F$
80 FOR I=1 TO 10
90 PRINT#2,I,I+1,I+2,I+3
100 NEXT
110 CMD1
120 PRINT:PRINT:PRINT:PRINT
130 LIST
```

SIGNED NUMBERS

As shown below, a trailing sign, when specified in a format statement, either prints a blank for a positive number or a minus (-) for a negative number. This form is sometimes desired by accountants.

A leading sign always prints a plus (+) or a minus (-) for the sign of the number.

If no sign is specified in the format, then no sign is printed. This works essentially like an absolute value on the number.

```
 1  -  1     1-
```

```
10 OPEN1,4
20 OPEN2,4,1
30 OPEN3,4,2
40 OPEN4,4,4
50 PRINT#4     :     REM ENABLE ERROR DIAGNOSTICS
60 F$="999  $999  999-"
70 PRINT#3,F$
80 PRINT#2,-1,-1,-1
90 CMD1
100 PRINT:PRINT:PRINT:PRINT
110 LIST
```

DECIMAL FRACTION SPECIFICATION

When the right hand side of the decimal point in a format statement has more digits of precision specified than contained in the formatted data, the remaining digit positions are filled with zeroes. If the converse is true, then the formatted number is truncated.

```
 5.00  -  1.25000    3
10.00  -  2.50000    6
15.00  -  3.75000    9
20.00  -  5.00000   12
25.00  -  6.25000   15
30.00  -  7.50000   18
35.00  -  8.75000   21
40.00  - 10.00000   24
45.00  - 11.25000   27
50.00  - 12.50000   30
```

```
10 OPEN1,4
20 OPEN2,4,1
30 OPEN3,4,2
40 OPEN4,4,4
50 PRINT#4     :     REM ENABLE ERROR DIAGNOSTICS
60 F$="999.99  $999.99999  999-"
70 PRINT#3,F$
80 FORI=1TO10
90 PRINT#2,-10*I/2,-5*I/4,3*I
100 NEXT
110 CMD1
120 PRINT:PRINT:PRINT:PRINT
130 LIST
```

FIXED AND FLOATING $ FIELDS

In fixed dollar fields the dollar sign ($) is printed in the leftmost position and leading unused digit positions are printed as blanks.

In floating dollar fields the $ is printed immediately to the left of the most significant figure or decimal point. Unused digit fields to the left of the $ are printed blank.

```
$28.32    $1.67   $  279   $.23
$70.15    $4.46   $ 9739   $.48
$68.84    $1.66   $  710   $.14
$83.96    $1.85   $ 1817   $.04
$78.13    $9.38   $    8   $.10
$28.17    $2.60   $ 9961   $.80
$31.51    $7.90   $ 6886   $.94
$24.17    $8.35   $ 8401   $.07
$39.04    $9.95   $ 8103   $.77
$72.46    $9.46   $ 5951   $.97
```

```
10 OPEN1,4
20 OPEN2,4,1
30 OPEN3,4,2
40 OPEN4,4,4
50 PRINT#4          :REM ENABLE ERROR DIAGNOSTICS
60 F$="$99.99  $$$.99  $99999  $.99"
70 PRINT#3,F$
80 FOR I=1 TO 10
90 PRINT#2,RND(1)*100,RND(1)*10,RND(1)*10000,RND(1)
100 NEXT
110 CMD1
120 PRINT:PRINT:PRINT:PRINT
130 LIST
```

MIXED FORMAT FIELD TYPES

Shown below are integer with leading zeroes, alpha field, floating dollar and integer with zero suppressed.

In line 80, CHR$(29), (skip) is used as a terminator for the alpha field so that it is formatted correctly.

```
00002  PET COMPUTER                    $795.00    50
```

```
10 OPEN1,4
20 OPEN2,4,1
30 OPEN3,4,2
40 OPEN4,4,4
50 PRINT#4         :REM ENABLE ERROR DIAGNOSTICS
60 F$="ZZZZZ  AAAAAAAAAAAAAAAAAAAAAAAAA  $$$$$.99  999"
70 PRINT#3,F$
80 A=2
90 B$="PET COMPUTER"
100 C=795
110 D=50
120 PRINT#2,A,B$,CHR$(29),C,D
130 CMD1
140 PRINT:PRINT:PRINT:PRINT
150 LIST
```

FORMATTING WITH LITERALS

In the example below, a literal appears at the end of the printout even though the format statement does not contain one in that position. Your printer automatically prints any literal that appears at the beginning of the format string at the beginning and end of the printout.

```
| 08 | $451  | 03.431  |
| 08 | $391  | 01.243  |
| 07 | $387  | 03.314  |
| 06 | $947  | 04.723  |
| 08 | $511  | 06.051  |
| 02 | $654  | 03.581  |
| 04 | $378  | 03.464  |
| 01 | $514  | 00.228  |
| 08 | $184  | 04.107  |
| 06 | $741  | 05.432  |
```

```
10 OPEN1,4
20 OPEN2,4,2
30 OPEN3,4,1
40 F$="■■ ZZ ■■ $$$$ ■■ ZZ.999 ■■ "
50 PRINT#2,F$
60 FOR I=1 TO 10:X=10*RND(1):Y=1000*RND(1):Z=8*RND(1)
70 PRINT#3,X;Y;Z:NEXT
80 CMD1:LIST
```

```
READY.
```

APPENDIX B

ASCII Character Codes

ASCII CODE	CHARACTER	ASCII CODE	CHARACTER	ASCII CODE	CHARACTER
000	NULL	043	+	086	V
001	SOH	044	,	087	W
002	STX	045	–	088	X
003	ETX	046	.	089	Y
004	EOT	047	/	090	Z
005	ENQ	048	0	091	[
006	ACK	049	1	092	bkslash
007	BEL	050	2	093	]
00	BS	051	3	094	↑
009	HT	052	4	095	back arr
010	LF	053	5	096	space
011	VT	054	6	097	a
012	FF	055	7	098	b
013	CR	056	8	099	c
014	SO	057	9	100	d
015	SI	058	:	101	e
016	DLE	059	;	102	f
017	DC1	060	<	103	g
018	DC2	061	=	104	h
019	DC3	062	>	105	i
020	DC4	063	?	106	j
021	NAK	064	@	107	k
022	SYN	065	A	108	l
023	ETB	066	B	109	m
024	CAN	067	C	110	n
025	EM	068	D	111	o
026	SUB	069	E	112	p
027	ESCAPE	070	F	113	q
028	FS	071	G	114	r
029	GS	072	H	115	s
030	RS	073	I	116	t
031	US	074	J	117	u
032	SPACE	075	K	118	v
033	!	076	L	119	w
034	"	077	M	120	x
035	#	078	N	121	y
036	$	079	O	122	z
037	%	080	P	123	;
038	&	081	Q	124	<
039	'	082	R	125	=
040	(	083	S	126	>
041	)	084	T	127	DEL
042	*	085	U		

ASCII codes are in decimal

APPENDIX C

SAMPLE PROGRAM OF BIT IMAGE PRINT

```
1000 REM **********************************
1010 REM * MODEL MPP-1361 BIT IMAGE PRINT *
1020 REM **********************************
1030 OPEN 4,4
1040 OPEN 6,4,6:PRINT#6,CHR$(8):REM SET 1/8 INCH LINE FEED
1050 OPEN 13,4,13:PRINT#13:REM CONDENSE PRINT MODE SET
1060 OPEN 17,4,17:REM OPEN STORING BIT IMAGE DATA INTO BUFFER
1070 OPEN 18,4,18:REM OPEN PRINT START BIT IMAGE DATA
1080 READ A:REM READ NUMBER OF LINES
1090 FOR I = 1 TO A
1100 READ X:REM READ NUMBER OF BIT IMAGE DATA PER LINE
1110 FOR J = 1 TO X
1120 READ C:PRINT#17,CHR$(C)CHR$(C);:REM BIT IMAGE DATA WRITE TO BUFFER
1130 NEXT J
1140 PRINT#18:REM PRINTING
1150 NEXT I
1160 CLOSE18:CLOSE17
1170 PRINT#6,CHR$(12);:REM SET 1/6 INCH LINE FEED
1180 CLOSE6:OPEN15,4,15:PRINT#15:CMD4:PRINT:LIST1000-
1190 REM ** SAMPLE OF DATA **
1200 DATA10
1210 DATA47,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,7,7,15,15,31,31,63,63,127,127,127
1220 DATA127,127,255,255,255,255,255,255,255,255,255,255,255,255,255,255,127
1230 DATA127,127
1240 DATA47,0,0,0,0,0,1,3,7,15,31,63,127,255,255,255,255,255,255,255,255,255
1250 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
1260 DATA255,255,255,255,255,255,255,255
1270 DATA81,0,0,3,15,31,127,255,255,255,255,255,255,255,255,255,255,255,255
1280 DATA255,255,255,255,255,254,252,248,240,240,224,224,224,192,192,192,192
1290 DATA192,192,192,192,192,192,224,224,224,240,240,255,15,15,15,15,15,15,15
```

```
 1300 DATA15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,14
 1310 DATA12,8
 1320 DATA77,3,63,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 1330 DATA255,255,252,240,192,128,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 1340 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 1350 DATA255,255,255,255,255,255,255,254,252,248,240,224,192,128
 1360 DATA69,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 1370 DATA255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,243,243
 1380 DATA243,243,243,243,243,243,243,243,243,243,243,243,243,243,243,243,243,243
 1390 DATA243,225,192,128
 1400 DATA75,128,252,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 1410 DATA255,255,31,7,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255
 1420 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 1430 DATA255,255,255,127,63,31,15,7,3,1
 1440 DATA81,0,0,192,240,252,255,255,255,255,255,255,255,255,255,255,255,255,255
 1450 DATA255,255,255,255,255,127,63,31,15,15,7,7,3,3,3,3,1,1,1,1,1,1,1,1,3,3,3
 1460 DATA7,255,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252
 1470 DATA252,252,252,252,252,252,252,252,252,252,252,252,252,252,124,60,28,12,4
 1480 DATA47,0,0,0,0,0,0,128,192,224,240,248,252,254,255,255,255,255,255,255,255
 1490 DATA255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 1500 DATA255,255,255,255,255,255,255,255,255,255,255
 1510 DATA47,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,192,224,240,240,248,248,252,252,254
 1520 DATA254,254,254,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
 1530 DATA255,255,254,254,254
 1540 DATA42,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,128
 1550 DATA128,128,128,128,128,128,128,128,128,128
 1560 REM ** SAMPLE OF DATA END **
READY.
```

commodore
COMPUTER