9831A

Peripheral Control

Hewlett · Packard
Desktop Computer

Peripheral Control Manual
HP 9831A Desktop Computer

Hewlett-Packard Calculator Products Division
P.O. Box 301, Loveland, Colorado 80537, Tel. (303) 667-5000
(For World-wide Sales and Service Offices see back of manual.)

HEWLETT · PACKARD

Manual Summary

Chapter 1: General Information

Introduces the 9831A's peripheral devices and input/output scheme.

Chapter 2: General I/O Operations

Describes each statement, function and command used to control the peripheral devices.

Chapter 3: HP 9863A Tape Reader

Tape Reader control operations are explained.

Chapter 4: HP 9864A Digitizer

Digitizer control operations are explained.

Chapter 5: HP 9866A/B Printer

Printer control operations are explained.

Chapter 6: HP 9869A Hopper Card Reader

Card reader control operations are explained.

Chapter 7: HP 9871A Printer

Printer control operations are explained.

Chapter 8: HP 9877A External Tape Memory

External Tape Memory control operations are explained.

Chapter 9: HP 9881A Line Printer

Unique printer control operations are explained.

Chapter 10: HP 9883A Tape Reader

Tape Reader control operations are explained.

Chapter 11: HP 9884A Tape Punch

Tape punch control operations are explained.

Table of Contents

Chapter 1: General Information

Introduction 1

9831A Peripherals 1

9831A I/O Scheme 5

HP 98032A Interface 5

Select Code 6

Input-Output Format 7

Peripheral Interrupt 7

Binary Coding and Conversions 7

Chapter 2: General I/O Operations

Introduction 11

Syntax Conventions 11

I/O Statements 12

The WRITE Statement 12

The FORMAT Statement 12

The OUTPUT Statement 15

The ENTER Statement 19

I/O Functions 22

The WBYTE Function 22

The RBYTE Function 23

The STAT Function 24

I/O Commands 25

The LIST Command 25

The PTAPE Command 25

Conversion Tables 26

Conversion in the ENTER Statement 27

Conversion in the OUTPUT Statement 27

Example of Code Conversion 27

Binary Functions 30

Binary And (BIAND) Function 30

Inclusive Or (INOR) Function 31

Rotate (ROT) Function 31

I/O Applications 32

External Data Transfer 32

Parity Generation 33

Absolute Tape 34

Chapter 3: HP 9863A Tape Reader

- Introduction 35
  - Description 35
- Data Input 37
  - Using Strings 38
  - Program Input 39
  - Data Mode 40

Chapter 4: HP 9864A Digitizer

- Introduction 43
  - Description 43
  - Digitizer Controls 44
- Entering Data 46
- The Beep 48
  - Document Alignment 48
- Line Length Program 49

Chapter 5: HP 9866A/B Printer

- Introduction 51
  - Description 51
- The Standard Printer Statement 52
- The PRINT Statement 53
- Printer Line Width 55
- Clearing the Buffer 56
- 9866A/B Printer Status 56
- Plotting 56
  - Character Plotting 56
  - 9866B Plot Mode 58
  - Printing Dot Patterns 58

Chapter 6: HP 9869A Hopper Card Reader

- Introduction 65
  - Description 65
- The Data Buffer 66
- Operating Modes 66
- Card Reader Functions 67
- The WRITE Statement 67
- The ENTER Statement 67
  - Free-Field Format 68
- FORMAT Statements 68
- Pick Fails 69

Continuous Mode Operation 69

The Stop Command 71

Demand Mode Operation 71

The Abort Command 72

The Reject/Select Function 73

The Retransmit Function 73

The Stop Command 74

Additional Operations 74

The Bell Function 74

The RBYTE Function 75

The Image Mode 79

Card Reader Status Line 81

The PTAPE Command 81

Chapter 7: The HP 9871A Printer

Introduction 83

Description 83

Data Output 85

The WRITE Statement 85

String Variables 86

Clearing the Printer 87

The STAT Function 87

9871A Printer Function Codes 89

Sending Function Codes 91

General 91

Horizontal Tabulation 94

Vertical Tabulation 95

Form and Margin Control 96

Spacing Control 98

Plotting Control 99

The 9871A Option 001 101

Chapter 8: The HP 9877A External Tape Memory

Introduction 103

Description 103

The TAPE Statement 104

Select Code Combination 104

Restrictions 104

Standard 104

9877A Option 002 104

9877A Option 004 105

Tape Duplication 106

Chapter 9: The HP 9881A Line Printer

Introduction 107

Description 107

Standard Printer 108

Printer Operations 108

Printer Status 109

Vertical Format Control 109

Vertical Format Unit Control 109

Line Count Control 110

Format Tape 111

Master Clear 112

Chapter 10: The HP 9883A Tape Reader

Introduction 113

Description 113

Entering Numeric Data 114

Using Strings 115

Control Characters and Special Characters 117

Entering Alphanumeric Data 118

Using Conversion Tables 119

The RBYTE Function 120

Entering Programs From Paper Tape 121

Chapter 11: HP 9884A Tape Punch

Introduction 123

Description 123

Data Output 124

The WRITE Statement 124

The LIST Command 127

The WBYTE Function 127

Parity Generation 128

Appendix:

Basic Syntax 129

ASCII Character Codes 130

Subject Index: 131

Error Messages: 133

CAUTION

THE 9831A DESKTOP COMPUTER CAN BE SEVERLY DAMAGED IF IT HAS NOT BEEN SET TO THE CORRECT LINE VOLTAGE; IF IN DOUBT, PLEASE REFER TO THE OPERATING AND PROGRAMMING MANUAL.

Chapter 1

General Information

Introduction

This manual describes the operation of the 9831A Desktop Computer with various peripheral devices. General peripheral control statements are described in Chapter 2. Operation of the peripheral devices is discussed in the later chapters. This manual together with the 9831A Operating and Programming Manual provides the information necessary to operate the 9831A with each peripheral device included in this manual.

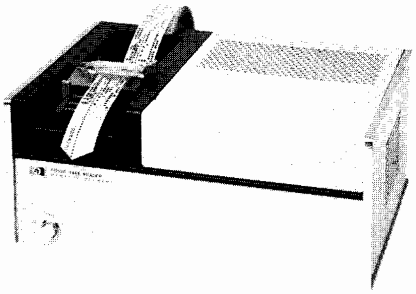
This chapter provides general information about the 9831A and its peripherals.

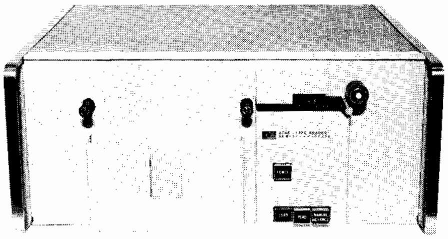
9831A Peripherals

Each of the following peripherals is available with the necessary interface cables. The standard 9831A can control all of these peripherals, with the exception of the Plotters and Flexible Disk Drive.

Paper Tape Readers

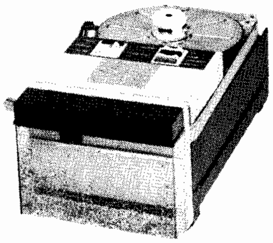
Data from analytical instruments, machine tools and computer terminals can be entered directly into the 9831A from each of these paper tape readers. The HP 9863A makes it easy to read data in a wide variety of formats at 20 characters per second. The HP 9883A Tape Reader, designed for high-speed, heavy-volume operations, optically reads tapes at up to 500 characters per second.

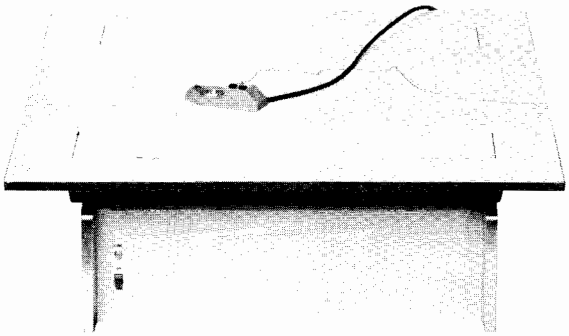




HP 9884A Tape Punch

Add high-speed output to your 9831A with a tape punch. This reliable, compact unit punches tape at 75 characters per second.



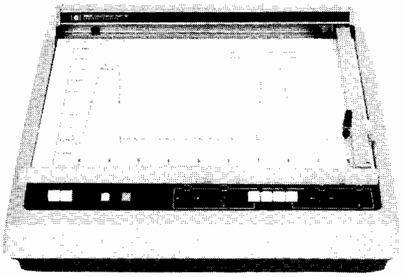


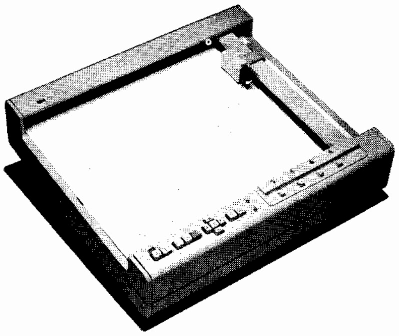
HP 9864A Digitizer

Use this peripheral to read a curve, or any irregular shape, as a series of discrete points and then convert these to a series of digital X-Y coordinates. To make entries, simply trace the shape; then the 9831A can find the dimensions of the line or area of the contained shape. With the proper programs, you can directly process graphical data, such as x-rays, blueprints, strip-chart recordings or cut-and-fill profiles.

X-Y Plotters

Histograms, pie charts, circuit diagrams, linear, log-log and polar plots – these are a few of the things you can do with the 9862A and 9872A Plotters. Additional features of the 9872A include programmable selection of four pens, selectable pen speed, point digitizing, window plotting and many more. Syntax and instructions are contained in the 98223A/B Matrix/Plotter ROMs and the Matrix/Plotter Programming Manual.

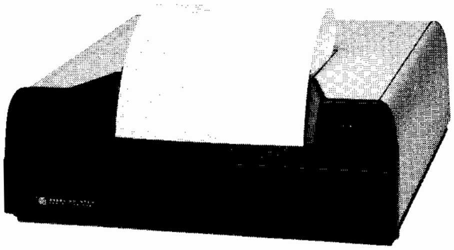




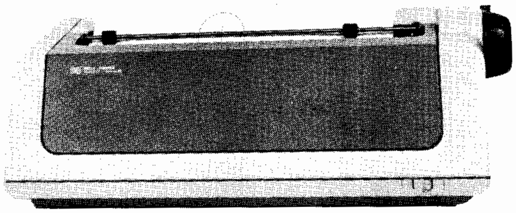
Output Printers

If you need data output in tables, charts or forms, one of these printers can do the job.

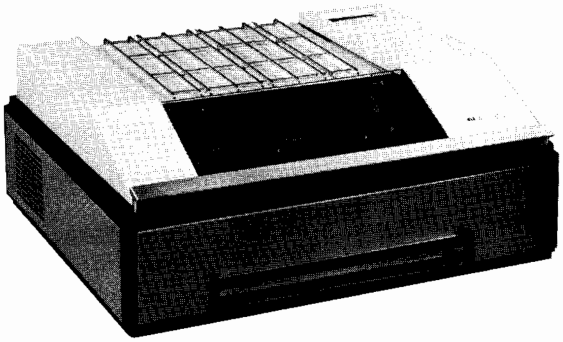
The HP 9866A/B Thermal Printers are fast (240 lines per minute) printers. The 9866B has a 95-character alphanumeric font with upper and lower case, producing fully-formatted text and tables.

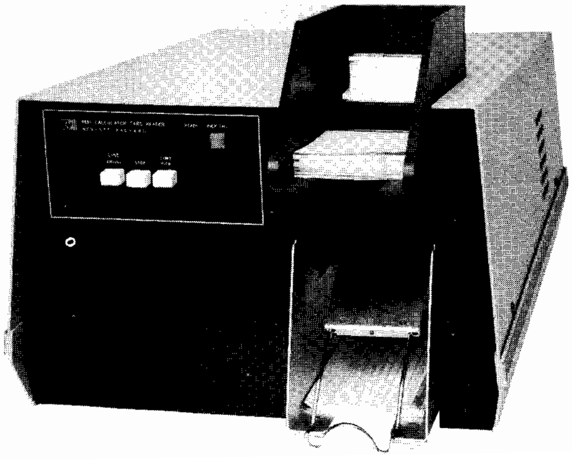


The HP 9871A Character Impact Printer provides 132-character wide, multiple copy output, and a full 96-character font. The average printing speed is 30 characters per second. This printer has additional plotting capabilities.



The HP 9881A Line Printer provides a reliable 5 x 7 dot matrix printer for such applications as payroll, inventory, accounting and report generation. This printer is a 132-column printer rated at 200 lines per minute. It can provide up to six consistently clean copies.

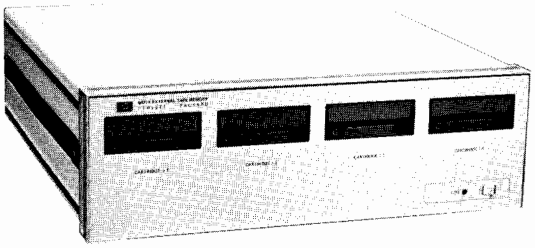


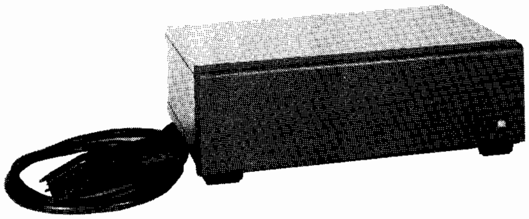


The HP 9869A Card Reader

The card reader provides a convenient form of data entry from punched or marked cards. Standard 80-column cards can be read at up to 300 cards per minute.

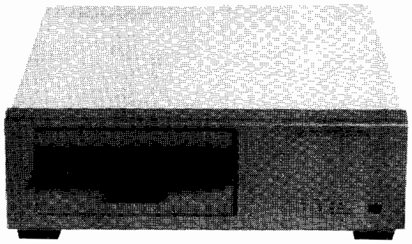
The HP 9877A External Tape Memory
The External Tape Memory provides two valuable additional capabilities. Offering up to one megabyte of tape memory, the unit presents a method of storing large volumes of data. Alternatively, the unit offers a convenient, easy and rapid way of obtaining duplicated tapes.





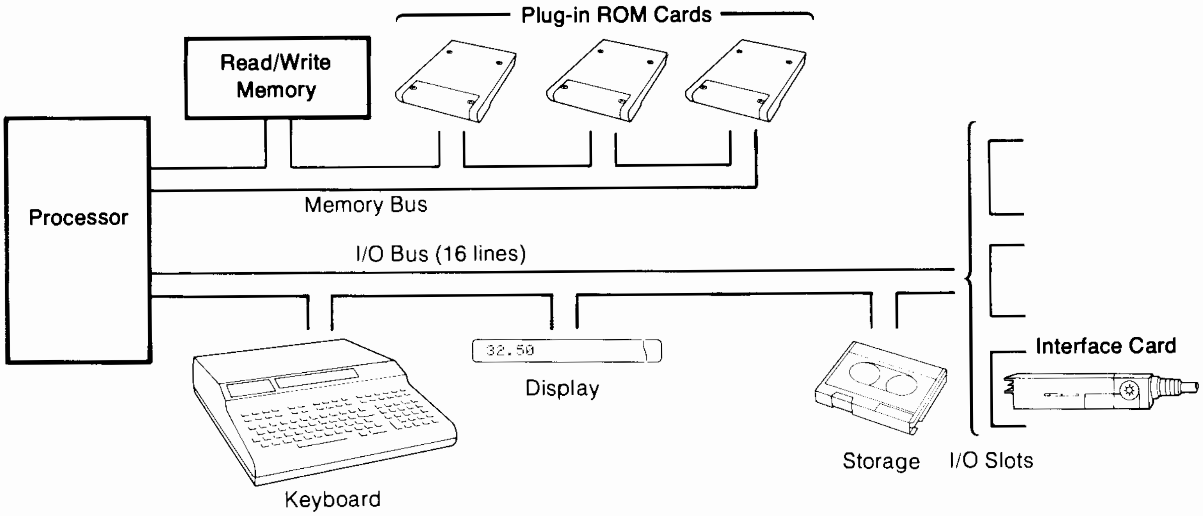
The HP 9878A I/O Expander
The I/O Expander provides six additional I/O slots which will accommodate all the 9831A I/O Cards. The expander has an integral I/O card connecting it to the computer. This card has a system integrity light to indicate that the computer is turned on, the expander is turned on, and the I/O card is properly connected. Another LED is illuminated when the 9878A power is on.

The HP 9885 Flexible Disk Drive
The Flexible Disk Drive provides the 9831A user a convenient, reliable method of transferring programs and data to and from the computer. The 9885 is a random access, removable, mass storage device with a capacity of up to 468,480 bytes per disk. Syntax for control of the 9885 is contained in the 98218A Disk ROM and program control information is found in the 9885 Flexible Disk Drive Operating and Programming Manual.



9831A I/O Scheme

The general 9831A I/O scheme is shown in the next figure. Notice that the I/O bus transfers data between the 9831A processor and peripheral devices. All incoming data is transferred through the processor before it is stored in memory.



The 9831A Input/Output Scheme

Each external device must be connected to the 9831A via an appropriate interface. The interface plugs into any of the I/O slots in the 9831A's back panel. External devices share the same I/O bus with the internal peripheral devices (keyboard, display, etc.). This allows the keyboard to respond to ENTER and RBYTE (read byte) operations. These operations are discussed later in Chapter 2.

HP 98032A Interface

Most of the peripherals use a modified HP 98032A Interface. This is a general purpose, 16-bit parallel, character-serial interface. The interface transfers data between the 9831A and a peripheral device in full-duplex; that is, the interface can have valid data on the output lines and be inputting data at the same time.

NOTE

When ordering your 98032A Interface, the last two digits of the option number (e.g., 98032A Opt. 171) correspond to the last two digits of the peripheral device (e.g. 9871). The first digit refers to the 9831A series of peripherals. These peripherals and interface cables are used with other controllers, therefore the option number of your 98032A interface may begin with a "0" or "1" followed by the last two digits of your device (e.g., 98032A Opt. 071 or Opt. 171).

Select Code

As described, each external device is connected to the 9831A via the same I/O bus. Since all external devices are "party-lined" on the same bus, each device is assigned a unique address, or select code, so that the correct device responds to each I/O operation.

For all external peripherals, the select code is an integer number from 2 thru 15 which is specified in each I/O operation and decoded by the corresponding interface. Each interface has a switch permitting the user to set any one of many different codes. The factory settings for these select codes are as follows –

Peripheral	**Select Code Setting**
Tape Readers	3
Printers	6
(9871A Printer)	2
Digitizer	4
Card Reader	3
Tape Punch	2

The 9831A's display, keyboard and tape drive are internal peripherals. Each internal peripheral has a fixed select code which is automatically specified by standard 9831A statements (display, tape commands, etc.). Both the display and the keyboard respond to select code 0, and the tape drive responds to select code 1. See the Note at the end of this chapter.

The 9831A automatically specifies select code 2 for its standard printer when first switched on. This select code can also be changed using the STDPR statement (see Chapter 8 of the 9831A Operating and Programming Manual).

The select code can be specified in the form of a constant, a variable, or an expression. Each of these WRITE statements, for example, is addressed to the device responding to select code 9 –

```
 10 WRITE (9,*) A,B,C
 10 WRITE (S,*) A,B,C        (S = 9)
10 WRITE (3+S,*) A,B,C       (S = 6)
```

For controlling instruments via the HP-IB, an address code is combined with the select code to form a single expression. A general select code syntax is – cc[dd]

cc = one or two-digit select code.

dd = optional HP-IB address code (must be two digits). An example of the HP-IB address code is given in Chapter 7.

Input-Output Format

The I/O bus connecting the processor with internal and external peripherals contains 16 data lines. Data is transmitted in a 16-bit-parallel, character-serial fashion. The WRITE, OUTPUT, PRINT, ENTER, LIST, TLIST and PTAPE operations send or receive data in standard 8-bit (US) ASCII[1] code. The 9831A sends and receives one 8-bit character at a time. The parity (most-significant) bit is not used with these I/O operations.

Peripheral Interrupt

Since the 9831A is intended as the controlling device in any system, there is no provision for peripheral interrupt operation (i.e., when an external device can call for an I/O operation). The 9831A must be in complete control of each device while that device is involved in data transfer. The interrupt scheme mentioned in some interface manuals is not available with the 9831A's standard I/O operations.

Binary Coding and Conversions

This section is a simplified explanation of binary number representations. For in depth information, refer to any text on the subject.

Binary is the base 2 number system, in which only 1's and 0's are used. By giving the 1's and 0's positional value, any decimal number can be represented. Integer-precision numbers are represented by binary in the 9831A. For example –

decimal 41 = binary 101001

Decimal		Binary					
10^1	10^0	2^5	2^4	2^3	2^2	2^1	2^0
10	1	32	16	8	4	2	1
4	1	1	0	1	0	0	1

[1] ASCII is an acronym for American Standard Code for Information Interchange.

Binary-Decimal Conversions

To convert from binary to decimal, the positional values for ones are added up. From the above example, this would be

$$2^5 + 2^3 + 2^0 = 32 + 8 + 1 = 41$$

To convert from decimal to binary, the decimal number is repeatedly divided by 2. The remainders provide the binary equivalent. For example –

	Remainder (read up)
2) 41	1
2) 20	0
2) 10	0
2) 5	1
2) 2	0
2) 1	1

ASCII Code

Binary is often used as a code to represent not only numbers, but also alphanumeric characters such as: A, ?, x, 2 or a comma. One of the most common binary codes used is ASCII. ASCII is an eight-bit code – seven data bits and one parity bit. The 9831A Computer uses ASCII for alphanumeric character representation. The parity bit is not used by the 9831A, however, it is available.

Examples –

Character	ASCII Binary Code	ASCII Decimal Code
A	01000001	65
B	01000010	66
?	00111111	63

For a complete list of ASCII characters and their equivalent binary and decimal representations, see the ASCII Character Code Table in the Appendix.

NOTE

Using select code zero with the ENTER statement, two types of problems can result when storing the data into a string variable.

If the data entered into a string corresponds to a command that the 9831A understands, the machine executes the command rather than storing the data into the string. This is true even if only the first part of the data corresponds to a command. Thus, if you use ENTER (0,*) A$ and type in automobile, it will take the word "auto" as a command and exit the program and display line number 10. This could be especially troublesome if the first five characters are "erase" as the 9831A will erase the mainline memory.

If data is entered from the keyboard and contains leading spaces, the 9831A drops as many trailing characters as there are leading spaces. Consider the following:

```
10  DIM A$(32)
20  ENTER (0,*) A$
30  DISP A$
40  END
```

Results –

Input Data	Display
123	123
Δ123	Δ12
ΔΔ123	ΔΔ1

NOTE: Δ = Space

Leading spaces in front of "commands" do not change the way the 9831A reacts; the commands are executed.

These problems occur only when doing an ENTER from the 9831A keyboard (select code zero). Entering data from some other select code works as expected.

The purpose of allowing the ENTER statement to use select code zero, is so a program can be written to allow data to come from one of several sources, e.g., keyboard or card reader. The program could use common routines for either entry point and simply change the select code, e.g., `ENTER (S, *) A$`. Using the INPUT statement where data is entered from the keyboard and the ENTER statement for peripheral devices, the problems stated above are avoided.

Chapter 2

General I/O Operations

Introduction

This chapter describes each of the I/O statements and functions which can be used to control external devices. Many of the operations introduced in this chapter are also covered in the 9831A Operating and Programming Manual. A list of error codes and a summary of I/O syntax is at the back of this manual.

When you are using the 9831A to control HP 9800-series peripherals, refer to the appropriate peripheral chapter of this manual for further explanation.

The I/O operation described in this chapter are classified as statements, functions or commands. The statements are programmable operations. Some statements allow the use of functions to obtain greater flexibility. Commands are keyboard operations and are not programmable. Each type is described in a separate section of this chapter.

Syntax Conventions

The rest of this chapter discusses each BASIC I/O operation. Here is a list of conventions and terms to be used in the BASIC syntax.

brackets [ ] – items within brackets are optional.

dot matrix – items printed in dot matrix must appear as shown.

character – a letter, a number or symbol.

constant – a number within the machine's computing range.

expression – a constant (like 16.4), a variable (like B or D [6]) or an expression (like 8*A2 or A = 6).

select code – an expression from 2 thru 15. An HP Interface Bus (HP-IB) address code can be added to the select code.

text – a series of characters within quotation marks.

variable – a simple variable (like B or B7) or an array variable (like F [9]).

Throughout this manual the term "print format" refers to the same format used with the PRINT statement. In the output statements the asterisk (*) designates the print format whereas with input statements the (*) designates free-field format.

The PRINT, LIST, TLIST, TRACE and print-all operations are automatically output to the device assigned as the "standard printer" (see Chapter 8 of the 9831A Operating and Programming Manual).

I/O Statements

The WRITE Statement

WRITE (select code , * or line number) [text and expressions]

The list of text and expressions is output to the specified device. Remember that the select code range is from 2 thru 15.

The * specifies use of the print format. With the print format the TAB, LIN (linefeed) and SPA (space) functions can be used, with commas or semicolons to determine the spacing between output items. A line number in place of the * references a FORMAT statement for enabling more complete output formatting. The line number must be a constant. When a FORMAT statement is referenced, each output item must be separated by a comma. A semicolon at the end of the WRITE statement suppresses the carriage return and linefeed (CR/LF) automatically sent after the last item.

Here are examples of the WRITE statement using the Print Format and FORMAT statement.

Print Format

```
30 WRITE (2,*) A,B,C
```

Format Statement

```
20 FORMAT F10.2
30 WRITE (2, 20) A,B,C;
```

Select Code

Format Line Number

Suppresses CR/LF

The FORMAT Statement

FORMAT specification$_1$, [specification$_2$, ...]

The FORMAT statement provides numeric and edit specifications for WRITE, OUTPUT and ENTER statements. The FORMAT statement can be located anywhere in the program. Formatting of numbers and spacing between successive items is easily controlled by referencing the FORMAT statement.

Numeric Specification

Numeric specs determine the form in which each numeric variable or constant is input or output. Numbers can be input or output in fixed point or floating point. The number of digits to the right of the decimal point and the character field width can also be controlled.

Here are the numeric specs –

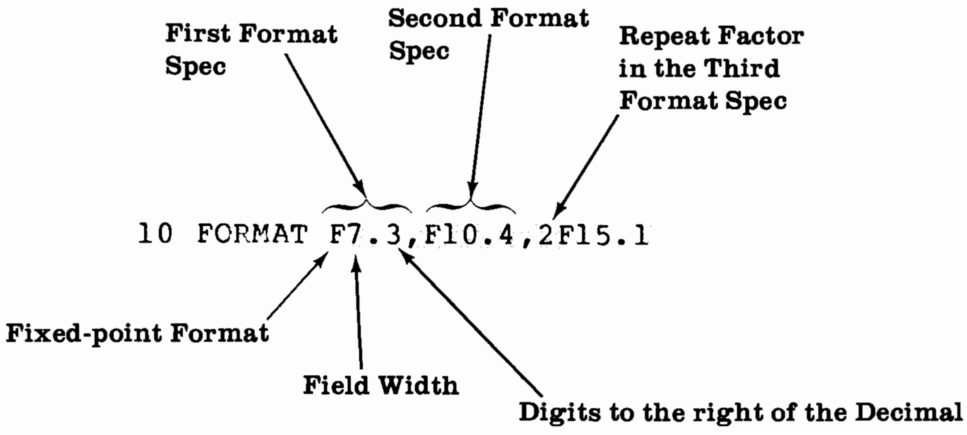
[r] Fw.d	F Specifies fixed-point format
[r] Ew.d	E Specifies exponential (floating-point) format
[r] B	B Specifies binary output of single characters

- r is an optional repeat factor. If r is omitted, 1 is assumed.
- w indicates total field width, in characters.
- d indicates the number of digits to the right of the decimal point.
- w, d and r parameters must be positive integer constants.

Here is a short program and output to explain the FORMAT statement further.

```
10 FORMAT F7.3,F10.4,2F15.1
20 A=23.25632
30 B=365.3445135
40 C=66.6666667
50 D=144
60 WRITE (2,10)A,B,C,D
70 END
```

```
 23.256  365.3445           66.7          144.0
```


First Format Spec
Second Format Spec
Repeat Factor in the Third Format Spec
10 FORMAT F7.3,F10.4,2F15.1
Fixed-point Format
Field Width
Digits to the right of the Decimal

The numeric spec F7.3 specifies a fixed-point number with three digits to the right of the decimal point. The number appears (right justified) in a seven character field. If d is 0 the decimal point will not be printed. The numbers are also rounded to the number of decimal places specified.

Any combination of specs can appear in the same FORMAT statement, but each spec must be separated by a comma. The same spec can be repeated by using the repeat factor.

In selecting a number for the field width remember that the sign (+ or −), decimal point and exponent are part of the number and must not exceed the specified field width. The exponent requires four character spaces (e.g., E ± 00). If the field width is exceeded it is filled with $'s and the number is not output.

For more information and examples on formatting, including exponential (floating point) refer to "FORMAT" in the 9831A Operating and Programming Manual.

Edit Specifications

Edit specs are used to control the placement of output data and character strings. [r] is an optional repeat factor.

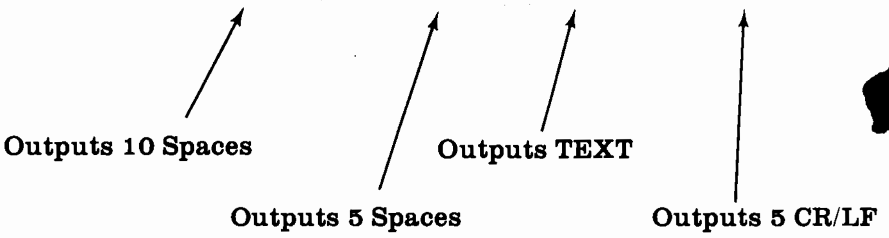
[r]X	X outputs a blank character space
[r]/	/ outputs a CR/LF
[r]"TEXT"	"TEXT" outputs text

The following program and output shows a combination of numeric and edit specs used in one FORMAT statement.

```
10 FORMAT 10X,2F7.2,5X,3"welcome  ",5/
20 A=46.56558
30 B=45.34679
40 WRITE (2,10)A,B
50 END
```

```
                    46.57  45.35     welcome  welcome  welcome
```

```
10 FORMAT 10X,2F7.2,5X,3"welcome  ",5/
```



The OUTPUT Statement

OUTPUT (select code or string name ; * or line number [; conversion table]) expression or text$_1$; [expression or text$_2$; ...]

OUTPUT is a general purpose statement used to send data or coded commands to an external device. If the external device requires non-ASCII code, conversion from ASCII to that code may be implemented through the use of the optional conversion table parameter. Conversion tables are described later in this chapter.

The line number references a FORMAT statement. If an asterisk (*) is used, it references the print format. An automatic carriage return and linefeed (CR/LF) is output at the end of the OUTPUT statement. To suppress them, end the OUTPUT statement with a comma or semicolon.

Output into a String Variable

If a string variable is specified instead of the select code, strings and variables can be output into a string variable instead of an external device. This is useful for generating characters which normally cannot be stored in the string directly. The conversion parameter cannot be specified when outputting into a string variable.

The following program stores the values between 1 and 255 into string variable A$. Then the string is printed, 32 characters at a time.

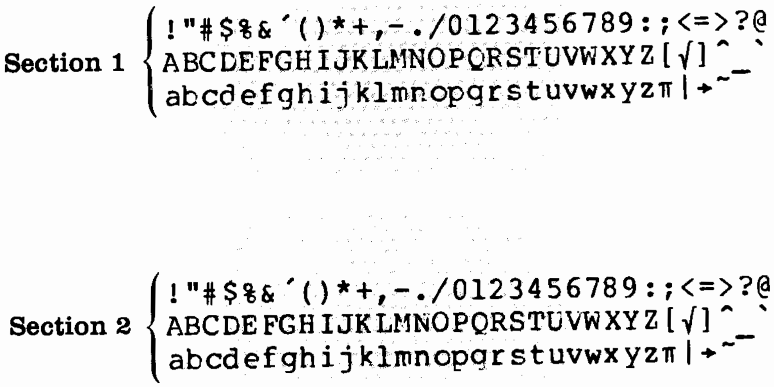
Output binary codes into A$, one character at a time.

```
10 DIM A$[255]
20 FORMAT B
30 FOR I=1 TO 255
40 OUTPUT (A$[I,I],20)I;
50 NEXT I
```

Output A$, 32 characters at a time.

```
60 I=1
70 OUTPUT (2,*)A$[I,I+31]
80 I=I+32
90 IF I+32<255 THEN 70
100 OUTPUT (2,*)A$[I,255]
110 END
```

Printout using a 9871A Printer –



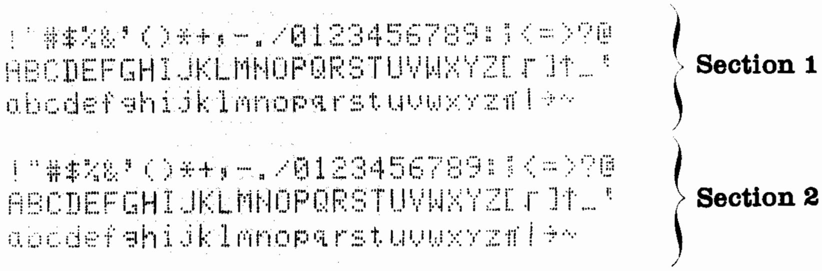
Section 1

```
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[√]^_`
abcdefghijklmnopqrstuvwxyzπ|→~
```

Section 2

```
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[√]^_`
abcdefghijklmnopqrstuvwxyzπ|→~
```

Printout using a 9866B Printer –



Section 1

```
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[r]↑_'
abcdefghijklmnopqrstuvwxyzπ|→~
```

Section 2

```
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[r]↑_'
abcdefghijklmnopqrstuvwxyzπ|→~
```

Note that many characters are missing (1 thru 31). This is because many ASCII characters are special characters. The 9866B Printer does not have characters corresponding to these special characters. Also, the output has two identical sections. This is because ASCII is an 8-bit code, but only 7 bits are data bits; the eighth bit is parity. Since the parity bit is ignored, the same characters are printed twice.

A different output device, such as the 9884A Tape Punch, could be specified in lines 70 and 100 where the eighth bit is used for data. If that device uses the eighth bit, then up to 256 characters would be available.

Suppressing CR/LF's

The 9831A automatically outputs a carriage return and linefeed (CR/LF) after completing each PRINT, WRITE or OUTPUT statement. These CR/LF's can be suppressed by ending the output statement with a comma or semicolon. A CR/LF is also output after the last format spec in a referenced format line, if the output statement has remaining items to be output.

Once the CR/LF has been suppressed as just mentioned (with a comma or semicolon) the 9831A will automatically output a CR/LF when it stops running (STOP, END or error detection). This final CR/LF is sent to the last device involved in an input/output operation.

When controlling a device which cannot accept the CR/LF characters, that device must not be the last device involved in the input/output operation of the program. In the following example the final CR/LF is sent to the display and not to the device at select code 2.

```
10 I=65
20 OUTPUT (2,*)WBYTEI;
30 DISP
40 END
```

The final CR/LF can be suppressed by referencing a FORMAT statement and placing a semicolon after the output statement. The FORMAT statement (Format B) is used in the following example, but Format 2F6.0 would have worked just as well.

```
10 FORMAT B
20 FOR I=65 TO 90
30 WRITE (2,10)I;
40 NEXT I
50 END
```

Both Required to Suppress CR/LF

There is no output on the 9866A/B Printer, but the characters are stored in the printer's buffer awaiting a CR/LF which will cause them to be printed. The 9871A Printer will print the characters, but will not perform a CR/LF.

Suppressing Leading Blanks

A special FORMAT spec causes leading blanks to be suppressed when using the OUTPUT statement. The FORMAT spec Fw.d or Ew.d is used with w equal to 1000 and d equal to the number of decimal places to the right of the decimal point. For example –

```
10 OUTPUT (2,20)-123.456789876
20 FORMAT F1000.9
40 END

-123.456789876
```

The value for the field width (w) can also be 1001 thru 1014 to specify the maximum field width (1 thru 14). The field width (w) specified by 1000 allows for 13 characters, the first character is reserved for a ± sign. A specified field width of 1001 thru 1014 can be used to limit the number of characters that are output.

If the value to be output is too large for the field, "$" will fill the field.

In this example, the output values are not too large for their fields –

```
10 OUTPUT (2,20)12345,-39
20 FORMAT 2F1007.0
30 END

12345-39
```

The values are too large for their fields here –

```
10 OUTPUT (2,20)12345,-39999
20 FORMAT 2F1005.0
30 END

$$$$$$$$$$
```

There is no field overflow when using 1000 for the field width. Integer numbers that exceed the field width are expressed in scientific notation and decimal values are rounded.

The ENTER Statement

ENTER (select code , * or line number [, conversion table])
variable$_1$ [, variable$_2$, ...]

The ENTER statement enables the 9831A to receive data from an external device. If the incoming data is not in ASCII code, conversion to ASCII code may be included in the ENTER statement, through the use of an optional conversion table parameter. An optional implied FOR/NEXT loop may be used to input multiple data items from one record into an array. (A record is a sequence of characters ending with a linefeed (LF).)

The ENTER statement causes the list of variables specified to be read from the device indicated by the select code. Character-by-character conversion to ASCII code is performed, if a conversion table is specified. Use of conversion tables is described later.

The line number references a FORMAT statement; an asterisk (*) indicates free-field format.

The variables can be simple variables, array variables, string variables or an implied FOR/NEXT loop. The implied FOR/NEXT loop follows this form –

(FOR variable = value$_1$ TO value$_2$[STEP value$_3$] , array name (variable))

For example – (FOR I = 1 TO 20 STEP 2, A(I))

This program uses the implied FOR/NEXT loop and is the same as the next one that follows. The implied FOR/NEXT loop can be used whenever delimiters are used to separate data in a record.

```
10 FOR I=1 TO 2
20 ENTER (3,*)(FORJ=1TO5,A[J])
30 PRINT A[1],A[2],A[3],A[4],A[5]
40 NEXT I
50 END
```

Free-Field Format

Free-field format is designated by an asterisk (*) in the ENTER statement. Characters are entered as ASCII characters. All characters except 0 thru 9, +, – and E are number delimiters. The linefeed (LF) is the record delimiter. Spaces are ignored.

For example, to read this data sequence (two records) –

1.3,2E-9,3,,4(LF)B1,2,3E5+2(LF)

Run this program –

```
10 FOR I=1 TO 2
20 ENTER (3,*)(FORJ=1TO5,A[J])
30 PRINT A[1],A[2],A[3],A[4],A[5]
40 NEXT I
50 END
```

Print out –

```
1.3             2.00000E-09     3          0              4

0               1               2          300000         2
```

When a + or – sign follows a number, as "2E–9" above, it is interpreted as exponential format. Two consecutive delimiters cause a zero to be entered, as seen with the two consecutive commas in this example, and also the "LF" followed by "B".

Input FORMAT Specifications

When a line number is specified in an ENTER statement, it references a FORMAT statement rather than the free-field input format. Refer to "The FORMAT Statement" in this chapter for more details on FORMAT syntax.

Numeric Input Specifications

[r]Fw.d	Fixed-point enters w number of characters with d number of decimal places. If the decimal point is part of the number being entered, then the d specification is ignored. If an "E" is in the field being entered, followed by one or two numbers, then the field is input using exponential format.
[r]B	The binary spec enters each byte (8 bits) as the decimal equivalent of a binary number, not as an ASCII character. For example, if the binary number 01000001 is entered, the decimal value 65 would be stored.

The following program can be used to read the data record from paper tape using a tape reader. For ease of use the select code is changed to 0 using the keyboard to enter the data. Pressing the execute key initiates the linefeed (LF) and continues the program. See the NOTE at the end of Chapter 1.

The FORMAT spec 4F4.2 is repeated 4 times. The 9831A looks for four spaces and places a decimal point after the first two.

For example, to read this data record –

123456E7-123987.(LF)

Run –

```
10 DIM C[4]
20 FORMAT 4F4.2
30 ENTER (0,20)C[1],C[2],C[3],C[4]
40 PRINT C[1],C[2],C[3],C[4]
50 END

 12.34          560000000      -1.23           987
```

Notice that reading the "E" in the data is interpreted as exponential format. Note, also that the decimal point in "987." overrides the FORMAT spec F4.2

Input Edit Specifications

[r]X The skip character spec causes individual characters to be skipped. For example, if four characters were to be skipped, 4X would be the spec used.

[r]/ The skip record format causes all the characters between two consecutive linefeeds to be skipped. For example, a format spec of 3/ would cause 3 records to be skipped.

NOTE

Text (quote fields) should not be used in a FORMAT statement referenced by an ENTER statement, since ERROR 73 (Improper FORMAT specification) results.

When an ASCII "←" (decimal 95) is encountered, a backspacing operation is simulated. For example, if the input characters are 12 ← 34, the resulting data value would be 134.

When an escape (ESC) character (decimal 27) is encountered, a search is made for a linefeed marking the end of the record. The entire record between the ESC character and the LF is ignored and the next record is entered.

I/O Functions

The WBYTE Function

WBYTE value

The WBYTE (write byte) function is used in the WRITE, PRINT and OUTPUT statements to output a single 8-bit binary equivalent of a decimal value. The WBYTE function provides a convenient method of controlling peripherals when used with WRITE, PRINT or OUTPUT statements. The following example outputs A thru Z using each output statement. Notice that WBYTE is also used to advance the printer paper. If the select codes were changed to output to different peripherals, each peripheral would have received the same data.

```
10 FOR I=65 TO 90
20 PRINT WBYTEI;
30 NEXT I
40 PRINT WBYTE10
50 FOR I=65 TO 90
60 OUTPUT (2,*)WBYTEI;
70 NEXT I
80 OUTPUT (2,*)WBYTE10
90 FOR I=65 TO 90
100 WRITE (2,*)WBYTEI;
110 NEXT I
120 WRITE (2,*)WBYTE10
130 END
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

The RBYTE Function

RBYTE select code

The RBYTE (read byte) function inputs one byte (an 8-bit character) of data from the specified peripheral device and returns the decimal-equivalent value. For example A = RBYTE2 inputs one data byte and stores its decimal equivalent value in A. RBYTE can also be used to input single key codes from the 9831A keyboard when select code 0 is specified.

As an interesting exercise, run this program sequence –

```
10 DISP RBYTE0
20 WAIT 300
30 GOTO 10
```

Now press each key, except RESET or STOP, to display its decimal key code. To halt the operation, press STOP. Notice that the decimal key codes returned by using this method always correspond to ASCII decimal codes. The ASCII decimal codes are in the Appendix.

The STAT Function

```
STAT select code
```

The STAT (status) function reads the status of the specified interface and returns a decimal value. The value, or status byte, is a combined code indicating the condition of individual status bits. Each bit can be either 0 (false) or 1 (true), depending on various peripheral status conditions.

For example, the 98032A Peripheral Interface has nine status bits which can be monitored using read status –

BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Peripheral Status	0	0	1	0	0	0	Data Ready	Buffer Ready

- Bits 0, 1 and 8 - Indicate states of optional peripheral status-input lines.
- Bits 2 and 3 - Indicate states of logic levels preset on the 98032A.
- Bits 4 and 5 - Are preset to 0 and 1, respectively, on the 98032A.
- Bits 6 and 7 - Indicate operating states on the interface card which are not controlled via the standard 9831A; these bits are always 0.

Notice that each bit has a unique binary value or weight: bit 0 has a true value of 1; bit 2 is 2; bit 5 is 32; and bit 8 is 256. The STAT function returns the sum of all bits currently true.

The 98032A Option 166 Interface is used with the HP 9866A/B Printer and uses only status bits 4, 5 and 8. Bits 4 and 5 are the interface I.D. (bit 4 = 0 and bit 5 = 1), while bit 8 remains at 1 except when the printer is out of paper. Here's a program segment that checks for bit 8 and stops when the printer is out of paper –

```
100 IF STAT2>256 THEN 115
105 DISP "PRINTER NOT READY"
110 STOP
115 PRINT A
```

I/O Commands

The LIST Command

LIST [# select code ;] [first line number [; last line number]]

The LIST command outputs program lines to either the standard printer (no select code specified) or to the specified select code. The optional line numbers can be used to list a limited number of program lines. To list a program onto paper tape at select code 9 execute the command –

```
LIST #9
```

For further explanation of the LIST command refer to Chapter 4 of the 9831A Operating and Programming Manual.

The PTAPE Command

PTAPE select code
or
PTA select code

The PTAPE (paper tape) command is used to input program lines from an external device. Compatible devices include the HP 9863A or 9883A Tape Readers, and the HP 9869A Card Reader.

If the device with the specified select code is not ready the 9831A waits until the device is ready, then completes the command. During this time the display remains blank.

When the program transfer is complete, press STOP to halt the command. PTAPE is a command and therefore is not programmable.

The PTAPE command inputs program lines written only in HP 9831A or HP 9830 BASIC language which are transmitted in ASCII code. The LIST command on the 9830 outputs program lines in the correct format. PTAPE cannot be used to input binary-coded programs.

The 9831A automatically syntax-checks each program line as it is loaded via PTAPE; if an error is seen, the line is rejected. When the print-all mode is on, each rejected line is listed during a PTAPE operation.

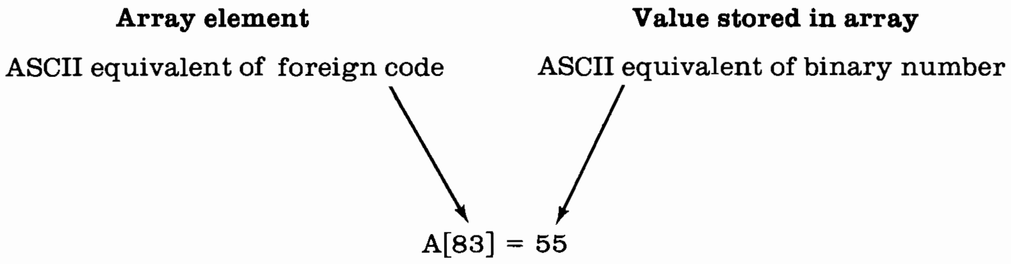
Conversion Tables

A conversion table can be used with the ENTER and OUTPUT statements to convert groups of characters, or arrays of data, from one code to another. As mentioned earlier, the 9831A makes use of standard ASCII codes. Throughout this manual all non-ASCII codes (used by printers, card readers, paper tape readers, punches, etc.) are referred to as "foreign" codes.

A conversion table must first be defined in a DIM statement as an integer array. For example –

```
10 DIM AI(150)
```

To create a conversion table, you must store the codes in an integer array. The array element correspond to the decimal equivalents of the characters in the foreign code. The decimal equivalents of the required ASCII characters are then stored in the appropriate array elements. The following is a simplified explanation of how a conversion table works.



Array element

ASCII equivalent of foreign code

Value stored in array

ASCII equivalent of binary number

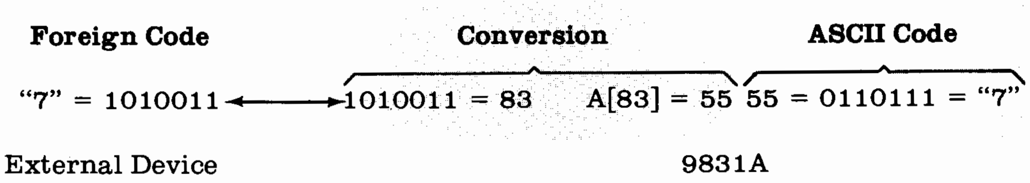
A[83] = 55

For example, assume that only the character "7" is to be input and output and that the external device uses some arbitrary foreign code. Suppose that in the foreign code the character "7" is represented by the binary number 01010011. The ASCII table in the Appendix shows that 01010011 is equivalent to decimal 83, so 83 is used to determine the array element used (A[83]).

ASCII "7" is represented by the binary number 00110111; this is the bit pattern which the 9831A normally outputs (or receives) for the character 7. The ASCII table shows that 00110111 is equivalent to decimal 55. 55 would then be stored in array element A[83]. A[83] = 55.

When the external device inputs a 7, it sends bit pattern 01010011. The 9831A interprets this as 83, so it looks at array element number 83 and finds decimal 55 (see diagram below). The equivalent bit pattern of 55 is 00110111, which is interpreted as the ASCII character 7. When the 9831A is to output the character 7, it looks for 55 in the array and finds it in the 83rd element. So the 9831A outputs the bit pattern for 83, which is 1010011. The external device interprets that bit pattern as the character 7.

A is the conversion table



Conversion in the ENTER Statement

```
ENTER (9,*,A)
```

When a conversion table is referenced by an ENTER statement, the incoming foreign character codes are used as array element values to look up the ASCII equivalents in the conversion table. If an array element value is referenced which is not defined in a conversion table, ERROR is displayed. It is important that the linefeed character be defined in the conversion table. For example A(108) = 10 would define the foreign code 108 as the ASCII linefeed (decimal 10).

Conversion in the OUTPUT Statement

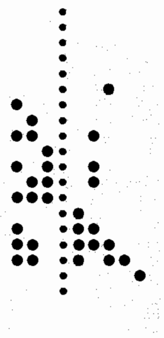
```
OUTPUT (3,*,A)
```

When the conversion table is referenced by an OUTPUT statement, the contents of the conversion table are searched sequentially for the outgoing ASCII character, and the array element value is the foreign code to be output. If a code is not found in the conversion table, no character is output.

Example of Code Conversion

Suppose you have a paper tape reader which uses EIA[1] coded tape. X and Y values are contained on the tape, separated by commas. The following table shows the tape punched with allowable codes, binary representation, the character, and the decimal equivalent codes for the EIA codes contained on the tape.

[1] Electronics Industries Association standard code.

EIA Coded Tape	Binary	Character	EIA Decimal Code Equivalent
	00100000	0	32
	00000001	1	1
	00000010	2	2
	00010011	3	19
	00000100	4	4
	00010101	5	21
	00010110	6	22
	00000111	7	7
	00001000	8	25
	00111011	,	59
	01101011	.	107
	10000000	CR	128

From the ASCII table in the Appendix, you can write a chart showing the ASCII decimal equivalents of the EIA codes –

Character	EIA Decimal Equivalent (Subscript)	ASCII Decimal Equivalent (Value)
0	32	48
1	1	49
2	2	50
3	19	51
4	4	52
5	21	53
6	22	54
7	7	55
8	8	56
9	25	57
,	59	44
.	107	46
delimiter	128 (CR)	10 (LF)

Notice that the ASCII linefeed (LF) is the delimiter used by the 9831A. This delimiter corresponds to the carriage return (CR) of the EIA code.

Now a conversion table can be made directly from the preceding EIA/ASCII table. First, dimension an integer array for the conversion table –

```
10 DIM AI[128]
```

The conversion table must be dimensioned to have at least 128 array elements, since 128 is the largest element to be used (the EIA column in the table).

Next the elements of the conversion table are defined –

```
20 A[32]=48
30 A[1]=49
40 A[2]=50
50 A[19]=51
60 A[4]=52
70 A[21]=53
80 A[22]=54
90 A[7]=55
100 A[8]=56
110 A[25]=57
120 A[59]=44
130 A[107]=46
140 A[128]=10
```

With the above instructions, definition of the conversion table is complete. Many elements in array A are undefined, since the table does not have as many symbols as there are spaces for elements in the array.

The following statement inputs two characters and automatically converts the code –

```
150 ENTER(9,*,A)A,B
```

If any foreign code is not defined in the conversion table, an ERROR will occur during the ENTER statement.

Notice that a variable A and a conversion table A can appear in the same program. With the above statement, EIA characters for the variables A and B are read from the paper tape reader (select code 9) and converted to ASCII code by conversion table A.

Now assume that you have to output your variables to a tape punch at select code 3. To output variables A and B again, they must first be converted back to EIA code. The following statement will do this –

```
160 OUTPUT(3,*,A)A,B
```

Note that the same conversion table is used in this case for both input and output.

Binary Functions

Three functions are available for manipulating numbers using their 16-bit binary values. The usable range of each value is from −32768 thru 32767

The Binary And (BIAND) Function

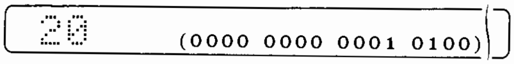
BIAND (decimal value$_1$; decimal value$_2$)

The BIAND function combines the given values using the logical AND operation and returns the result. A truth table of the AND operation is shown on the right

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

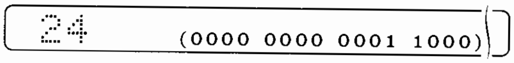
For example, to AND decimal 20 and 24, execute these lines –

```
10 A=20
```



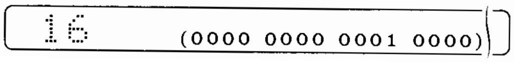
20 (0000 0000 0001 0100)

```
20 B=24
```



24 (0000 0000 0001 1000)

```
30 C=BIAND(A,B)
```



16 (0000 0000 0001 0000)

The Inclusive OR (INOR) Function

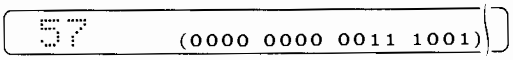
INOR (decimal value$_1$, decimal value$_2$)

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

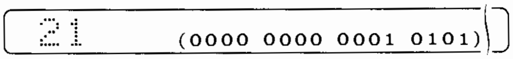
The INOR function combines the given values in an inclusive OR operation and returns the result. A logical OR truth table is shown on the right.

For example, to combine 57 and 21 in an inclusive OR operation –

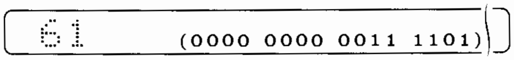
```
10 A=57
```


57 (0000 0000 0011 1001)

```
20 B=21
```


21 (0000 0000 0001 0101)

```
30 C=INOR(A,B)
```

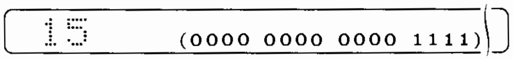

61 (0000 0000 0011 1101)

The Rotate (ROT) Function

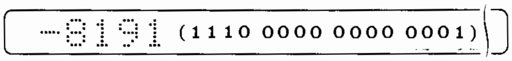
ROT (decimal value , number of places)

The ROT function right-rotates a 16-bit integer value a specified number of places and returns the result. No bits are lost in the rotation. Although the number of places rotated can be greater than 15, the actual rotation is from 0 to 15 locations. For example –

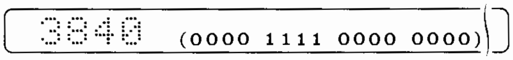
```
10 A=15
```


15 (0000 0000 0000 1111)

```
20 B=ROT(A,3)
```


-8191 (1110 0000 0000 0001)

```
30 C=ROT(A,8)
```


3840 (0000 1111 0000 0000)

As another example, here is a method to combine two 8-bit values (A & B) into one 16-bit value (C). Notice that A and B can each be an integer from 0 thru 255.

```
C=INOR(ROT(A,8),B)
```

Later, A and B can be separated by executing –

```
A=BIAND(ROT(C,8),255)

B=BIAND(C,255)
```

I/O Applications

This section contains some sample I/O applications, such as external data transfer, parity generation and absolute tape. When controlling an HP 9800-series peripheral, refer to the operating manual furnished with the peripheral device and one of the appropriate peripheral chapter in this manual.

External Data Transfer

You can perform direct, byte-by-byte transfer of data from an input device to an output device without code translation. For example, using an input device with select code 7 and an output device with select code 9, this program reads a byte of data from select code 7 and outputs the byte to select code 9 using binary format.

```
10 WRITE (9,20)RBYTE7;
20 FORMAT B
30 GOTO 10
```

Parity Generation

The following program creates a parity conversion table for sending data with even parity.

```
10 DIM AI[256]
20 FOR I=1 TO 256
30 A[I]=0
40 NEXT I
50 FOR I=10 TO 127
60 D=0 ←──────────────────── Initialized Bit Counter
70 FOR J=1 TO 7
80 IF ROT(I,J)>0 THEN 100←── Checks for 1 in Most Significant Bit
90 D=D+1←─────────────────── Add 1 to Bit Counter
100 NEXT J
110 IF BIAND(D,1)#0 THEN 140←Check if D is Odd or Even
120 A[I]=I←───────────────── I Has Even Parity
130 GOTO 150
140 A[128+I]=I←───────────── I Has Odd Parity
150 NEXT I
```

To use the conversion table, you would include a statement such as –

```
160 OUTPUT (2,*,A)"ABCDEF12345!#$%"
```

The following lines can be used to obtain a printout of the values in the parity conversion table.

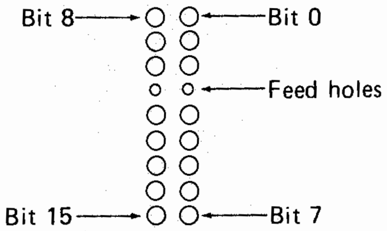
```
170 FOR I=1 TO 255 STEP 4
180 FOR J=I TO I+3
190 PRINT "A("J")="A[J];
200 NEXT J
210 PRINT
220 NEXT I
```

To create a conversion table for odd parity, statement 110 above should be changed to –

```
110 IF BIAND(D,1)=0 THEN 140
```

Absolute Tape

Absolute tape contains instructions or data in word lengths (2 bytes of 8 bits each), where each word is represented by two frames on the paper tape –



Assuming select code 9 for the tape reader, the instruction for the 9831A to read one word (two frames) from the tape into X would be –

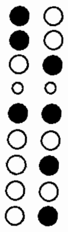
```
50 X=INCR(ROT(RBYTE9,8),RBYTE9)
```

To simplify the preceding statement, let's use A and B as temporary variables and separate the operations in the above instruction into separate program steps –

```
1 B=RBYTE9
2 A=ROT(B,8)
3 B=RBYTE9
4 X=INOR(A,B)
```

For the sample data word at right, the binary representation of B after step 1 would be –

$$\overbrace{00000000}^{\textbf{BYTE}}\ \overbrace{00001011}^{\textbf{BYTE}}$$



A Sample Data Word

One byte is read from the tape into the low-order bits of B. The high order bits (15 thru 8) of B are initialized to 0.

Chapter 3

HP 9863A Tape Reader

Introduction

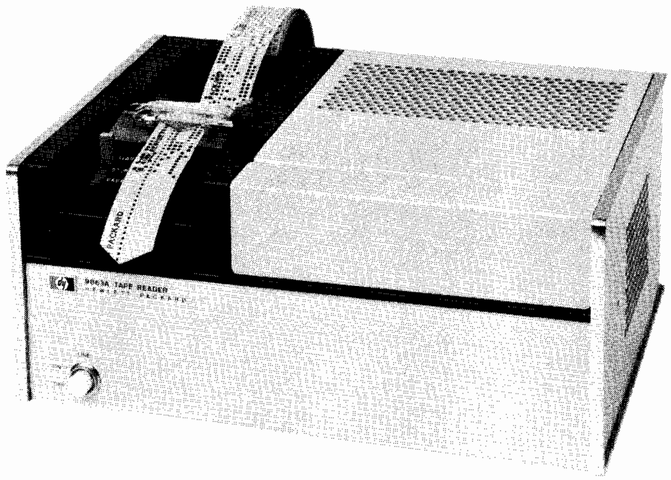
This chapter describes the general I/O operations used to control the HP 9863A Tape Reader: the ENTER statement, the RBYTE function and the PTAPE command. Numeric data, string data and programs can be entered from the tape reader using the above operations.

The HP 9863A Tape Reader is connected to the HP 9831A via a 98032A Option 163 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90041). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

Punched tape output from teletype machines, computer terminals and various measuring instruments can be read directly into the HP 9831A Desktop Computer using the HP 9863A Tape Reader.

Paper or mylar tapes up to 2.54cm (1 inch) wide can be read at speeds up to 20 characters per second by the 9863A. The tape reader accepts ASCII or ISO codes. Depending on the source, the format of data on punched tape often differs (e.g., the use of various characters as delimiters). Data delimiters are defined by a convenient front panel patchboard (Data mode only) and can be any ASCII/ISO characters. This feature allows the 9863A to read a wide variety of punched codes.



9863A Tape Reader

It is assumed that you are familiar with the operation of the 9831A and HP 9863A Tape Reader, as explained in their operating manuals.

Here is a summary of the statements and functions used to control the 9863A Tape Reader. More information on these operations is in Chapter 2.

ENTER — Receives data from an external device; can reference a FORMAT statement or free-field format and can be used with a conversion table.

RBYTE — Inputs one byte of data from the specified peripheral device and returns a decimal equivalent value.

PTAPE — Command used to input program lines from an external device; press STOP to halt the command. PTAPE is not programmable.

Data Input

Although non-ASCII coded tape can be used with the 9863A Tape Reader, the examples in this section use ASCII coded tape only. Refer to "Conversion Tables" in Chapter 2 for information about the use of non-ASCII coded tape. Also, if you plan to use the tape reader for external data transfer, refer to "External Data Transfer" in Chapter 2.

ENTER (select code , * or line number [, conversion table])
variable$_1$ [, variable$_2$, ...]

Each ENTER statement causes a record to be read. A record is a sequence of data items, separated by commas or semicolons, which ends with a linefeed (LF). Commas and semicolons are character delimiters; the linefeed is the record delimiter. For example, this sequence of ASCII characters is from a tape. Two records are shown –

12.8,−300,.99,5E6(LF)15.0,198,.32,1.99E-6(LF)

This short program reads and prints the individual data items –

```
10 DIM Y[4]
20 FOR I=1 TO 2
30 ENTER (7,*)(FORJ=1TO4,Y[J])
40 PRINT Y[1],Y[2],Y[3],Y[4]
50 NEXT I
60 END
```

```
 12.8              -300              0.99             5000000
 15                 198              0.32             1.99000E-06
```

In the program, free-field format is used. Since commas are used to separate the data, there is no problem. But to read a record which contains only numbers (no character delimiters), such as –

972193425(LF)693345786(LF)667238425(LF)

a different format spec would have to be used. In this case if the numbers have three characters each, three numbers are contained in each record. To read the three records, the following program could be used.

This program reads and prints the individual data items –

```
10 FOR I=1 TO 3
20 ENTER (7,30)A,B,C
30 FORMAT 3F3.0
40 PRINT A,B,C
50 NEXT I
60 END
```

```
 972                193                425
 693                345                786
 667                238                425
```

In line 30, three characters are read; no numbers to the right of the decimal point are specified.

Using Strings

In addition to numeric data, alphabetic data can also be input using the tape reader. The following alphanumeric characters are entered as strings. The Δ indicates a blank space.

ROGERS,ΔJOHNΔHT5-10,WT160(LF)SMITH,ΔARTHURΔHT6-2,WT230(LF)

This program enters and prints the two records –

```
10 DIM A$[80]
20 FOR I=1 TO 2
30 ENTER (7,*)A$
40 PRINT A$
50 NEXT I
60 END
```

```
ROGERS, JOHN HT5-10,WT160
SMITH, ARTHUR HT6-2,WT230
```

The linefeed (LF) is the string delimiter. Note that commas are interpreted as part of the string that is entered rather than as data delimiters. To enter mixed string and numeric data, such as student names followed by grades –

ROGERS,ΔJOHNΔΔΔ80,75,83(LF)SMITH,ΔARTHURΔΔΔ90,93,88(LF)

This program can be used –

```
10 DIM A$[40]
20 PRINT "NAME","   ","AVG."
30 OUTPUT (2,40)
40 FORMAT 34"="
50 FOR I=1 TO 2
60 ENTER (7,*)A$[1,16],A,B,C
70 FIXED 0
80 PRINT A$,(A+B+C)/3
90 NEXT I
100 END
```

```
NAME                          AVG.
==================================
ROGERS, JOHN                   79
SMITH, ARTHUR                  90
```

In line 60, only 16 characters are entered into A$, then the numbers A, B and C are entered. In line 80, the student name, A$, and the average of A, B and C are printed.

Program Input

Programs which have been punched on paper tape, via the LIST command, can be entered into the 9831A using the PTAPE command.

PTAPE select code
or
PTA select code

When the program transfer is complete, press (STOP) to halt the command. PTAPE is not programmable.

The PTAPE command inputs program lines written in HP 9831A or HP 9830 Basic language and transmitted in ASCII code only. For example, store this short program in the 9831A.

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
40 END
```

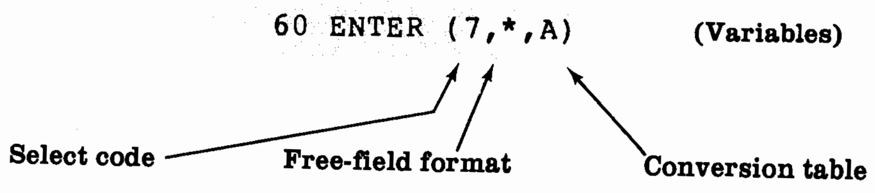
If a tape punch is at select code 9, then LIST #9 will store the program on paper tape. At a later time, the program can be re-entered into the 9831A from the 9863A Tape Reader, if select code 7 is used, then execute PTAPE 7).

Data Mode

The 9863A Tape Reader can be used in the data mode if a conversion table is used (see Conversion Tables in Chapter 2). A conversion table is necessary because the apostrophe (decimal 39) is sent as the end of character delimiter instead of the comma (decimal 44), and a comma (decimal 44) is sent as the record delimiter instead of a linefeed (decimal 10). The following lines generate the necessary conversion table –

```
10 DIM AI[70]
20 FOR I=1 TO 70
30 A[I]=I
40 NEXT I
50 A[44]=10
60 A[39]=44
```

In data mode, only numerics, plus and minus signs, and the letter E (decimal 69) for exponents are allowed. For this reason, only 70 characters are needed. In line 50, A(44) = 10 converts the decimal equivalent of a comma to the decimal equivalent of a linefeed. The ENTER statement used would appear as follows –



```
60 ENTER (7,*,A)
```

The following example illustrates the use of the 9863A Tape Reader in data mode.

Data entered –

1.3,2E−9,3,,4 LF B1,2,3E5,+2 LF

This program reads and prints the data items –

```
10 DIM AI[70]
20 PRINT " A"," B"," C"," D"," E"
30 FOR I=1 TO 70
40 A[I]=I
50 NEXT I
60 A[44]=10
70 FOR I=1 TO 2
80 ENTER (7,*,A)A,B,C,D,E
90 PRINT A,B,C,D,E
100 NEXT I
110 END
```

```
A               B               C          D          E
1.3             2.00000E-09     3          0          4
0               1               2          300000     2
```

NOTE

If the 9831A is switched off and on while the tape reader is ready to read, the tape may advance by one character.

3

Chapter 4

HP 9864A Digitizer

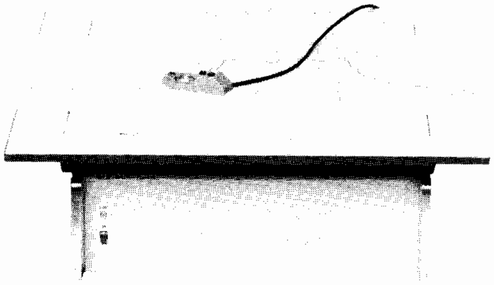
Introduction

This chapter describes the general I/O operations which are used to control the HP 9864A Digitizer.

The HP 9864A Digitizer is connected to the 9831A via the 98032A Option 164 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90042). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

The 9864A Digitizer is used to read a curve or any irregular shape as a series of discrete X-Y coordinates. The digitizer can read graphical data such as x-rays, blueprints, strip-charts or scope traces. The usable digitizing area is 17 x 17 inches (43.2 x 43.2 cm) minimum, and the resolution is 0.01 inch (0.25 mm).



9864A Digitizer

This chapter assumes that you are familiar with the operation of 9831A and the digitizer, as explained in their operating manuals.

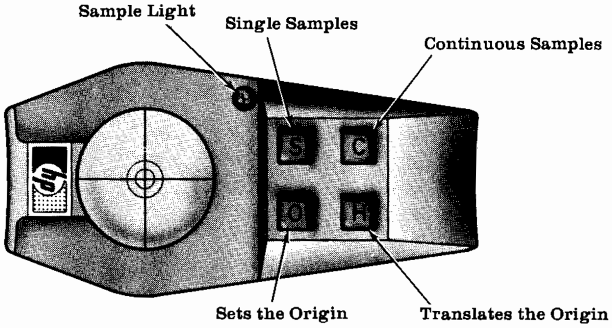
Two general I/O operations are used to control the digitizer; these are the ENTER and WRITE statements. Complete information on these operations are in Chapter 2.

ENTER — Receives data from an external device, can reference a FORMAT statement or free-field format.

WRITE — The list of text and expressions is output to the specified device, can reference a FORMAT statement or the print format.

Digitizer Controls

Except for the LINE switch on the mainframe, all digitizer controls are located on the cursor (see below). A brief review of these controls follows.



The Digitizer Cursor

Setting the Origin

The origin (coordinates 0,0) is set by placing the cross-hairs of the cursor over the point that is to be the origin, and then pressing (O). The origin may be set anywhere on the digitizing surface. The sample light indicates when the origin is set. Setting an origin cancels the Hold function described later.

If you move the cursor improperly, such as lifting it from the platen, the digitizer beeps and the sample light goes out indicating that the origin is lost. The origin must then be re-established before any further entries can be made.

Single Sample Mode

After an ENTER statement is executed, pressing (S) enters the current coordinates of the cursor cross-hairs. (S) must be released and pressed again to take another data sample. The sample light blinks each time (S) is used to take a data sample.

Continuous Sample Mode

When the digitizer is placed in the Continuous mode, each data request causes a data sample to be taken automatically. The digitizer is placed in the Continuous mode by pressing C.

The digitizer remains in the Continous mode until C is pressed again.

In the Continuous mode, the digitizer makes each entry as soon as the 9831A requests it. The Continuous mode is useful when digitizing a figure composed of curved lines, since you can simply trace around the figure. Be careful in moving the cursor, however, as any extraneous movements will result in the entry of erroneous coordinates.

Pressing O during the Continuous mode sets a new origin. Improper cursor movement halts data sampling (the origin is lost) but does not cancel the Continuous mode. Pressing RESET cancels the Continuous mode.

The Hold Function

Pressing H activates the Hold function. It will remain active until either H is pressed again, or O is pressed. While Hold is active you may remove the cursor from the digitizing surface without losing the origin.

The digitizer can take single and continuous mode samples while Hold is active. The coordinates entered, however, all correspond to the location of the cross-hairs at the time when Hold was activated.

The Hold function permits easy handling of documents larger than the digitizing surface. The Hold feature enables the operator to translate (shift) the origin in any direction. Translation is accomplished as a genuine movement, rather than by simply setting a new origin. It is possible to continue to translate the origin until it is no longer located on the digitizing surface. When this happens, the coordinates of points on the digitizing surface have ordinates and abscissas whose absolute values are greater than 17 inches (the size of the digitizing surface). By correctly translating the origin, it is possible to effectively locate the origin anywhere on the document, even though the document is larger than the digitizing surface.

The maximum values of X and Y coordinates when the origin is repeatedly translated, are ±99.99 inches. Further movement causes each value to "roll over" to +0.00. Refer to the Digitizer Peripheral Manual for more details on using the Hold function.

Maximum Sample Rate

The maximum rate at which the 9831A can request and accept data samples from the digitizer is over 170 samples per second. Since the sample rate may be considerably slower due to program execution time, you should take care to move the cursor slowly, in order to obtain the maximum practical sample density. The effects of sample rate and sample density are discussed in the Digitizer Peripheral Manual.

Entering Data

ENTER (select code , *) X coordinate , Y coordinate

The ENTER statement inputs one set of data coordinates from the digitizer. The free-field format is used, since each data item is followed by a non-numeric character. The input sequence for each data sample is shown below.

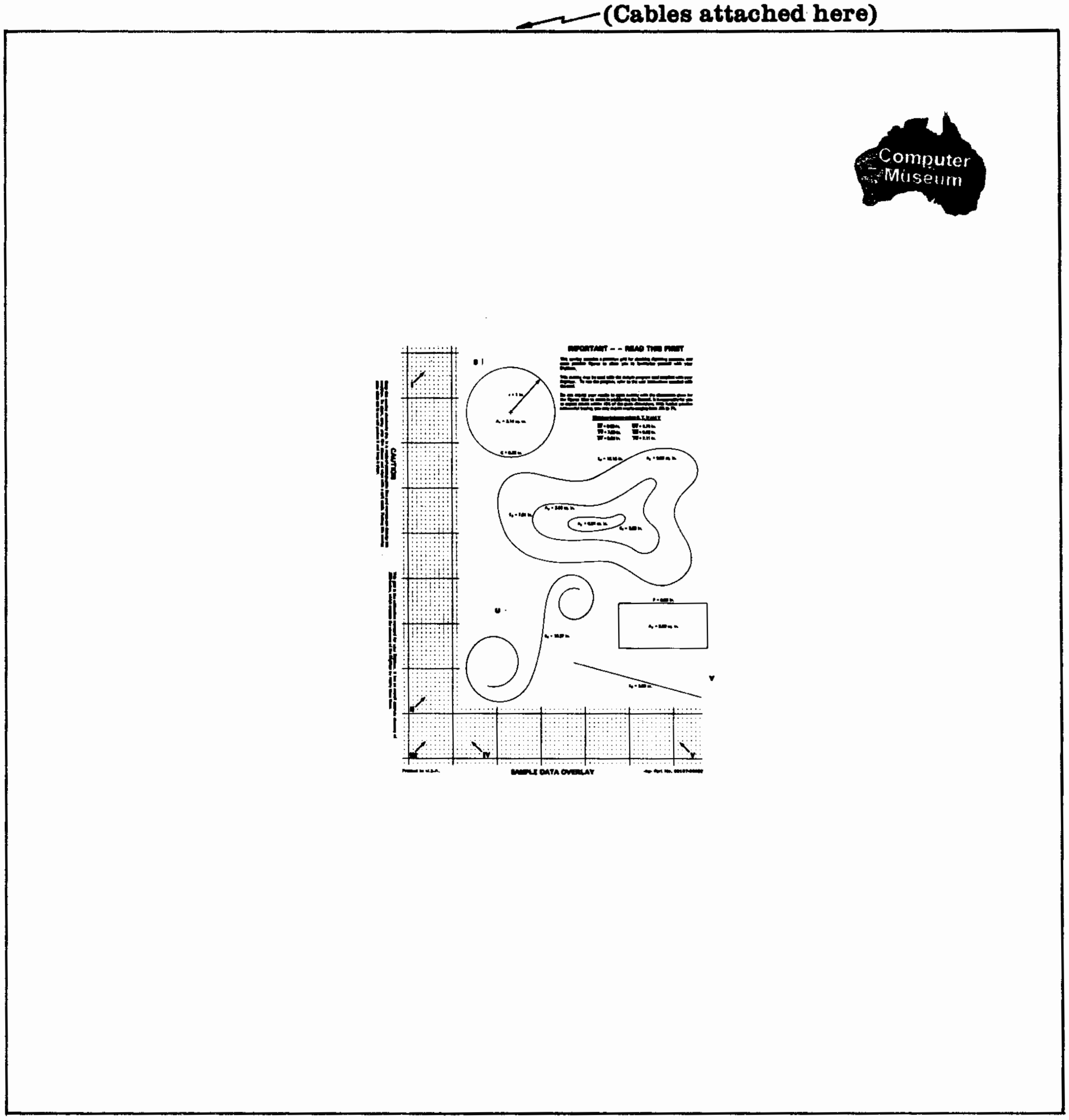
$$\underbrace{\pm DD.DD}_{X_{\text{coordinate}}},\underbrace{\pm DD.DD}_{Y_{\text{coordinate}}}\text{(CR)(LF)}$$

```
10 FIXED 2
20 ENTER (4,*)X,Y
30 DISP "X="X,"Y="Y
40 WAIT 1000
50 GOTO 20
60 END
```

This program continually inputs and displays data samples. The digitizer interface is assumed to be set to select code 4.

To run the program using the Sample Data Overlay supplied with digitizer –

1. Attach the overlay to the digitizer platen, as shown on the following page.
2. Set the origin (press $\boxed{\text{O}}$ on the cursor) approximately over point U on the overlay. Press $\boxed{\text{RUN}}$ $\boxed{\text{EXEC}}$. The 9831A is now waiting for a data sample from the digitizer.
3. To take continuous samples, press $\boxed{\text{C}}$ and slowly slide the cursor across the digitizing area; the corners of the digitizing area are indicated by the black dots on the platen. When the Continuous mode is used with this program, data points are being digitized at the rate of about 55 samples per minute.
4. To stop the sampling, but not the program, press $\boxed{\text{C}}$. Now press $\boxed{\text{S}}$ several times; the digitizer supplies one data sample each time $\boxed{\text{S}}$ is pressed.
5. To stop the program, press $\boxed{\text{STOP}}$.



(Front of the digitizer platen)

Sample Data Overlay on the Digitizer Platen

The Beep

```
WRITE (4,*)
```

The WRITE syntax causes the digitizer to sound a tone which lasts about one-tenth of a second. A series of these sequences separated by wait statements, produce a pattern of beeps which can be used to signal the operator during program operation. A practical use of the BEEP syntax is shown in the next program.

Document Alignment

This program can be used to align a document on the digitizing surface. The general procedure used here is described in the Digitizer Peripheral Manual.

```
10 FIXED 2
20 ENTER (4,*)X,Y
30 DISP "X="X
40 WAIT 1000
50 IF X>0.1 THEN 20
60 IF X<-0.1 THEN 20
70 IF X=0 THEN 100
80 WRITE (4,*)
90 GOTO 20
100 FOR I=1 TO 50
110 WRITE (4,*)
120 WAIT 10
130 NEXT I
140 END
```

1. Attach the Sample Data Overlay, or another document which is to be digitized, to the digitizing surface as shown on the preceding page.
2. Place the cursor cross-hairs exactly over point I (the upper left-hand corner of the document) and press (O).
3. Moving to the 9831A, press (RUN) (EXEC).
4. Slide the cursor to point III (the upper left-hand corner of the document); position the cross-hairs exactly over point III and press (C).
   If point III is currently within 00.1 inch of the X axis as established over point I, the digitizer will beep repeatedly.
5. Slowly move the cursor and the overlay (together), either right or left, until the display equals 0.00 and the continuous signal indicates alignment.
6. Without moving the overlay, tape the remaining three corners of the document to the platen. If necessary, retape the first corner.

Line Length Program

The following program calculates the length of a curved or a straight line represented by points digitized along the line. The line length is calculated by using the equation –

L=((Xi−Xj)↑2+(Yi−Yj)↑2)↑.5

```
10 DISP "LINE LENGTH PROGRAM"
20 WAIT 2000
30 DISP "DIGITIZE START POINT"
40 WAIT 2000
50 FIXED 3
60 X=Y=L=N=0
70 ENTER (4,*)A,B
80 DISP "DIGITIZE LINE"
90 WAIT 2000
100 ENTER (4,*)X,Y
110 N=N+1
120 L=L+((A-X)^2+(B-Y)^2)^0.5
130 A=X
140 B=Y
150 GOTO 100
160 PRINT "No. of Points ="N
170 PRINT "Line Length ="L
180 END
```

To run the program –

1. Establish an origin on the document by positioning the cursor and pressing O.
2. Start the program by pressing RUN EXEC.
3. To digitize the starting point, position the cursor at the beginning of the line and press S.
4. Now digitize the line, using either S or C.
5. After the last point is taken, turn off the Continuous mode (if it's on) and press STOP CONT 1 6 0 EXEC.

The final data is printed as shown here –

```
No. of Points = 55.000
Line Length = 2.542
```

Chapter 5

HP 9866A/B Printer

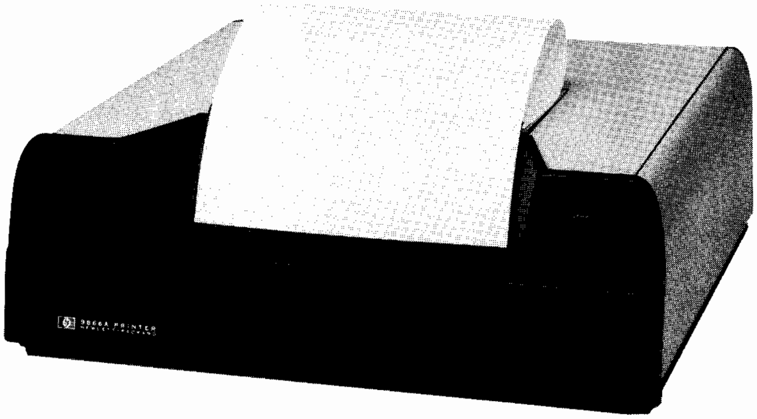
Introduction

This chapter describes the general I/O operations and unique operations which are used to control the HP 9866A and the HP 9866B Thermal Printers.

The HP 9866A/B Printer is connected to the HP 9831A via the 98032A Option 166 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90043). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

The HP 9866A and HP 9866B are thermal line printers providing fast, quiet data output. Both printers operate at 240 lines per minute and produce 80 characters per line, using a 5x7 dot matrix. The 9866A has a 65-character set. The 9866B printer has a 95-character set, including both upper and lower case alphanumeric. In addition to the standard alphanumeric output, the 9866B is capable of unidirectional plotting which is suitable for such things as strip charts and bar charts.



9866A/B Printer

Here are the statements and functions used to control the 9866A/B Printer.

STDPR
Specify the "standard printer" for PRINT, LIST, TLIST, TRACE and print-all operations.

PRINT
Print text numbers on the standard printer.

TAB and SPA
Output a specified block of spaces to position output items with PRINT and WRITE statements.

LIN
Outputs a specified number of linefeeds; use with PRINT and WRITE statements.

WRITE
Output text and numbers to any output device. Items are output in either the PRINT format or in the form specified by a referenced FORMAT statement.

FORMAT
Specify the form for variables output and specify edit specs for controlling an output device.

LIST
Outputs program lines to either the standard printer or another output device.

TLIST
Print, on the standard printer, information about each tape file.

The Standard Printer Statement

STDPR select code

Successive PRINT, LIST, TLIST, TRACE and print-all operations are automatically output to that device. The select code range is an expression from 2 thru 15. The 9831A assumes select code 2 for its standard printer when first switched on.

The Print Statement

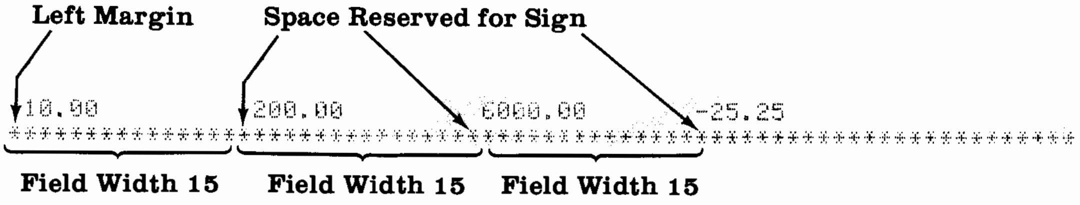
PRINT [any combination of text and expressions]

The PRINT statement enables text and variables to be output to the standard printer. Variables, expressions, constants and text (enclosed in quotation marks) can be included in the same PRINT statement, separated by either commas (for maximum spacing between successive outputs) or by semicolons (for minimum spacing between successive outputs). If no parameters follow PRINT, the printer performs a carriage return and linefeed (CR/LF). The maximum printout per line using PRINT is 72 characters.

Each value output by PRINT is left justified within its respective field, with the leftmost character space reserved for the sign. The current number format (STANDARD, FIXED or FLOAT) determines the form of each variable printed.

Here's an example, using commas, showing a field width of 15 characters and the space reserved for the sign.

```
10 FIXED 2
20 A=10
30 B=200
40 C=6000
50 D=-25.25
60 PRINT A,B,C,D
70 PRINT "************************************************************
80 END
```



```
 10.00          200.00         6000.00       -25.25
************************************************************
```

Now execute this line –

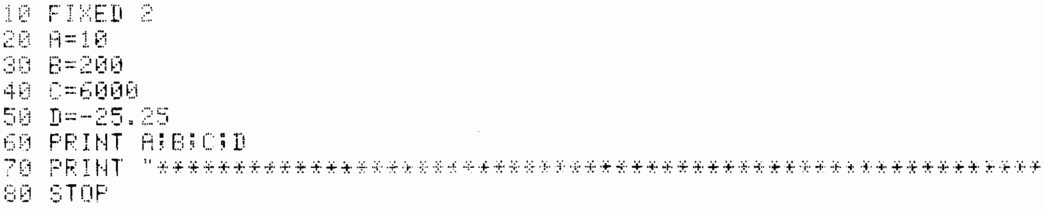
```
90 PRINT A,B*5,A+C,A*B*C,A/1E-09
```

Result –

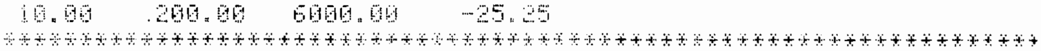
```
10.00          1000.00        6010.00        12000000.00
10000000000.00
```

Notice that a CR/LF is given after four data items are printed. Then another CR/LF is given after the data list is exhausted.

Using the preceding program with semicolons in place of commas gives minimum spacing between successive outputs.

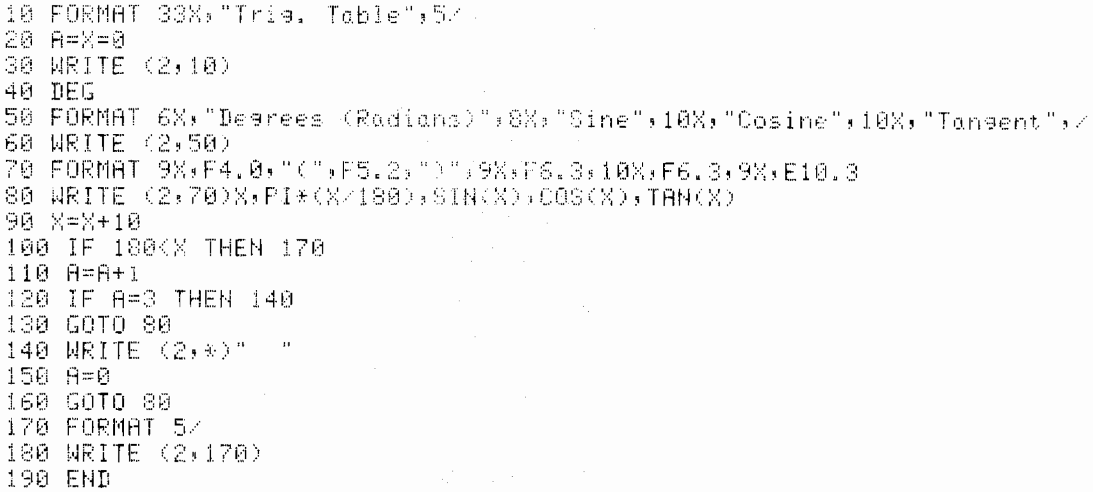


```
10 FIXED 2
20 A=10
30 B=200
40 C=6000
50 D=-25.25
60 PRINT A;B;C;D
70 PRINT "**********************************************************
80 STOP
```



```
 10.00    .200.00    6000.00     -25.25
********************************************************************
```

The following program produces a table of trigonometric values using the formatted WRITE statement described in Chapter 2.



```
10 FORMAT 33X,"Trig. Table",5/
20 A=X=0
30 WRITE (2,10)
40 DEG
50 FORMAT 6X,"Degrees (Radians)",8X,"Sine",10X,"Cosine",10X,"Tangent",/
60 WRITE (2,50)
70 FORMAT 9X,F4.0,"(",F5.2,")",9X,F6.3,10X,F6.3,9X,E10.3
80 WRITE (2,70)X,PI*(X/180),SIN(X),COS(X),TAN(X)
90 X=X+10
100 IF 180<X THEN 170
110 A=A+1
120 IF A=3 THEN 140
130 GOTO 80
140 WRITE (2,*)"  "
150 A=0
160 GOTO 80
170 FORMAT 5/
180 WRITE (2,170)
190 END
```

```
                              Trig. Table




Degrees (Radians)         Sine          Cosine          Tangent

       0( 0.00)          0.000          1.000          0.000E+00
      10( 0.17)          0.174          0.985          1.763E-01
      20( 0.35)          0.342          0.940          3.640E-01

      30( 0.52)          0.500          0.866          5.774E-01
      40( 0.70)          0.643          0.766          8.391E-01
      50( 0.87)          0.766          0.643          1.192E+00

      60( 1.05)          0.866          0.500          1.732E+00
      70( 1.22)          0.940          0.342          2.747E+00
      80( 1.40)          0.985          0.174          5.671E+00

      90( 1.57)          1.000          0.000          9.999E+99
     100( 1.75)          0.985         -0.174         -5.671E+00
     110( 1.92)          0.940         -0.342         -2.747E+00

     120( 2.09)          0.866         -0.500         -1.732E+00
     130( 2.27)          0.766         -0.643         -1.192E+00
     140( 2.44)          0.643         -0.766         -8.391E-01

     150( 2.62)          0.500         -0.866         -5.774E-01
     160( 2.79)          0.342         -0.940         -3.640E-01
     170( 2.97)          0.174         -0.985         -1.763E-01

     180( 3.14)          0.000         -1.000          0.000E+00
```

Here is a brief analysis of the program –

- Lines 10 and 30 print the table title.
- Line 50 prints the column titles.
- Lines 70 and 80 sets the column spacing and prints the columns.

Printer Line Width

The maximum 9866A/B line width is 80 characters. The printer will print only the first 80 characters which are output between linefeed operations. For example if the FORMAT statement in line 10 of the preceding program appeared as this –

```
10 FORMAT 72X,"Trig. Table",5/
20 WRITE (2,10)
```

then the title would be printed incompletely like this –

```
                                                                        Trig. Ta
```

Clearing the Buffer

The printer has an 80 character line buffer that stores characters until a linefeed (LF) is output. Using the RESET key clears the line buffer and readies the 9866A/B for printing operations from the 9831A.

9866A/B Printer Status

The 9866A/B Printer uses bits 4, 5 and 8 of the 98032A Interface status byte. Bits 4 and 5 are the interface ID. (bit four = 0 and bit five = 1), while bit 8 remains at 1 except when the printer is out of paper.

Here's a program segment that checks bit 8 and stops when the printer is out of paper.

```
110 IF STAT2=288 THEN 140
120 DISP "Printer not ready"
130 STOP
140 PRINT A
```

Plotting

Plotting with 9866A/B can be done with characters. In addition, dot patterns can be printed by using the 9866B plot mode.

Character Plotting

Character plotting is accomplished, line-by-line, by spacing to a desired character position on each line and printing the desired character. The following program and printout is an example of character plotting.

```
10 DEG
20 B=0
30 B=B+10
40 PRINT TAB35+INT(SIN(B)*20),"*"
50 IF B>360 THEN 70
60 GOTO 30
70 WRITE (2,*)
80 END
```

Notice that in this program TAB is used with the PRINT statement to output the "asterisk" character to the desired position, as explained in Chapter 8 of the 9831A Operating and Programming Manual.

```
                 *
                    *
                        *
                           *
                              *
                                 *
                                    *
                                      *
                                       *
                                        *
                                        *
                                        *
                                       *
                                      *
                                    *
                                 *
                              *
                           *
                        *
                    *
                 *
              *
           *
        *
     *
   *
  *
 *
*
 *
  *
   *
     *
        *
           *
              *
                 *
```

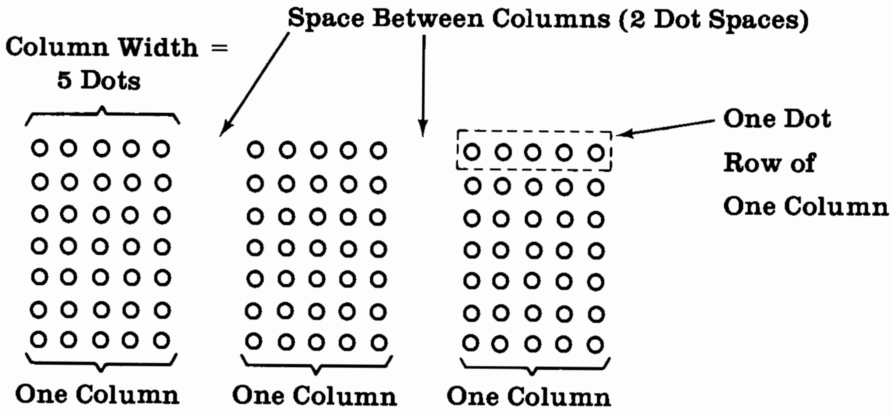
9866B Plot Mode

The plot mode enables you to use your printer as an incremental plotter. Horizontal and vertical bar charts and mathematical data graphs are examples of the plots that can be generated using this mode. Plotting is accomplished by printing a specified dot pattern in a specified column for each point on the plot.

Printing Dot Patterns

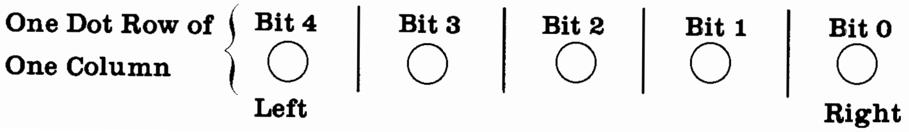
The plot mode allows any dot pattern listed in Table 1 (at the end of this chapter) to be printed in any desired column on the printer. The dot patterns are printed one dot-row at a time. After a row is printed in the plot mode, the printer is reset to the normal print mode by a linefeed (LF). Thus the plot mode must be set before each dot row is printed. Remember that the printer's line width is 80 columns, and each column is 5 dots wide.

See the figure below.


Space Between Columns (2 Dot Spaces)
Column Width = 5 Dots
One Dot Row of One Column
One Column
One Column
One Column

The ASCII character DC_1 (decimal 17) sets the printer to the plot mode. A linefeed character (decimal 10) causes the dot row to be printed and cancels the plot mode. The plot mode must then be reset to continue plotting.

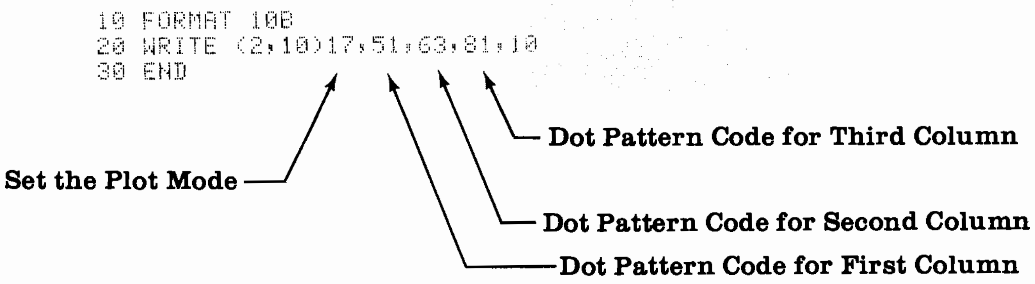
When in the plot mode, the printer will interpret each character received as a representation of a five dot pattern as shown below. Only the character's lower 5 bits are used by the printer (bits 5-7 not used).



A dot is printed in a dot row when the corresponding bit is a 1.

Table 1 (at the end of this chapter) lists the plot-mode dot-patterns and the characters and codes that produce them. Note that since only the last 5 bits of a character are used to make a pattern, more than one character or code will produce the same pattern.

Here is an example that prints one dot row in three columns.



```
10 FORMAT 10B
20 WRITE (2,10)17,51,63,81,10
30 END
```

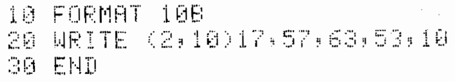
Set the Plot Mode
Dot Pattern Code for Third Column
Dot Pattern Code for Second Column
Dot Pattern Code for First Column

The linefeed (10) cancels the plot mode. The result is a row of dots in the first three columns, printed with the dot patterns corresponding to the decimal codes.

●○○●●	●●●●●	●○○○● ← ONE DOT ROW
Column 1 Dot Pattern for Decimal 51	**Column 2 Dot Pattern for Decimal 63**	**Column 3 Dot Pattern for Decimal 81**

Note that for any particular dot pattern, any one of the decimal codes or any one of the ASCII characters for that pattern listed in Table 1 can be used to print that pattern.

Here is how the second dot-row of the preceding example might be printed.

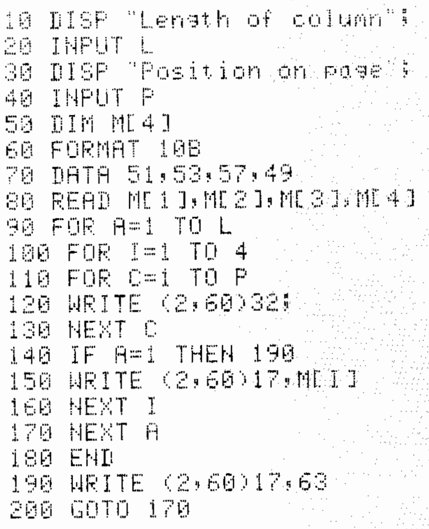


```
10 FORMAT 10B
20 WRITE (2,10)17,57,63,53,10
30 END
```

```
●○○●●     ●●●●●     ●○○○●
●●○○●     ●●●●●     ●○●○●
```

Second Dot Row Printed

Here is an example of plotting one column as might be used in a veritical bar chart.



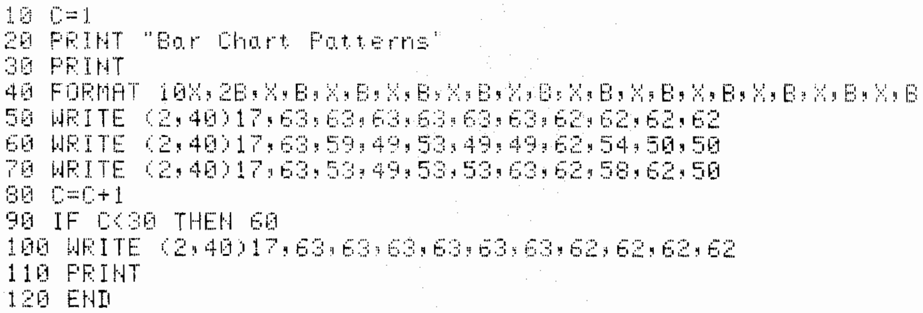
```
10 DISP "Length of column";
20 INPUT L
30 DISP "Position on page";
40 INPUT P
50 DIM M[4]
60 FORMAT 10B
70 DATA 51,53,57,49
80 READ M[1],M[2],M[3],M[4]
90 FOR A=1 TO L
100 FOR I=1 TO 4
110 FOR C=1 TO P
120 WRITE (2,60)32;
130 NEXT C
140 IF A=1 THEN 190
150 WRITE (2,60)17,M[I]
160 NEXT I
170 NEXT A
180 END
190 WRITE (2,60)17,63
200 GOTO 170
```

Program Analysis –

- Line 10 – The variable L = the length of the column.
- Line 30 – The variable P = the horizontal spacing of the column on the page.
- Line 80 – An array is used to hold the pattern for the column.
- Lines 90 thru 190 – These lines are responsible for printing the column, the specified length and horizontal spacing.

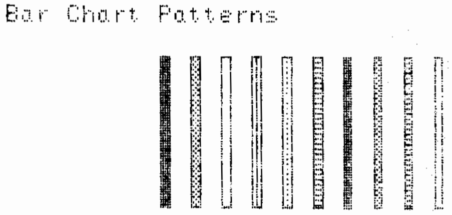
Here is an example printout when L = 20 and P = 10.

The following program prints example patterns that could be used in vertical bar charts.



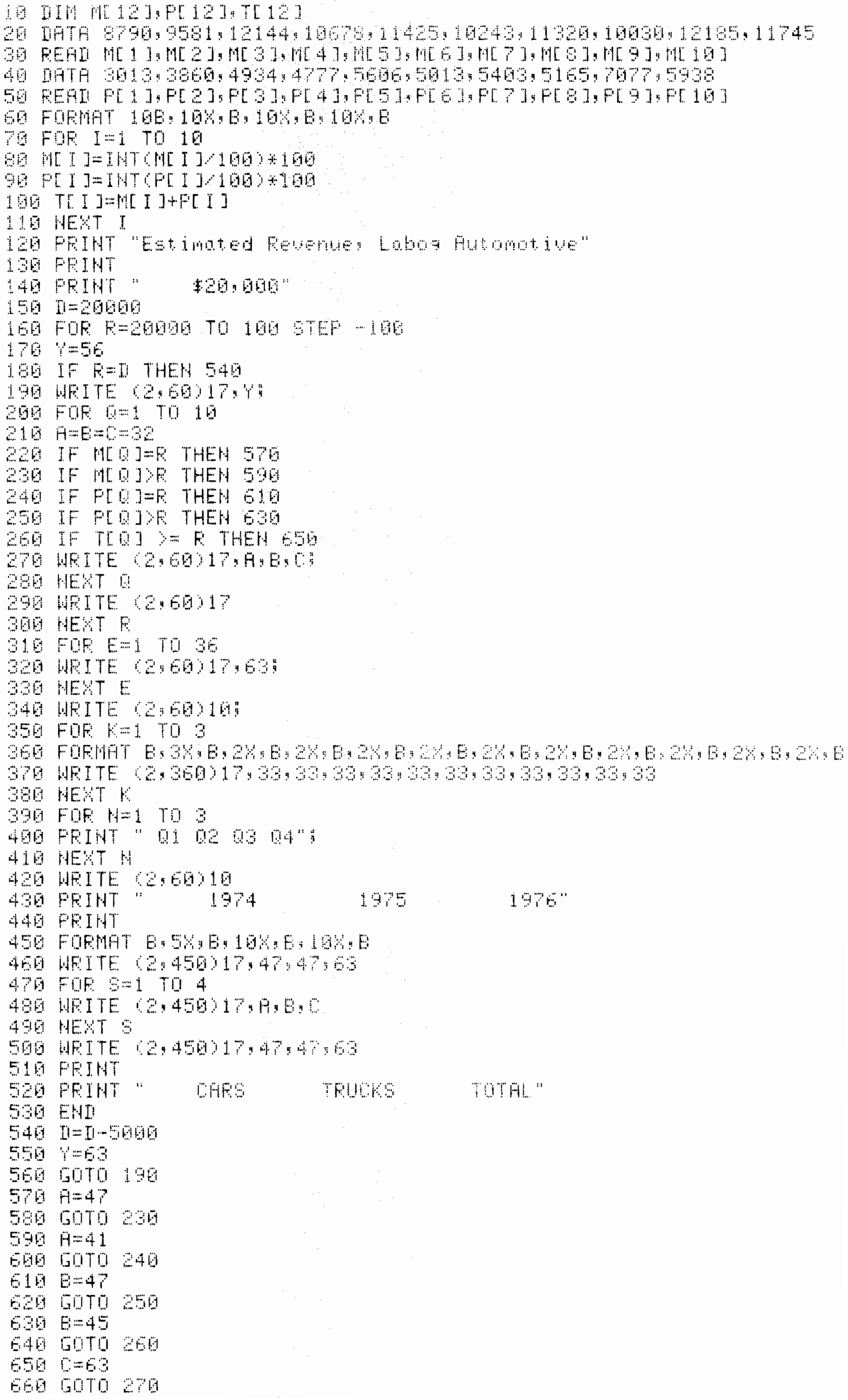
```
10 C=1
20 PRINT "Bar Chart Patterns"
30 PRINT
40 FORMAT 10X,2B,X,B,X,B,X,B,X,B,X,B,X,B,X,B,X,B,X,B,X,B,X,B
50 WRITE (2,40)17,63,63,63,63,63,63,62,62,62,62
60 WRITE (2,40)17,63,59,49,53,49,49,62,54,50,50
70 WRITE (2,40)17,63,53,49,53,53,63,62,58,62,50
80 C=C+1
90 IF C<30 THEN 60
100 WRITE (2,40)17,63,63,63,63,63,63,62,62,62,62
110 PRINT
120 END
```

Result –



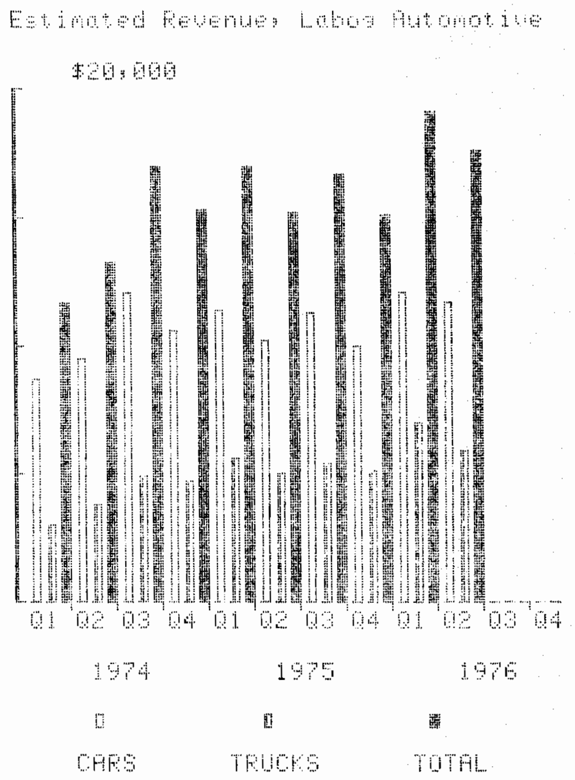
Bar Chart Patterns

Here is a program that is used to plot a bar chart –



```
10 DIM M[12],P[12],T[12]
20 DATA 8790,9581,12144,10678,11425,10243,11320,10030,12185,11745
30 READ M[1],M[2],M[3],M[4],M[5],M[6],M[7],M[8],M[9],M[10]
40 DATA 3013,3860,4934,4777,5606,5013,5403,5165,7077,5938
50 READ P[1],P[2],P[3],P[4],P[5],P[6],P[7],P[8],P[9],P[10]
60 FORMAT 10B,10X,B,10X,B,10X,B
70 FOR I=1 TO 10
80 M[I]=INT(M[I]/100)*100
90 P[I]=INT(P[I]/100)*100
100 T[I]=M[I]+P[I]
110 NEXT I
120 PRINT "Estimated Revenue, Labos Automotive"
130 PRINT
140 PRINT "     $20,000"
150 D=20000
160 FOR R=20000 TO 100 STEP -100
170 Y=56
180 IF R=D THEN 540
190 WRITE (2,60)17,Y;
200 FOR Q=1 TO 10
210 A=B=C=32
220 IF M[Q]=R THEN 570
230 IF M[Q]>R THEN 590
240 IF P[Q]=R THEN 610
250 IF P[Q]>R THEN 630
260 IF T[Q] >= R THEN 650
270 WRITE (2,60)17,A,B,C;
280 NEXT Q
290 WRITE (2,60)17
300 NEXT R
310 FOR E=1 TO 36
320 WRITE (2,60)17,63;
330 NEXT E
340 WRITE (2,60)10;
350 FOR K=1 TO 3
360 FORMAT B,3X,B,2X,B,2X,B,2X,B,2X,B,2X,B,2X,B,2X,B,2X,B,2X,B,2X,B
370 WRITE (2,360)17,33,33,33,33,33,33,33,33,33,33,33
380 NEXT K
390 FOR N=1 TO 3
400 PRINT " Q1 Q2 Q3 Q4";
410 NEXT N
420 WRITE (2,60)10
430 PRINT "     1974          1975          1976"
440 PRINT
450 FORMAT B,5X,B,10X,B,10X,B
460 WRITE (2,450)17,47,47,63
470 FOR S=1 TO 4
480 WRITE (2,450)17,A,B,C
490 NEXT S
500 WRITE (2,450)17,47,47,63
510 PRINT
520 PRINT "     CARS        TRUCKS        TOTAL"
530 END
540 D=D-5000
550 Y=63
560 GOTO 190
570 A=47
580 GOTO 230
590 A=41
600 GOTO 240
610 B=47
620 GOTO 250
630 B=45
640 GOTO 260
650 C=63
660 GOTO 270
```

Result —



Here is a brief program analysis –

- Lines 10 thru 50 – Data values for the plot are stored in arrays.
- Lines 80 thru 90 – Data values are rounded.
- Lines 120 thru 140 – Text printed before chart.
- Lines 220 thru 260 – Each data item is checked as the chart is printed to see if that data bar should be printed.
- Lines 310 thru 380 – Base line of the plot is printed.
- Lines 400 thru 520 – Text after chart is printed.

Table 1: 9866B Plot Mode Characters

Dot Pattern	Decimal Codes	ASCII Characters		
○ ○ ○ ○ ○	32, 64, 96	space	@	
○ ○ ○ ○ ●	33, 65, 97	!	A	a
○ ○ ○ ● ○	34, 66, 98	"	B	b
○ ○ ○ ● ●	35, 67, 99	#	C	c
○ ○ ● ○ ○	36, 68, 100	$	D	d
○ ○ ● ○ ●	37, 69, 101	%	E	e
○ ○ ● ● ○	38, 70, 102	&	F	f
○ ○ ● ● ●	39, 71, 103	'	G	g
○ ● ○ ○ ○	40, 72, 104	(	H	h
○ ● ○ ○ ●	41, 73, 105	)	I	i
○ ● ○ ● ○	42, 74, 106	*	J	j
○ ● ○ ● ●	43, 75, 107	+	K	k
○ ● ● ○ ○	44, 76, 108	,	L	l
○ ● ● ○ ●	45, 77, 109	–	M	m
○ ● ● ● ○	46, 78, 110	.	N	n
○ ● ● ● ●	47, 79, 111	/	O	o
● ○ ○ ○ ○	48, 80, 112	Ø	P	p
● ○ ○ ○ ●	49, 81, 113	1	Q	q
● ○ ○ ● ○	50, 82, 114	2	R	r
● ○ ○ ● ●	51, 83, 115	3	S	s
● ○ ● ○ ○	52, 84, 116	4	T	t
● ○ ● ○ ●	53, 85, 117	5	U	u
● ○ ● ● ○	54, 86, 118	6	V	v
● ○ ● ● ●	55, 87, 119	7	W	w
● ● ○ ○ ○	56, 88, 120	8	X	x
● ● ○ ○ ●	57, 89, 121	9	Y	y
● ● ○ ● ○	58, 90, 122	:	Z	z
● ● ○ ● ●	59, 91, 123	;	[	π
● ● ● ○ ○	60, 92, 124	<	√	'
● ● ● ○ ●	61, 93, 125	=	]	→
● ● ● ● ○	62, 94, 126	>	↑	∿
● ● ● ● ●	63, 95	?	–	
Set Plot Mode	17	DC_1		
End of Plot Row	10	LF		

Chapter 6

HP 9869A Hopper Card Reader

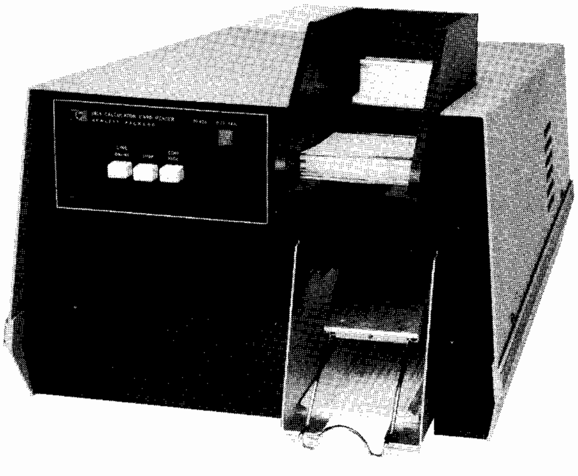
Introduction

This chapter describes the input/output operations used to control the HP 9869A Card Reader: the ENTER, WRITE and FORMAT statements and the RBYTE (read byte) function. These operations were introduced in Chapter 2.

The HP 9869A Hopper Card Reader is connected to the 9831A via the 98032A Option 169 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90044). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

The HP 9869A is a hopper-fed card reader. Hollerith-coded, standard data processing cards can be read at speeds up to 300 cards per minute. The 9869A reads both marked and punched cards with equal reliability. Punches and marks can be intermixed on the same card. The 9869A can also be set, by a back panel switch, to read data processing cards with or without clock marks. The internal card reader buffers allow the multiple transfer of data from a card to the computer under program control. An optional select hopper provides rapid-sort capability for data file search and reduction.



9869A Card Reader

Throughout this manual the term "print format" refers to the same format used with the PRINT statement. In the output statements the asterisk (*) designates the print format whereas with input statements the (*) designates free-field format.

The Data Buffer

As information on each data card is read, the card reader stores it in a temporary memory area called a buffer. When the card reader is switched on or RESET is pressed, the buffer is cleared. The buffer is loaded as cards are picked and then emptied by using ENTER statements. The buffer can hold 160 card-columns of data.

Operating Modes

The card reader has two sets of operating modes. The Demand and Continuous modes determine how the cards are picked; the Normal and Image modes determine how each card-column of data is read.

Each mode is set by sending the card reader an appropriate ASCII character and remains set until either the opposite mode is set or the card reader is reset. The mode control characters are listed below. Operation with each mode is described in the following pages.

Mode Control Characters

Mode	Character	Operation
Continuous	C	Cards are picked automatically, as needed.
Demand mode	D	Cards are picked one at a time.
Normal mode	N	Data is read, column by column, in Holerith code.
Image mode	I	Data is read in ½ column, 4-bit binary form.

Card Reader Functions

Functions such as Bell, Reject and Reset are controlled using ASCII control characters. A complete list is below. An example use of each function is shown in the following pages.

Function Control Characters

Function	Character	Operation
Stop	S	Stops (resets) the card reader and cancels the operating mode(s).
Abort	A	Clears the buffer (Demand mode only).
Reject/Select	J	Outputs the currently picked card to the reject hopper (9869A Option 002 and Demand mode only).
Bell	B	Rings the card reader's bell (9869A Option 004).
Retransmit	T	Holds the data from the last card for retransmission (Demand mode only).

The WRITE Statement

The WRITE statement is used to set operating modes and to activate functions on the card reader. A simplified syntax is shown here –

WRITE (select code ; *) "control character"

This syntax uses the print format to send control characters as free text.

Once an operating mode is set, data cards are picked and the data is stored in the card reader buffer. Then ENTER statements can be used to transfer the data to the 9831A.

The ENTER Statement

ENTER (select code ; * or line number) variable$_1$ [; variable$_2$...]

The ENTER statement enables the 9831A to receive data from the card reader. This empties the buffer automatically. If incoming data is not in ASCII code, conversion to ASCII code may be included in the ENTER statement using an optional conversion table parameter. Also, optional FOR-NEXT parameters may be used to input multiple data from one data processing card into an array. These optional parameters are described more completely in Chapter 2.

As the ENTER statement is executed, the first data item in the buffer is stored in the first variable specified; the second item is stored in the second variable; etc. In general, the number of variables should correspond to the number of data items to be read from each card. Each numeric data item can consist of the digits 0 through 9, plus or minus sign, a decimal point and an "E" (exponential) character. All other characters are treated as input delimiters. The data item itself can assume the same form as any number entered from the keyboard.

The line number can be used to reference a FORMAT statement. The * is used in place of a line number to specify the free-field format.

Free-Field Format

The free-field format allows reading numeric data in virtually any form, provided that each item is followed by at least one delimiter (any non-numeric character). For each variable in the ENTER list, the 9831A ignores all delimiters until a numeric character is read. Then, after reading the data item, reading any non-numeric character terminates and stores the data item. After the last variable in the list has been filled, reading a LF (linefeed) terminates the read operation. The card reader automatically places a LF in the buffer directly after the last character read from each card.

When using free-field format, each data item must be separated by a non-numeric character.

FORMAT Statements

FORMAT specification$_1$ [, specification$_2$, ...]

Use of FORMAT statements provides flexible and complete control of WRITE and ENTER statements. A FORMAT statement is referenced by replacing the * in the ENTER statement with a line number to indicate the location of the FORMAT statement.

The list of numeric and edit specifications determine the exact characters to be read. Numeric specs are used to determine which characters are input from a data input string, and in what form the data will appear. A general numeric input spec syntax is –

[r] Fw . d

- r is the number of consecutive times the spec is to be used (if r is 1 it may be omitted).
- w is the width of the data field to be read.
- d is the number of decimal places. If a decimal point is in the number, d is ignored. r, w and d must be interger constants.

A numeric spec like F10.2 calls for reading ten numeric characters; all non-numerics which precede a numeric are counted but not entered. If an "E" is read, exponential format is assumed.

Most examples in this chapter use FORMAT statements, rather than the free-field format. For more information on input format specs, refer to Chapter 2.

Pick Fails

The card reader tries to pick a card when instructed to do so. If the card reader does not sense that a card has been picked, it tries to pick two more times. If it fails to pick a card successfully three times, it will stop, set the status line to "1" and light the yellow Pick Fail light. A pick fail also cancels the Continuous, Demand and Image modes. The Image mode and the status line are explained at the end of this chapter.

To clear a pick fail, press STOP, remove the defective card and then execute CONT to resume program operation.

Continuous Mode Operation

```
WRITE (3,*) "C"
```

The Continuous mode allows the card reader to pick cards without further instruction. This statement causes the card reader to pick data cards continuously until the buffer is full and then automatically pick cards, as required, until either the input hopper is empty or the output hopper is full.

The following program causes the card reader to pick cards continuously, input the three data items on each card to the 9831A, and print the data in the same order as it is read. It's assumed that each data item is separated by a comma (free-field format is used) and that select code 3 is set. When in the continuous mode, the card reader picks each new card after half of the data in the buffer is output to the 9831A.

```
10 WRITE (3,*)"C"
20 ENTER (3,*)A,B,C
30 PRINT A,B,C
40 GOTO 20
50 END
```

Notice that once the card reader has been instructed to pick cards in the Continuous mode, there is no need to instruct it to pick individual cards again. Re-executing the ENTER statement empties the buffer, causing successive cards to be picked.

For reading data, the Continuous mode is faster than the Demand mode, but the Reject, Abort and Retransmit functions can only be executed in the Demand mode.

6

Since the previous program doesn't count cards, it will halt and ERROR 72 will appear when the card reader is finished reading cards (i.e., when the input hopper is empty or the output hopper is full). This error can be avoided by using a dummy card for the last card as shown in the next program.

This program checks for a dummy card on which the first data item is 1E99 –

```
10 WRITE (3,*)"C"
20 ENTER (3,*)A,B,C
30 IF A=1E+99 THEN 60
40 PRINT A,B,C
50 GOTO 20
60 END
```

The STOP Command

```
WRITE (3,*)"S"
```

This statement halts the card reader, clears the buffer, and terminates the Continuous mode –

Before the motor shuts off, however, the buffer must be emptied using ENTER statements.

Demand Mode Operations

```
WRITE (3,*) "D"
```

The Demand mode is used when cards are to be picked one at a time. This statement causes the card reader to pick a single data card (unless the buffer is full, the input card hopper is empty, or the output card hopper is full). The statement must be executed each time a card is to be picked. Also, the buffer must be empty before the Demand mode can be set.

For example, here is a program to read and print ten cards from the deck of HP 9869A Test Cards furnished with the card reader. Each card has the same five data items. One of the test cards is shown on the following page.

```
10 FORMAT 4F15.0,F20.0
20 FOR I=1 TO 20
30 WRITE (3,*)"D"
40 ENTER (3,10)A,B,C,D,E
50 PRINT A,B,C,D,E
60 NEXT I
70 END
```

```
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
1.23456789     98.76543211     1         -1         -100
```

A 9869A Test Card

The Abort Command (Demand Mode Only)

```
WRITE (3,*) "A"
```

The Abort function clears the first half of the card reader buffer, permitting another card to be read. The aborted data is held in the second half of the buffer, and can be re-used until another card is read; see the "Retransmit Function", explained later.

This program reads three data items from each card. If the first item is positive, the data is printed. If the item is negative, the data is aborted. Line 50 checks for a dummy card, as described previously.

```
10 FORMAT 3F10.0
20 WRITE (3,*)"D"
30 ENTER (3,10)A,B,C
40 IF A<0 THEN 80
50 IF A=1E+99 THEN 100
60 PRINT A,B,C
70 GOTO 20
80 WRITE (3,*)"A"
90 GOTO 20
100 END
```

The Reject/Select Function (Demand Mode Only)

```
WRITE (3,*)"J"
```

This function is used to sort data cards and can be used only when the card reader has Option 002, (reject hopper). The buffer must be emptied before the function will be executed.

For example, the following program causes the card reader to sort all data cards on which the first data item is less than or equal to zero. (It's assumed that each card contains more than one data item.) Line 50 checks for a dummy card. When that last card is seen, line 110 prints the number of rejected cards.

```
10 FORMAT F8.0
20 WRITE (3,*)"D"
30 ENTER (3,10)A
40 IF A <= 0 THEN 70
50 IF A=1E+99 THEN 100
60 GOTO 20
70 WRITE (3,*)"J"
80 J=J+1
90 GOTO 20
100 DISP J"CARDS REJECTED"
110 END
```

The Retransmit Function (Demand Mode Only)

```
WRITE (3,*) "T"
```

The Retransmit function enables previously-read or aborted data to be reused more than one time. This function transfers data back to the first half of the buffer, where it remains until it is either reread or another card is read. This function can be repeated as often as needed.

This program checks the first item on each card; if the item is negative, the card is rejected. But if the item is positive, line 60 recalls the data to the buffer so that lines 70 and 80 can read and print it.

```
10 FORMAT F15.0
20 FORMAT 5F10.0
30 WRITE (3,*)"D"
40 ENTER (3,10)A
50 IF A <= 0 THEN 100
60 WRITE (3,*)"T"
70 ENTER (3,20)A,B,C,D,E
80 PRINT A,B,C,D,E
90 GOTO 30
100 WRITE (3,*)"J"
110 GOTO 30
120 END
```

NOTE

Since the last data card cannot be rejected under any circumstances, it's suggested that a blank card be added to the end of the card deck. This permits the last data card to be rejected, if desired.

The Stop Command

```
WRITE (3,*) "AS"
```

The STOP command clears the buffer, halts the card reader, and terminates the Demand mode.

Additional Operations

The following operations can be used when either the Continuous or Demand mode is set.

The Bell Function

```
WRITE (3,*) "B"
```

The Bell function is used to signal the operator that a predetermined system condition is reached. This function can be executed only if the card reader has Option 004, (bell).

This program causes the card reader to ring its bell (actually, it's a beep) after each group of 10 data cards –

```
10 FOR I=1 TO 10
20 WRITE (3,*)"D"
30 ENTER (3,*)A,B,C
40 PRINT A,B,C
50 NEXT I
60 WRITE (3,*)"B"
70 GOTO 10
80 END
```

The RBYTE Function

RBYTE select code

Once the card reader is set to an operation mode, the RBYTE function can be used to read each individual column on each data card. This feature is useful when reading cards on which the marking format may or may not be in standard 128-character Hollerith code (the common format for keypunched and marked cards). When RBYTE is executed, the information on each column is converted to a decimal code that's equivalent to the ASCII-coded character. Then the decimal code is used in the statement. A table of Hollerith-ASCII characters is in the HP 9869A Peripheral Manual. Here's a short program which reads and prints a code for each column on a data card. The program halts after reading the LF (decimal 10) which follows the last column.

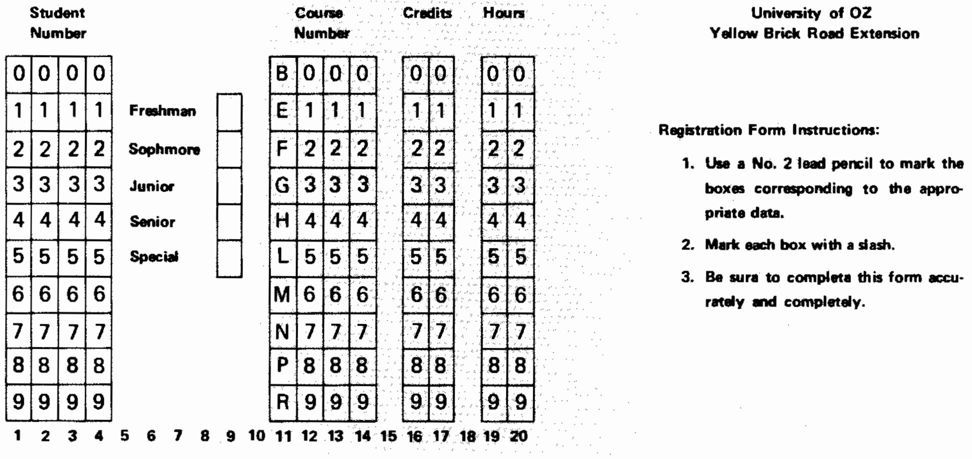
```
10 WRITE (3,*)"D"
20 A=RBYTE3
30 PRINT A
40 IF A=10 THEN 60
50 GOTO 10
60 END
```

The beginning and end of the printout obtained by reading one of the 9869A Test Cards is shown below.

```
49
46
50
51
52
53
54
55
56
57
32
32
32
32
32
32

32
32
32
45
48
48
10
```

As another example, suppose that the registrar of a small university wishes to compare the average work load per semester of both part-time and full-time students. To solve this problem, this 40-column class registration card was designed –



Student Number						Course Number				Credits		Hours	
0	0	0	0			B	0	0	0	0	0	0	0
1	1	1	1	Freshman	□	E	1	1	1	1	1	1	1
2	2	2	2	Sophmore	□	F	2	2	2	2	2	2	2
3	3	3	3	Junior	□	G	3	3	3	3	3	3	3
4	4	4	4	Senior	□	H	4	4	4	4	4	4	4
5	5	5	5	Special	□	L	5	5	5	5	5	5	5
6	6	6	6			M	6	6	6	6	6	6	6
7	7	7	7			N	7	7	7	7	7	7	7
8	8	8	8			P	8	8	8	8	8	8	8
9	9	9	9			R	9	9	9	9	9	9	9

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

University of OZ
Yellow Brick Road Extension

Registration Form Instructions:

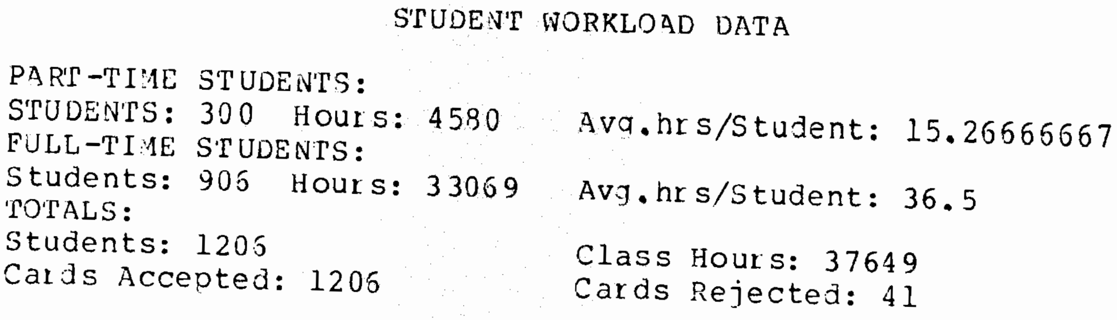
1. Use a No. 2 lead pencil to mark the boxes corresponding to the appropriate data.
2. Mark each box with a slash.
3. Be sure to complete this form accurately and completely.

Then this program was written. An analysis of the program follows.

```
10 DIM A[20]
20 C1=C2=F1=F2=P1=P2=0
30 B=0
40 WRITE (3,*)"D"
50 FOR I=1 TO 20
60 A[I]=RBYTE(3)
70 NEXT I
80 IF A[I]=84 THEN 230
90 C1=C1+1
100 B=(10*(A[19]-48)+A[20]-48)
110 IF B<40 AND B>0 THEN 140
120 WRITE (3,*)"J"
130 C2=C2+1
140 GOTO 30
150 IF B <= 12 THEN 190
160 F1=F1+B
170 F2=F2+1
180 GOTO 210
190 P1=P1+B
200 P2=P2+1
210 WRITE (3,*)"A"
220 GOTO 30
230 WRITE (3,*)"AS"
240 PRINT
250 PRINT TAB20"STUDENT WORKLOAD DATA"
260 PRINT
270 PRINT "PART-TIME STUDENTS:"
280 PRINT "STUDENTS:";P1;"Hours:";P2;"Avg.hrs/Student:";P2/P1
290 PRINT "FULL-TIME STUDENTS:"
300 PRINT "Students:";F1;"Hours:";F2;"Avg.hrs/Student:";F2/F1
310 PRINT "TOTALS:"
320 PRINT "Students:";P1+F1,"Class Hours:";P2+F2
330 PRINT "Cards Accepted:";C1,"Cards Rejected:";C2
340 PRINT
350 END
```

- Lines 10 and 20 initialize the working variables.
- Line 40 sets the Demand mode (the first card is picked).
- Lines 50 thru 70 transfer the first 20 columns of data to the 9831A.
- Line 80 checks the data from column 1. If rows 0, 1 and 2 are marked (this corresponds to decimal code 84), the card is a "dummy" card which is placed at the end of the card deck.
- Line 100 converts the decimal codes from column 19 and 20 (stored in A [19] and A [20] to decimal numbers Hours).
- Line 110 checks to make sure that the number of hours is greater than 0 but less than 40. If not:
- Lines 120 thru 140 reject the card and return the program to pick the next record.
- Lines 160 thru 200 accumulate full-time or part-time data from each card.
- Lines 210 and 220 clears the buffer and returns the program to pick the next card.

The remainder of the program reduces the data gathered and produces the following printout (sample data is shown) –



```
                    STUDENT WORKLOAD DATA
PART-TIME STUDENTS:
STUDENTS: 300  Hours: 4580     Avg.hrs/Student: 15.26666667
FULL-TIME STUDENTS:
Students: 906  Hours: 33069    Avg.hrs/Student: 36.5
TOTALS:
Students: 1206                 Class Hours: 37649
Cards Accepted: 1206           Cards Rejected: 41
```

The Image Mode

By setting the Image mode, the information from each card column can be transferred to the 9831A in two binary-coded 8-bit bytes, rather than in one decimal code as previously described. This is useful when more than one row per card-column is expected to contain information. The Image mode can be used only with the RBYTE function and can be specified when either Continuous or Demand mode is set.

This statement specifies the Image mode –

```
WRITE (3,*) "I"
```

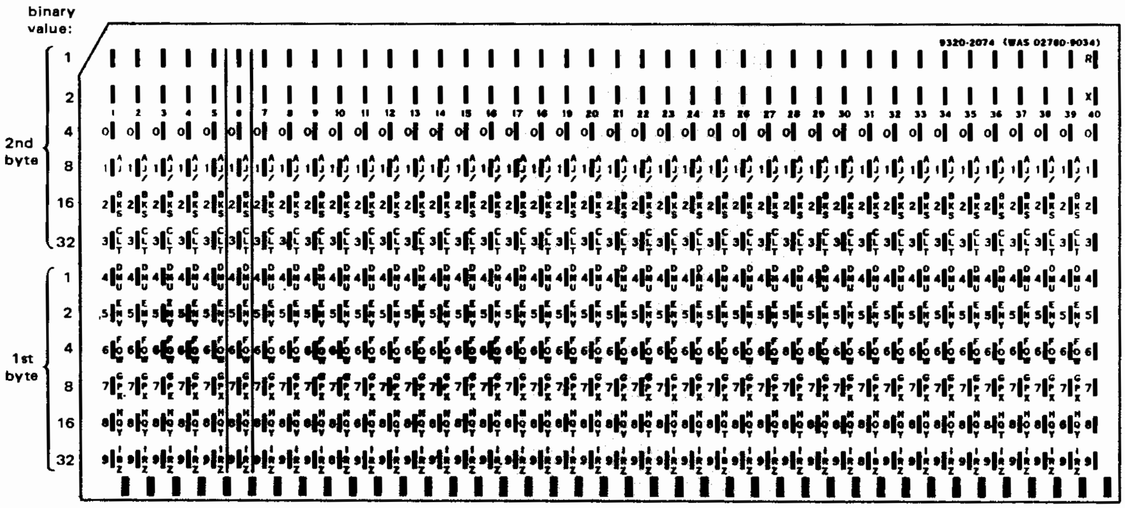
This statement cancels the Image mode and resets the Normal mode –

```
WRITE (3,*) "N"
```

The Image mode is also cancelled by pressing RESET or by executing the Stop Command (see the following programs).

For example, use this program to read the next data card –

```
10 WRITE (3,*)"DI"
20 FOR I=1 TO 16
30 PRINT "Column:"I,"1st Byte:"RBYTE3,"2nd Byte:"RBYTE3
40 NEXT I
50 WRITE (3,*)"AS"
60 END
```



```
Column: 1     1st Byte: 1     2nd Byte: 2
Column: 2     1st Byte: 3     2nd Byte: 4
Column: 3     1st Byte: 5     2nd Byte: 6
Column: 4     1st Byte: 7     2nd Byte: 8
Column: 13    1st Byte: 25    2nd Byte: 26
Column: 14    1st Byte: 27    2nd Byte: 28
```

Compare the printout for each card-column with the marks on the card. Notice that after the Image mode is set (line 10), the RBYTE functions input the decimal equivalent of the binary information on each half-column.

Here's a program that may be used to reject any card which has more than one mark per column on the first 20 columns (the registrar mentioned earlier could use this program to sort out poorly marked cards). An analysis of the program follows.

```
10 A=B=0
20 WRITE (3,*)"DI"
30 FOR I=1 TO 20
40 A=RBYTE(3)
50 IF A=28 THEN 140
60 IF A#0 AND A#1 AND A#2 AND A#4 AND A#8 AND A#16
                                  AND A#32 THEN 90
70 B=RBYTE(3)
80 IF B#1 AND B#2 AND B#4 AND B#8 AND B#16 AND B#32 THEN 90
90 IF A=0 AND B>0 THEN 110
100 IF A>0 AND B=0 THEN 110
110 WRITE (3,*)"J"
120 J=J+1
130 GOTO 10
140 NEXT I
150 WRITE (3,*)"AD"
160 GOTO 10
170 WRITE (3,*)"AS"
180 DISP J"CARDS REJECTED";
190 END
```

- Line 20 sets the Demand and Image modes.
- Lines 40 thru 60 inputs the first half-column of data and determines if more than one row is marked. Line 50 checks for the last card, on which rows 0, 1 and 2 are marked in the first column.
- Lines 70 and 80 inputs the second half-column of data and determines if more than one row is marked in that half-column.
- Lines 90 thru 110 rejects the card if more than one mark, or no mark, is found in the entire column.
- Line 150 picks another card after 20 columns of data from the current card have been read.
- Line 170 clears the buffer, halts the card reader, and cancels the Image mode.

Card Reader Status Line

The card reader status line is connected to the interface and can be monitored using the STAT function. The status line corresponds to interface status bit 8 (decimal 256) and is 0 during any of these card reader conditions –

- A read lamp fails.
- The input hopper is full.
- The output hopper is full.
- A pick fail occurs.

The STAT function and interface status bits are explained in Chapter 2.

Here is a sequence which checks the status line before picking each new card. If the line is 0, "Check Card Reader" is displayed and the program halts.

```
50 IF STAT3>255 THEN 90
60 DISP "CHECK CARD READER";
70 STOP
80 GOTO 50
90 WRITE (3,*)"D"
```

The PTAPE Command

PTAPE select code

or

PTA select code

Programs that are marked or punched on the proper cards can be entered into the 9831A from the 9869A. When the PTAPE statement is executed with the card reader, the CONTINUOUS PICK button on the front of the reader must be pressed to pick the first card. STOP must then be pressed after the program is entered to terminate the PTAPE command.

Chapter 7

The HP 9871A Printer

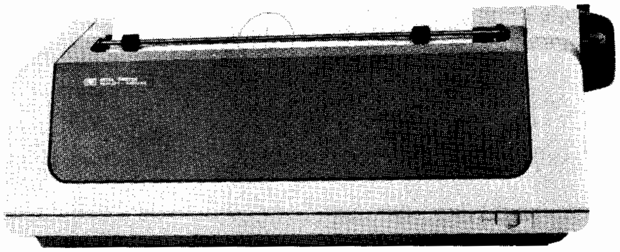
Introduction

This chapter describes the general and unique operations used to control the HP 9871 Printer. A brief explanation and syntax of the controlling statements is given. Chapter 2 of this manual has been devoted to introduction and explanation of the controlling statement. A complete list of the error codes is in the Appendix.

The HP 9871A Printer is connected to the HP 9831A via the 98032A Option 171 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90045). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

The HP 9871A is a full-character serial impact printer for use with the 9831A. The platen accommodates paper up to 381 mm (15 in.) wide. The 9871A prints a standard 132 columns at 10 characters per inch; however, character and line spacing can be defined to increase or decrease the number of characters per line. The 96-character interchangeable print disk provides full-character impact on the fixed carriage. Bidirectional motion of the platen and print mechanism provide plotting capabilities for charts and graphs.



9871A Printer

This chapter assumes that you are familiar with the operation of the 9831A and the 9871A Printer, as explained in their operating manuals.

Here are the statements and functions used to control the 9871A Printer.

STDPR   Specifies the "standard printer" for PRINT, LIST, TLIST, TRACE and print-all operations.

PRINT   Prints text and numbers on the standard printer.

TAB and SPA   Outputs a specified block of spaces to position output items with PRINT, WRITE and DISP statements

LIN   Outputs a specified number of linefeeds; used with PRINT and WRITE statements.

WRITE   Outputs text and numbers to any output device. Items are output in either the PRINT format or in the form specified by a referenced FORMAT statement.

FORMAT   Specifies the output form for variables and edit specifications for controlling an output printer.

LIST   Prints program lines on the standard printer.

TLIST   Prints information about each tape file on the standard printer.

The following printer conditions are specified at "power up" on the 9871A.

- The carrier is moved to the extreme left and the character disk hub is rotated into a mechanical stop. This synchronizes the internal logic with the mechanical position. The carrier is then moved to the first print position (column 1).
- All tabs are cleared.
- Horizontal spacing is set to 10 characters per inch. Variable horizontal spacing is disabled.
- Vertical spacing is set to 6 lines per inch.
- The view-delay function is disabled.
- Top of form is assumed at the current platen position.
- Left margin is assumed at column 1.

- Text length is assumed to be 11 inches.
- Form length is assumed to be 11 inches.
- The standard "shift-in" character-set is assumed.
- Text width is assumed to be 13.2 inches (right margin).
- All character replacements are cleared.

Data Output

Alphanumeric data is usually output using WRITE statements. Either the PRINT format or FORMAT statements can be referenced to govern the data form and delimiters. If individual characters or values are to be output without character fields, use the WBYTE function.

The 9871A Printer can be assigned as the standard printer for the 9831A.

STDPR select code

Successive PRINT, LIST, TLIST, TRACE and print-all operations are then automatically output to the 9871A.

The WRITE Statement

WRITE (select code , * or line number) [any combination of text and expressions]

Text and numeric data are usually printed using WRITE statements. Text items must be enclosed by quotes and are printed exactly as they appear in the quotes. Numeric data items (digits, sign and decimal point) are printed in a print format unless a format number is specified in the WRITE statement and a similarly numbered format statement has been executed. Refer to WRITE and FORMAT statements in Chapter 2 for more information about these operations.

For example, this program sequence prints six numbers with text using the PRINT format.

```
10 A=255
20 FIXED 3
30 WRITE (2,*)A,A/4,A/6,A/3,"Print Format",A/15
40 END
```

Here is the printout –

```
255.000      63.750      42.500      31.875      Print Format
17.000
```

The next program sequence prints six numbers with text using WRITE with a FORMAT statement.

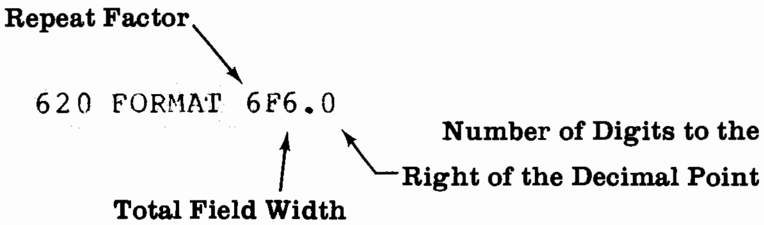
```
610 A=255
620 FORMAT 6F6.0
630 WRITE (2,620)A,A/4,A/6,A/8," Format Statement",A/15
640 END
```

Here is the printout –

```
255     64     43     32 Format Statement     17
```

The WRITE statement references the FORMAT statement in line 620. Formatted WRITE statement are covered in Chapter 2.

Here is a brief explanation of the FORMAT statement used –

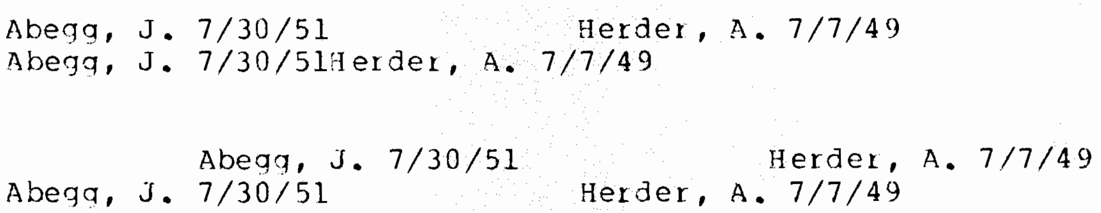


String Variables

Alphanumeric strings can be printed in several ways. The PRINT statement can be used with commas or semicolons to determine spacing. The WRITE statement, referencing the print format or FORMAT statement, can also be used.

In this example, the two strings from lines 20 and 30 are printed 4 times; each is printed a different way.

```
10 DIM A$[25],B$[25]
20 A$="Abegg, J. 7/30/51"
30 B$="Herder, A. 7/7/49"
40 PRINT A$,B$
50 PRINT A$;B$
60 FORMAT 2/,10X
70 WRITE (2,60)A$,B$
80 WRITE (2,*)A$,B$
90 END
```



```
Abegg, J. 7/30/51             Herder, A. 7/7/49
Abegg, J. 7/30/51Herder, A. 7/7/49

          Abegg, J. 7/30/51             Herder, A. 7/7/49
Abegg, J. 7/30/51             Herder, A. 7/7/49
```

Lines 40 and 50 print the two strings (A$ and B$), showing the spacing obtained using commas or semicolons.

Line 70 uses the formatted WRITE statement. As shown, the FORMAT statement can determine the position of the printout (horizontally and vertically).

Line 80 is a WRITE statement which uses the print format; the printout is the same using PRINT with commas.

Clearing the Printer

By pressing RESET on the 9831A the following takes place in the printer –

- The printer buffer is cleared of all characters.
- Any current printer movement is completed.
- Form feeds are terminated.

The STAT Function

STAT select code

The STAT (status) function reads the status lines of the specified interface card and returns a decimal value. The value, or status byte, is a combined code indicating the condition of up to nine individual status lines or bits. Each bit can be either 0 (false) or 1 (true), depending on various interface and peripheral status conditions.

The status bits available with the 9871A Printer (via a 98032A Interface) are shown on the following page.

9871A Printer Status Bits

BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Peripheral Status	0	0	1	0	0	0	Data Ready	Buffer Ready

Bit 0 – Buffer Ready 0 indicates that the printer buffer is filled. 1 indicates that space is available for more characters.

Bit 1 – Data Ready 0 indicates that the printer is ready to accept another data character. 1 indicates that the printer is not ready for more data.

Bits 2, 3, 6 and 7 – The bits are always 0 with the 9831A.

Bits 4 and 5 – Interface I.D. These bits are always 0 and 1, respectively on a 98032A Interface.

Bit 8 – Peripheral Status 1 indicates that the printer is connected and ready to use. 0 indicates that the printer is not ready: the front cover is off, the interface is not connected properly or the printer is not switched on.

Notice that each bit has a unique binary value or weight; bit 0 has a true value of 1, bit 2 is 2, bit 5 is 32 and bit 8 is 256. The STAT function returns the sum of all bits currently true. Since bit 5 is always true (32) and bit 8 is true when the printer is OK (decimal 256), all "printer OK" status bytes will be 288 or greater.

Here is a list of typical status bytes and their meanings –

9871A Printer Status Bytes

Status Byte	Condition
0	No interface at select code.
32	Printer disconnected or turned off.
33	Printer front cover is off.
288	Printer is OK but the buffer is full and the printer is not ready for data.
289	Printer is OK, buffer space is available, but the printer is not ready for data.
290	Printer is OK and ready for data, but the buffer is full.
291	The printer is OK, ready for data, and the buffer is available.

The next program checks 9871A Printer status and either waits until the printer is ready to accept more data or stops the program because the printer is not operational.

```
70 IF STAT2=291 THEN 120
80 DISP "Printer Not Ready"
90 WAIT 500
100 IF STAT2<288 THEN 80
110 GOTO 70
120 PRINT A$
```

9871A Printer Function Codes

The many form control features of the 9871A Printer can be used to allow virtually any type of form or format to be printed. This section lists the printer functions and the function codes used to select each control function. Examples are also given for applying the function codes to various printing requirements. For a complete description of each printer function, refer to the 9871A Printer Manual.

Although some of the functions are standard ASCII control characters, many of them are codes that are unique to the 9871A Printer. These unique codes are prefixed by the ASCII escape character ESC (decimal 27). The following table is a summary of the control function codes and the code parameters used to specify them.

9871A Function Codes

COMMAND	COMMAND CODE		
	ASCII CHARACTER	DECIMAL CODE	DECIMAL PARAMETER
GENERAL			
• Bell	BELL	7	
• Backspace	BS	8	
• Linefeed	LF	10	
• Carrier Return	CR	13	
• Shift Out	SO	14	
• Shift In	SI	15	
• View Delay	ESC, D	27, 68	INT(N/64),INT N
• Reverse Line Feed	ESC, LF	27, 10	
• Character Replacement	ESC, C	27, 67	"CHAR", N, "LIST"
• Reset	ESC, E	27, 69	
• Self Test	ESC, R	27, 122	
HORIZONTAL TABULATION			
• Set Horizontal Tab	ESC, 1	27, 49	
• Clear Horizontal Tab	ESC, 2	27, 50	
• Clear All Horizontal Tabs	ESC, 3	27, 51	
• Horizontal Tab Right	Htab	9	
• Horizontal Tab Left	ESC, 4	27, 52	
VERTICAL TABULATION			
• Set Vertical Tab	ESC, 5	27, 53	
• Clear Vertical Tab	ESC, 6	27, 54	
• Clear All Vertical Tabs	ESC, 7	27, 55	
• Vertical Tab Up	ESC, 8	27, 56	
• Vertical Tab Down	Vtab	11	
FORM AND MARGIN CONTROL			
• Set Top of Form	ESC, T	27, 84	
• Form Length	ESC, F	27, 70	INT(N/64), INT N
• Text Length	ESC, L	27, 76	INT(N/64), INT N
• Form Feed	FF	12	
• Set Left Margin	ESC, M	27, 77	
• Text Width	ESC, W	27, 87	INT(N/64), INT N
SPACING CONTROL			
• Horizontal Spacing	ESC, H	27, 72	INT(N/64), INT N
• Vertical Spacing	ESC, V	27, 86	INT(N/64), INT N
• Variable Spacing	ESC, S	27, 83	
PLOTTING CONTROL			
• Absolute Plot	ESC, A	27, 65	INT(X/64), INT X, INT(Y/64), INT Y
• Relative Plot	ESC, R	27, 82	INT(X/64), INT X, INT(Y/64), INT Y
• Character Fill Setup	ESC, .	27, 46	P1, INT(P2/64), INT P2, P3
• Absolute Plot With Fill	ESC, a	27, 97	INT(X/64), INT X, INT(Y/64), INT Y
• Relative Plot With Fill	ESC, r	27, 114	INT(X/64), INT X, INT(Y/64), INT Y
• Plot Origin	ESC, O	27, 79	INT(X/64), INT X, INT(Y/64), INT Y

Sending Function Codes

The function codes and parameter values can be sent to the printer with a WRITE statement referencing a FORMAT statement containing an appropriate number of B numeric specifications. Alternately, the WBYTE (write byte) function can be used. Each function code must be sent to the printer as the binary equivalent of its decimal value. The WBYTE syntax is –

WBYTE decimal value

The WBYTE function sends the characters to the printer in binary form without referencing a FORMAT statement. For example, to sound the printer's bell two times would require executing this statement –

```
30 WRITE (2,*)WBYTE7,WBYTE7
```

General

Bell (7)

Example – To sound the printer's bell, execute the following lines.

```
10 FORMAT B
20 WRITE (2,10)7
30 END
```

Backspace (8)

Example – Five backspaces are executed by the following lines.

```
10 FORMAT 5B
20 WRITE (2,10)8,8,8,8,8;
30 END
```

Linefeed (10)

Example – Two linefeeds are executed by the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)10,10;
30 END
```

Reverse Linefeed (27,10)

Example – The paper will move back one line, by executing the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,10;
30 END
```

Carrier Return (13)

Example – The following lines will move the carrier to the left margin.

```
10 FORMAT B
20 WRITE (2,10)13;
30 END
```

Shift Out (14)

Example – To shift out, execute the following lines.

```
10 FORMAT B
20 WRITE (2,10)14;
30 END
```

Shift In (15)

Example – To shift in, execute the following lines.

```
10 FORMAT B
20 WRITE (2,10)15;
30 END
```

View Delay (27,68,INT(N/64),INT N)

Example – To delay the view feature 750 milliseconds, execute the following lines.

```
10 FORMAT 4B
20 WRITE (2,10)27,68,INT(750/64),750;
30 END
```

Character Replacement (27,67,"CHARACTER",N,"LIST")

Example – The following lines will replace the "G" key with the word "GRASP". Each time "G" is printed, the word "GRASP" will be substituted.

```
10 FORMAT 4B
20 WRITE (2,10)27,67,"G",5,"GRASP";
30 END
```

Reset (27,69)

Example – To reset the printer, execute the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,69;
30 END
```

Test (27,122)

Example – To test the printer either press the TEST button on the printer's rear panel, or execute the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,122;
30 END
```

The following program utilizes some of the general control functions to title and print a table of the values N, N^2 and $\sqrt{N}$ ranging from 1 thru 100.

```
10 I=1
20 FORMAT 13X,"Number Values",4B,8X,"N",10X,"N",2B,
                              "2",10X,"2",2B,"√N",B
30 WRITE (2,20)13,10,10,10,27,10,8,10,10
40 FORMAT 5X,F4.0,F11.0,F15.5
50 FOR N=0 TO 100
60 WRITE (2,40)N,N^2,SQR(N)
70 I=I+1
80 IF I <= 10 THEN 110
90 WRITE (2,*)
100 I=1
110 NEXT N
120 FORMAT 5B
130 WRITE (2,120)10,10,10,10,10
140 END
```

The first WRITE statement references the FORMAT statement in line 20 to title the table. Notice that the function codes are sent from the WRITE statement as they are needed, while the text and the appropriate B specs are placed in the FORMAT statement. The remainder of the program prints the calculated values in appropriate sized fields and number formats. Lines 70 thru 100 puts a space between each group of 10 rows. Line 130 sends five linefeeds to advance the paper when the table is complete. Here's part of the printout –

Number Values

```
                 2          2
     N           N         √N

     0           0     0.00000
     1           1     1.00000
     2           4     1.41421
     3           9     1.73205
     4          16     2.00000
     5          25     2.23607
     6          36     2.44949
     7          49     2.64575
     8          64     2.82843
     9          81     3.00000

    10         100     3.16228
    11         121     3.31662
    12         144     3.46410
    13         169     3.60555
    14         196     3.74166
    15         225     3.87298
    16         256     4.00000
    17         289     4.12311
```

Line 60 in the program listing on the previous page shows the ↑ (exponential) as a ∧ (circumflex). "Shift Out" (14) replaces the circumflex (∧ decimal 94) character with the exponentiation (↑) and the cute accent (′decimal 39) character with an apostrophe ('). This is cancelled by "Shift In" (15).

Horizontal Tabulation

Set Horizontal Tab (27,49)

Example – If the carrier has been set to the desired tab position, the following lines set a tab at that position.

```
10 FORMAT 2B
20 WRITE (2,10)27,49;
30 END
```

Clear Horizontal Tab (27,50)

Example – To clear a tab, move to the tab position and execute the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,50;
30 END
```

Clear all Horizontal Tabs (27,51)

Example –

```
10 FORMAT 2B
20 WRITE (2,10)27,51;
30 END
```

Horizontal Tabulation Right (9)

Example – The following lines move the carrier two tab positions to the right.

```
10 FORMAT 2B
20 WRITE (2,10)9,9;
30 END
```

Horizontal Tabulation Left (27,52)

Example – The following lines move the carrier one tab position tc .he left.

```
10 FORMAT 2B
20 WRITE (2,10)27,52;
30 END
```

In the following example, the RBYTE function in line 20 is used to read keys from the keyboard using select code 0. In lines 30 thru 60, the carrier can be moved either left or right by pressing ← or →, respectively. Then, when the carrier is positioned in the desired location, the horizontal tab can be set or cleared by pressing STORE or CLEAR, respectively (lines 70 thru 100). When CONT is pressed (line 110) the program ends.

```
10 REM"Horizontal Tab Routine"
20 A=RBYTE(0)
30 IF A#14 THEN 50
40 WRITE (2,*)WBYTE8;
50 IF A#15 THEN 70
60 WRITE (2,*)WBYTE32;
70 IF A#13 THEN 90
80 WRITE (2,*)WBYTE27,WBYTE49;
90 IF A#18 THEN 110
100 WRITE (2,*)WBYTE27,WBYTE50;
110 IF A=25 THEN 130
120 GOTO 20
130 END
```

Vertical Tabulation

Set Vertical Tab (27,53)

Example – To set a vertical tab, use linefeed commands to move the platen to the desired tab position. Then execute the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,53;
30 END
```

Clear Vertical Tab (27,54)

Example – To clear a tab, move to the tab position and execute the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,54;
30 END
```

Clear All Vertical Tabs (27,55)

Example – To clear all the vertical tabs, execute the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,55;
30 END
```

Vertical Tab Down (11)

Example – The following lines move the paper up (print position down) to the next vertical tab position.

```
10 FORMAT B
20 WRITE (2,10)11;
30 END
```

Vertical Tab Up (27,56)

Example – The following lines move the paper down (print position up) to the next vertical tab position.

```
10 FORMAT 2B
20 WRITE (2,10)27,56;
30 END
```

The next example program uses the RBYTE (read byte) function in line 20 to read keys and to set or clear vertical tabs. When ↑ or ↓ is pressed the platen is moved up or down, respectively (lines 30 thru 60). When the platen is in the desired location, press STORE or CLEAR to set or clear the vertical tab at that location (lines 70 thru 100). Press CONT to halt the program.

```
10 REM"Vertical Tab Routine"
20 A=RBYTE(0)
30 IF A#16 THEN 50
40 WRITE (2,*)WBYTE10;
50 IF A#17 THEN 70
60 WRITE (2,*)WBYTE27,WBYTE10;
70 IF A#13 THEN 90
80 WRITE (2,*)WBYTE27,WBYTE53;
90 IF A#18 THEN 110
100 WRITE (2,*)WBYTE27,WBYTE54;
110 IF A=25 THEN 130
120 GOTO 20
130 END
```

Form and Margin Control

Top of Form (27,84)

Example – Execute the following lines to set the top of form.

```
10 FORMAT 2B
20 WRITE (2,10)27,84;
30 END
```

Form Length (27,70,INT(N/64),INT N)

Example – To set a form length of 8 inches (768 increments), execute the following lines.

```
10 FORMAT 4B
20 WRITE (2,10)27,70,INT(768/64),768;
30 END
```

Text Length (27,76,INT(N/64),INT N)

Example – To set a text length of 7 inches (672 increments), execute the following lines.

```
10 FORMAT 4B
20 WRITE (2,10)27,76,INT(672/64),672;
30 END
```

Formfeed (12)

Example – These lines move to the top of the next form.

```
10 FORMAT B
20 WRITE (2,10)12;
30 END
```

Set Left Margin (27,77)

Example – After spacing to the desired left margin, execute the following lines.

```
10 FORMAT 2B
20 WRITE (2,10)27,77;
30 END
```

Text Width (27,87,INT(N/64),INT N)

Example – To define a text width of 6 inches (720 increments), execute the following lines.

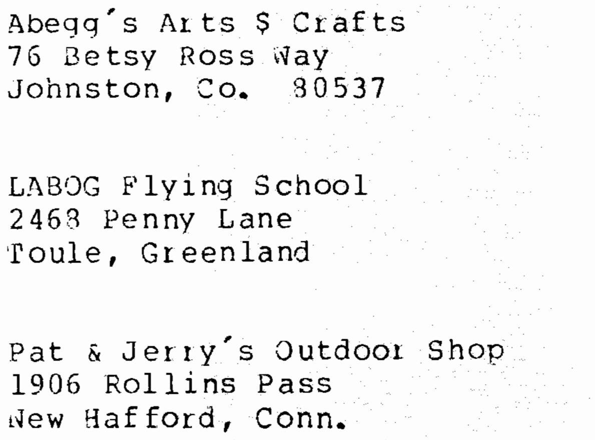
```
10 FORMAT 4B
20 WRITE (2,10)27,87,INT(720/64),720;
30 END
```

This program uses some of the form and margin control functions to print shipping labels. It assumes that a list of names and addresses have previously been recorded on track 0 of the tape cartridge. Each file contains one string variable holding a complete name and address. A program in Chapter 8 of the 9831A Operating and Programming Manual was used to store the data on tape.

```
10 DIM A$[90]
20 DISP "SET TOP OF FORM & CONTINUE";
30 STOP
40 FORMAT 4B
50 WRITE (2,40)27,84;
60 WRITE (2,40)27,70,INT(144/64),INT144
70 WRITE (2,40)27,76,INT(96/64),INT96
80 WRITE (2,40)27,87,INT(360/64),INT360
90 DISP "STARTING FILE";
100 INPUT F
110 FOR I=F TO F+20
120 LOAD  DATA I,A$
130 IF A$="STOP" THEN 190
140 WRITE (2,*)A$[1,30]
150 WRITE (2,*)A$[31,60]
160 WRITE (2,*)A$[61,90]
170 WRITE (2,40)12
180 NEXT I
190 WRITE (2,40)12
200 END
```

- Line 50 sets the top of form.
- Line 60 sets the form lengths to 1 1/2 inches.
- Line 70 sets the text length to 1 inch.
- Line 80 sets the text width to 3 inches.
- Lines 90 and 100 request and input the starting file number.
- Lines 110 thru 180 print the 20 labels. The first 25 characters of C$ are the name; the second 25 characters are the address, and the last 25 characters are the city, state, zip code. Line 170 sends a form feed to the printer to advance the form. If the first four characters of A$ are "STOP", the program exits the printout loop.

Here are some example printouts –



```
Abegg´s Arts $ Crafts
76 Betsy Ross Way
Johnston, Co.  80537


LABOG Flying School
2468 Penny Lane
Toule, Greenland


Pat & Jerry´s Outdoor Shop
1906 Rollins Pass
New Hafford, Conn.
```

Spacing Control

Horizontal Spacing (27,72,INT(N/64),INT N)

Example – To increase the character spacing to twenty, 1/120 inch increments per character, execute the following lines.

```
10 FORMAT 4B
20 WRITE (2,10)27,72,INT(20/64),20;
30 END
```

Vertical Spacing (27,86,INT(N/64),INT N)

Example – To decrease the spacing between lines to twelve, 1/96 inch increments (8 lines/inch) execute the following lines.

```
10 FORMAT 4B
20 WRITE (2,10)27,86,INT(12/64),12;
30 END
```

Variable Spacing (27,83)

Example – To stretch out a word, the following lines can be used. The maximum spacing parameter in this mode is 31.

```
10 FORMAT 10B
20 WRITE (2,10)27,83;
30 WRITE (2,10)"L"10"O"12"N"14"G"16"E"18"R"
40 END
```

Result –

```
LONG E R
```

Plotting Control

Plot Absolute (27,65,INT(X/64),INT X,INT(Y/64),INT Y)

Example – The following program can be used to plot with the absolute plot function. Enter the X and Y coordinates (in inches) and enter a character to be printed at that point.

```
10 DISP "ABS X,Y,CHARACTER";
20 INPUT X,Y,S$[1,1]
30 X=X*120
40 Y=Y*96
50 WRITE (2,60)27,65,INT(X/64),INTX,INT(Y/64),INTY,S$[1,1];
60 FORMAT 10B
70 GOTO 10
80 END
```

Plot Relative (27,82,INT(X/64),INT X,INT(Y/64),INT Y)

Example – The following program can be used to plot with the relative plot function. Enter the X and Y coordinates (in inches) and enter a character to be printed at that point.

```
10 DISP "REL X,Y,CHARACTER";
20 INPUT X,Y,S$[1,1]
30 X=X*120
40 Y=Y*96
50 WRITE (2,60)27,82,INT(X/64),INTX,INT(Y/64),INTY,S$[1,1];
60 FORMAT 10B
70 GOTO 10
80 END
```

Character Fill Setup (27,46,P1,INT(P2/64),P2,P3)

Example – The following program can be used to set the character fill parameters. Enter the fill character, spacing and offset desired (as indicated).

```
10 DISP "FILL CHAR, SPACING,OFFSET";
20 INPUT S$[1,1],X,Y
30 WRITE (2,40)27,46,S$[1,1],INT(X/64),INTX,INTY;
40 FORMAT 10B
50 END
```

Absolute Plot With Fill (27,97,INT(X/64),INT X,INT(Y/64),INT Y)

Example – The following program can be used to plot with the absolute plot fill function. Enter the X and Y coordinates (in inches) and enter a character to be printed at that point.

```
10 DISP "ABS & FILL X,Y,CHARACTER";
20 INPUT X,Y,S$[1,1]
30 X=X*120
40 Y=Y*96
50 WRITE (2,60)27,97,INT(X/64),INTX,INT(Y/64),INTY,S$[1,1];
60 FORMAT 10B
70 GOTO 10
80 END
```

Relative Plot With Fill (27,114,INT(X/64),INT X,INT(Y/64),INT Y)

Example – The following program can be used to plot the relative plot with fill function. Enter the X and Y coordinates and enter a character to be printed at that point.

```
10 DISP "REL & FILL X,Y,CHARACTER";
20 INPUT X,Y,S$[1,1]
30 X=X*120
40 Y=Y*96
50 WRITE (2,60)27,114,INT(X/64),INTX,INT(Y/64),INTY,
                                          S$[1,1];
60 FORMAT 10B
70 GOTO 10
80 END
```

Plot Origin (27,79,INT(X/64),INT X,INT(Y/64),INT Y)

Example – The following lines move the plot origin 2 inches (240 increments) to the right of the left margin (X) and 3 inches (288 increments) up from the lower margin (Y).

```
10 FORMAT 6B
20 WRITE (2,10)27,79,INT(240/64),240,INT(288/64),288
30 END
```

The 9871A Option 001

The Option 001 Printer is designed to be used with the HP-IB (Hewlett Packard Interface Bus). The I/O operations described in this chapter are also used with the Option 001 printer; however, the select code for HP-IB is somewhat different. The HP-IB select code format follows the form – ccdd

cc is a one or two digit number from 2 thru 15 specifying the interface card select code.

dd is a two digit number from 00 thru 30 which specifies the device address of the 9871A Printer on the HP-IB.

The address code within each select code parameter specifies the appropriate address (origin or destination) of the device on the bus. All I/O operations which have an address code automatically transmit the computer's preset talker/listener address and specified address code in the appropriate order on the bus. This address sequence is preceded by the bus unlisten command to clear all listeners previously set. For example, this statement `WRITE (711,10)` sends this address sequence before sending data –

- Unlisten command
- Computer talk address
- Listen address 11 (on bus 7).

7

The `10` in the WRITE statement references a FORMAT statement.

9871A Character Disks

Decimal Code	Spoke Number	Standard Characters
33	48	!
34	61	"
35	23	#
36	24	$
37	5	%
38	9	&
39	2	′
39 (SO)	89	'
40	38	(
41	36	)
42	26	*
43	25	+
44	6	,
45	27	-
46	33	.
47	31	/
48	18	0
49	17	1
50	19	2
51	16	3
52	20	4
53	15	5
54	21	6
55	14	7
56	22	8
57	13	9
58	52	:
59	59	;
60	10	<
61	29	=
62	12	>
63	54	?
64	7	@
65	47	A
66	58	B
67	39	C
68	53	D
69	45	E
70	57	F
71	34	G
72	41	H
73	50	I
74	66	J
75	30	K
76	55	L
77	37	M
78	49	N
79	51	O
80	56	P
81	65	Q
82	43	R
83	44	S
84	46	T
85	60	U
86	62	V
87	32	W
88	28	X
89	35	Y
90	63	Z
91	42	[
92	3	√
93	40	]
94	1	^
94 (SO)	8	↑
95	11	—
96	95	\
97	85	a
98	93	b
99	75	c
100	86	d
101	82	e
102	74	f
103	73	g
104	81	h
105	79	i
106	91	j
107	72	k
108	87	l
109	90	m
110	78	n
111	77	o
112	92	p
113	69	q
114	83	r
115	84	s
116	80	t
117	76	u
118	71	v
119	88	w
120	70	x
121	67	y
122	68	z
123	4	π
124	64	|
125	94	→
126	96	∿

Here are the unique characters found on the ASCII character disk.

Decimal Code	Spoke Number	ASCII Character
92	3	\
123	4	{
125	94	}

Here are the unique characters found on the European character disk.

Decimal Code	Spoke Number	European Character
35	23	£
39 (SO)	89	°
92	3	Ç
94 (SO)	8	¿
123	4	¨ (U.C.)
125	94	¨ (L.C.)

The character spokes are numbered clockwise, 1 to 96, beginning with the circumflex (^) character. Hold the character disk with the characters facing you and the locating tab up. The circumflex character (spoke 1) will be on top.

Chapter 8

The HP 9877A External Tape Memory

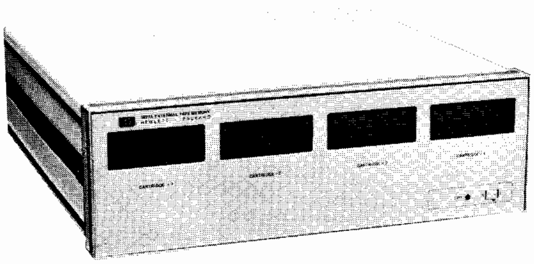
Introduction

This chapter describes the unique operations used to control the HP 9877A External Tape Memory. The tape cartridge units fitted into the external tape memory are identical in all respects to the tape cartridge unit in the 9831A. Tape care, format and statements are also identical. Therefore, the user should be familiar with Chapter 6 "Tape Cartridge Operation" in the 9831A Operating and Programming Manual.

Description

The HP 9877A provides the HP 9831A with two valuable additional capabilities. Offering up to one megabyte of tape memory, the peripheral presents an inexpensive method of storing large volumes of data. Alternately, the peripheral offers a convenient, easy and rapid method of obtaining duplicated tapes.

Three versions of the HP 9877A are available, fitted with one (Standard), two (Option 002) or four (Option 004) tape cartridge units. A 1.5m (5 ft.) terminated interface cable is supplied with the peripheral.



9877A Tape Memory

Use the System Test Booklet and System Test Cartridge to verify proper operation of the 9831A and 9877A System.

The TAPE Statement

TAPE select code

The TAPE statement specifies the address or select code, for all tape cartridge operations. The select code range is 1 thru 15. The internal cartridge uses select code 1, which is automatically specified for tape cartridge statements whenever ERASE A is executed or the 9831A is switched on.

Select Code Combination

The following description is the select code setting for each of the three versions of the HP 9877A.

Restrictions

For all versions select codes 0 and 1 must never be used as these are automatically specified for the computer internal peripherals (display, keyboard and tape unit). Select code 2 is automatically specified for the standard printer when the 9831A is switched on; this can be changed with the STDPR statement. See Chapter 8 of the 9831A Operating and Programming Manual. In addition, select codes 2 and 3 must not be used with the 9877A Option 004 (four cartridge unit).

Standard (One Cartridge Unit)

Select codes 2 thru 15 are available. This interface select code setting must be the same number used with the TAPE statement executed from the keyboard or from a program.

8

9877A Option 002 (Two Cartridge Unit)

Select Code	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tape Cartridge	#1	#2	#1	#2	#1	#2	#1	#2	#1	#2	#1	#2	#1	#2

Select codes 2 thru 15 are available. To identify the two cartridge units as one peripheral the select codes are divided into pairs. Setting one number on the interface automatically specifies its partner number for the same interface. Either number can then be used with the TAPE statement to specify the individual cartridge unit contained in the 9877A. For each number pair, the lower number always specifies tape cartridge #1 and the higher number tape cartridge #2.

For example, if the select code on the interface were set to 5, execute the following statements to store a program on tape cartridge #1.

`TAPE 4`	Press EXEC
`STORE# 1,0`	Press EXEC

The program in the 9831A would then be stored on tape cartridge #1, track 1, file 0.

9877A Option 004 (Four Cartridge Unit)

Select Code	4	5	6	7	8	9	10	11	12	13	14	15
Tape Cartridge	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4

Select codes 4 thru 15 are available, and these are divided into groups of four as shown. Setting one number on the interface automatically specifies the remaining three numbers for the same interface; they cannot be used for other peripheral devices. Each group of four numbers are then used with the TAPE statement to identify the individual cartridge units. The lower number specifies cartridge unit #1 whereas the higher number specifies cartridge unit #4.

For example, if the select code on the interface were set to 8, execute the following statements to load a program from cartridge #3.

`TAPE 10`	Press EXEC
`LOAD #1,3`	Press EXEC

The program which was stored on tape cartridge #3, track 1, file 3 would then be reproduced in the 9831A memory.

Tape Duplication

Tape duplication can be accomplished with a very simple program. The programs to be duplicated are loaded into the 9831A memory then stored onto another tape. With the following program, duplication is accomplished using the LINK and STORE statements. The program assumes that the files are already marked on the tape which is to receive the data or programs. Program transfer with this duplication program will have a change in line numbers. The line numbers of the duplicated program will begin with 100. This is done so as not to overlay the duplicating program which is stored in memory.

Here's the program –

```
10 I=0
20 TAPE 11
30 LINK I,100
40 TAPE 8
50 STORE I,100
60 I=I+1
61 IF I>50 THEN 70
62 DISP I
63 WAIT 100
65 GOTO 20
70 END
```

The number of files to be duplicated are controlled by the variable I. Files are loaded, via LINK line 20, from cartridge #4 which is specified by tape 11. The loaded file is then stored on cartridge #1 using the TAPE 8 and STORE I,100 statements.

Chapter 9

The HP 9881A Line Printer

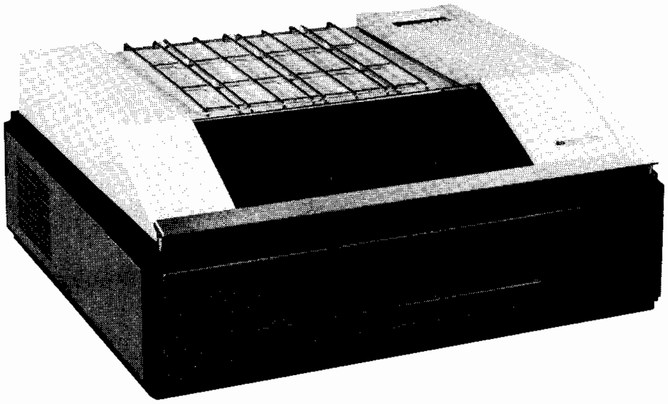
Introduction

This chapter describes the functions and codes unique to the 9881A. A brief explanation of the controlling statements is given. Chapter 2 of this manual is devoted to introduction and explanation of the controlling statements. A complete list of the error codes is in the Appendix.

The HP 9881A Line Printer is connected to the 9831A via the 98032A Option 181 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90049). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

The HP 9881A Line Printer is a 5 x 7 dot matrix printer that can be used for such applications as payroll, inventory, accounting and report generation. The 9881A prints 200 lines per minute using full 132-column line width. An 8-channel Vertical Format Unit (VFU) facilitates automatic form-filling and vertical formatting under program control. The 64 ASCII character set is standard on the HP 9881A Printer.



HP 9881A Line Printer

Standard Printer

STDPR select code

The HP 9881A Line Printer can be used as the "standard printer" for the 9831A using the (STDPR) statement. Successive PRINT, LIST, TLIST, TRACE and print-all operations are then output to the 9881A Printer. The select code range is 2 thru 15. At turn on the 9831A automatically specifies select code 2 for its standard printer.

Printer Operations

Data and messages are transferred to the printer by the interface. The printer stores the characters in a buffer until a linefeed command is given or the buffer is full (132 characters). The statements and commands used to control the printer are described in Chapter 2 of this manual and also in Chapter 8 of the 9831A Operating and Programming Manual. You may also wish to refer to "Printer Character Codes" later in this chapter for a complete list of the printer's character set and decimal codes. The 9881A Option 001 provides lower case alpha characters. Following is a list of the operations with a brief description.

PRINT Prints text and numbers on the standard printer.

TAB and SPA Outputs a specified block of spaces to position output items with PRINT, WRITE and DISP statements.

LIN Outputs a specified number of linefeeds; used with PRINT and WRITE statements.

WRITE Outputs text and numbers to any output device. Items are output in either the PRINT format or in the form specified by a referenced FORMAT statement.

FORMAT Specifies the output form for variables and edit specifications for controlling an output printer.

LIST Prints program lines on the standard printer.

TLIST Prints information about each tape file on the standard printer.

Printer Status

This function returns a decimal code indicating the current status of the printer. The STAT syntax, status codes and definitions are listed below.

STAT select code

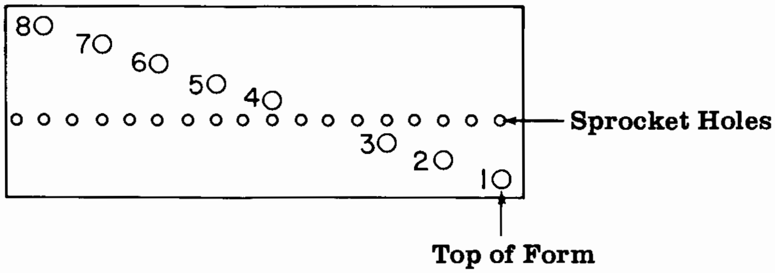
Status Code	Meaning
288	Printer ready.
32	No printer power or Print switch not pressed or Printer out of paper.

Vertical Format Control

Special paper instructions from the 9831A can be used to control the movement of paper. When a paper instruction command is received, a print cycle will occur if at least one printable character has been received by the printer since the previous print cycle. If no print cycle is pending, the paper movement will commence immediately. Two sets of paper instructions are available, the Vertical Format Unit control and the Line Count control.

Vertical Format Unit Control

This set of paper instruction commands is used to control the duration of movement according to the punches in the printer's Vertical Format Unit (VFU) tape. The paper will move until a hole is sensed on the VFU tape in the channel specified by the paper instruction (see following figure).



Refer to the 2607A Operator's Manual (HP P/N 02607-90005) for information to construct the format control loop tape.

The decimal codes used to select the various VFU channels are listed below.

VFU Codes

9831A Decimal Code	VFU Channel Selected
320	1 – Top of Form
321	2
322	3
323	4
324	5
325	6
326	7
327	8

Line Count Control

This set of paper instructions is used to control the duration of movement according to the number of lines specified in the command. Since paper advance is an integral part of the printing process, a movement of one line occurs as part of every print cycle. The action taken upon receipt of a movement command, as shown below, depends upon whether or not printable characters have been sent to the printer since the last paper instruction. The following table lists the decimal codes used to move the paper and the number of lines skipped.

Line Count Codes

	Number of lines skipped	
9831A Decimal Code	No Print Cycle	With Print Cycle
256	0	0
257	1	0
258	2	1
259	3	2
260	4	3
261	5	4
262	6	5
263	7	6
264	8	7
265	9	8
266	10	9
267	11	10
268	12	11
269	13	12
270	14	13
271	15	14

All the paper instruction codes are used by including the decimal code in a WRITE statement, and referencing the WRITE statement to a FORMAT B statement. FORMAT B is used to output codes not otherwise available on the 9831A.

Here is an example of a paper instruction as it might be used in a program. The following program segment outputs the decimal code 263, which will cause the paper to skip 7 lines (6 with a print cycle).

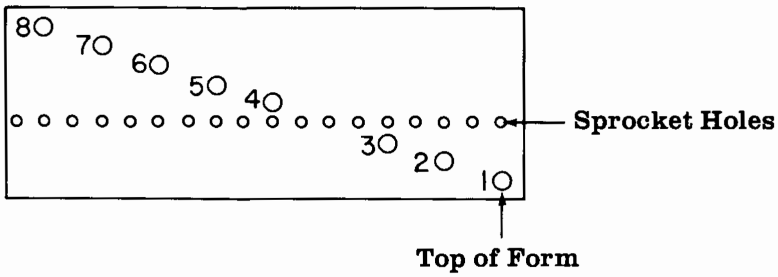
```
80 FORMAT B
90 WRITE (6,80)263
```

Format Tape

The Standard vertical format tape shipped with the printer provides a format for two 11 inch forms at 6 lines per inch. This tape is punched as follows –

Channel	Function	Hole Positions
1	Move to top of next form	1 and 67
2	Move to bottom of form	60 and 126
3	Single space	1,2,3...60; 67,68...126
4	Double space	1,3,5...59; 67,69...125
5	Triple space	1,4,7...58; 67,70...124
6	Move to half page line	1,31; 67,97
7	Move to next quarter page	1,16,31,46; 67,82,97,112
8	Move to next sixth page	1,11,21...51; 67,77,87...117

Note that paper movement is from one hole in a channel to the next hole in the same channel.



VFU Tape Channels

Master Clear

The printer circuits can be reset by sending a master clear code to the printer. The master clear decimal code (512) is sent to the printer in the same manner as the paper instructions. The master clear will clear the printer's input buffer and terminate any paper movement or VFU operation in progress. If a print cycle is in progress, it will be completed.

Printer Character Codes

Decimal Code	Character	Decimal Code	Character	Decimal Code	Character	Opt-001 Character
32	SPACE	60	<	90	Z	
33	!	61	=	91	[	
34	"	62	>	92	\	
		63	?	93	]	
35	#	64	@	94	^	
36	$					
37	%	65	A	95	—	
38	&	66	B	96	@	
39	'	67	C	97	A	a
		68	D	98	B	b
40	(	69	E	99	C	c
41	)					
42	*	70	F	100	D	d
43	+	71	G	101	E	e
44	,	72	H	102	F	f
		73	I	103	G	g
45	—	74	J	104	H	h
46	.					
47	/	75	K	105	I	i
48	0	76	L	106	J	j
49	1	77	M	107	K	k
		78	N	108	L	l
50	2	79	O	109	M	m
51	3					
52	4	80	P	110	N	n
53	5	81	Q	111	O	o
54	6	82	R	112	P	p
		83	S	113	Q	q
55	7	84	T	114	R	r
56	8					
57	9	85	U	115	S	s
58	:	86	V	116	T	t
59	;	87	W	117	U	u
		88	X	118	V	v
		89	Y	119	W	w
				120	X	x
				121	Y	y
				122	Z	z
				123	[	{
				124	\	¦
				125	]	}
				126	^	~
				127	—	▒

Chapter 10

The HP 9883A Tape Reader

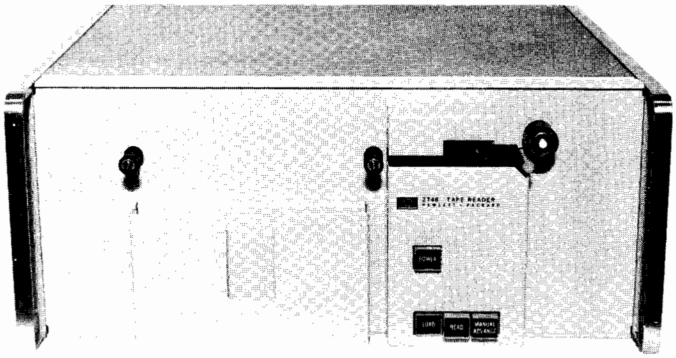
Introduction

This chapter describes the general I/O operations used to control the tape reader: ENTER statement, RBYTE function and the PTAPE command. Numeric data, string data and programs can be entered from the tape reader.

The HP 9883A Tape Reader is connected to the HP 9831A via a 98032A Option 183 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90050). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

The HP 9883A is a high-speed tape reader used with the 9831A. Standard one-inch paper, mylar or plastic tapes punched on teletypewriters or other tape punches can be read by the 9883A at speeds up to 500 characters per second. The 9883A can read both oiled and opaque tapes without adjustments. Both data and programs can be entered into the 9831A using the appropriate operation.



9883A Tape Reader

The chapter assumes you are familiar with the operation of the 9831A and the tape reader, as explained in their operating manuals.

Here are the statements and functions used to control the 9883A Tape Reader and a brief description of each. Complete information on these operations is in Chapter 2.

ENTER	Receives data from an external device, can reference a FORMAT statement or Free-Field format and can be used with a conversion table.
RBYTE	Inputs one byte of data from the specified peripheral device and returns a decimal equivalent value.
PTAPE	Command used to input program lines from an external device; press (STOP) to halt the command.

Entering Numeric Data

ENTER (select code , * or line number) variable$_1$[, variable$_2$...]

Numeric data can be entered easily using the ENTER statement with the free-field (*) format. One record is entered at a time. A record is a sequence of data items separated by commas or semicolons which ends with a linefeed (LF). Commas and semicolons are called character delimiters. The linefeed is called the record delimiter. For example, this sequence of ASCII characters is from a tape. Two records are shown –

12.8,−300,.99,5E6 (LF) 15.0,198,.32,1.99E−6 (LF)

This short program reads and prints the individual data items from the tape reader at select code 3.

```
10 DIM Y[4]
20 FOR I=1 TO 2
30 ENTER (3,*)(FORJ=1TO4,Y[J])
40 PRINT Y[1],Y[2],Y[3],Y[4]
50 NEXT I
60 END
```

```
12.8          -300          0.99          5000000

15             198          0.32          1.99000E-06
```

In the program free-field format is used, since commas separate the input data. To read a record which contains only numbers, such as –

972193425 (LF) 693345786 (LF) 667223842 (LF)

A different format spec would have to be referenced. If the numbers have 3 characters each, 3 numbers are contained in each record. To read the above 3 records, the following program could be used –

```
10 FOR I=1 TO 3
20 ENTER (3,30)A,B,C
30 FORMAT 3F3.0
40 PRINT A,B,C
50 NEXT I
60 END
```

Printout –

```
972               193               425
693               345               786
667               223               842
```

In line 30, 3 characters are read; no digits to the right of the decimal point are specified.

Using Strings

In addition to numeric data, alphabetic data can also be input using the tape reader. The following alphanumeric characters are entered as strings. The Δ indicates a blank space.

ROGERS,ΔJOHNΔHT5–10WT160 (LF) SMITH,ΔARTHURΔHT6–2,WT230 (LF)

This program enters and prints the two records –

```
10 DIM A$[80]
20 FOR I=1 TO 2
30 ENTER (3,*)A$
40 PRINT A$
50 NEXT I
60 END
```

```
ROGERS, JOHN HT5-10,WT160
SMITH, ARTHUR HT6-2,WT230
```

The linefeed (LF) serves as the string delimiter. Commas in numeric fields are used as delimiters for numeric data, but when entering strings, commas are interpreted as part of the string that is entered. To enter mixed string and numeric data, such as student names followed by grades this program can be used –

ROGERS,ΔJOHNΔΔΔΔ80,75,83 LF SMITH,ΔARTHURΔΔΔ90,93,88 LF

```
10 DIM A$[80]
20 PRINT "NAME","  ","AVG."
30 OUTPUT (2,40)
40 FORMAT 34"="
50 FOR I=1 TO 2
60 ENTER (3,*)A$[1,16],A,B,C
70 FIXED 0
80 PRINT A$,(A+B+C)/3
90 NEXT I
100 END
```

Printout –

```
NAME                                AVG.
==================================
ROGERS, JOHN                         79
SMITH, ARTHUR                        90
```

In line 60, 16 characters are entered into A$, and followed by the student grades which are stored in A, B and C. In line 80, the student name, A$, and the average of A, B and C are printed.

More than one string can be entered at once. For example, to enter two strings at once the following line could be programmed.

```
30 ENTER(3,*)A$,B$
```

Control Characters and Special Characters

Strings which contain ASCII control characters such as The Record Separator (RS) character or The End of Transmission Block (ETB) character and others can be entered and stored, even though they may not be able to be printed or displayed. To check for the presence of a control character use the LEN function. For example, if a record contained an RS (decimal 30), it could be entered from paper tape into A$ by –

```
10 DIM A$[128]
20 ENTER (3,*)A$
30 DISP LEN(A$)
40 END
```

The length of A$ would be 1, the value displayed.

There are four characters which have special meanings when entering data. They are –

ASCII Character	Symbol	Decimal Code	Special Meaning
linefeed	LF	10	End of record delimiter.
escape	ESC	27	Delete record.
left arrow	←	95	Delete previous character.
alternate mode	ALT	126	Delete record.

If an escape character or alternate mode character is encountered in a block of data, the rest of the block is searched for a linefeed and the entire block is omitted.

When a left arrow character is encountered, the character preceding it is ignored. For example, if DAI←Y is entered from paper tape, DAY is the data which is stored.

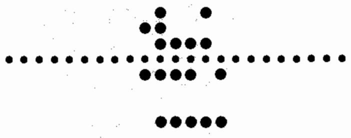
The linefeed is, of course, interpreted as the record delimiter.

The RBYTE function or a conversion table should be used when other meanings are given to the codes in the previous table. Also, the RBYTE function should be used when there is no record delimiter. This is often the case when many data items are entered and the data is to be manipulated in, say, 75-character blocks.

Entering Alphanumeric Data

Entering alphanumeric data can be done by entering ASCII characters as binary values. Sorting, comparing, printing and other operations can be performed on alphanumeric data.

As an example, a record on paper tape contains the following codes in binary –

Decimal Code	ASCII Character	Tape
72	H	
69	E	
76	L	
76	L	
79	O	
10	linefeed	

This program enters the data from the tape and prints it.

```
10 DIM AI[30]
20 ENTER (3,30)(FORX=1TO30,A[X])
30 FORMAT 30B
40 FOR X=1 TO 30
50 WRITE (2,30)A[X];
60 NEXT X
70 PRINT
80 END
```

```
HELLO
```

Note that in line 20, 30 characters can be read, but only 5 characters and a linefeed are actually entered. The linefeed cancels the rest of the implied FOR...NEXT loop. Then, in lines 40 through 70, the message entered from the paper tape is printed using binary (B) format. Line 70 causes the ASCII characters to be printed on the standard printer.

Using Conversion Tables

The use of conversion tables is covered in depth in Chapter 2. An example is included here for your information.

A conversion table is used when reading paper tape which is punched in a code other than ASCII or when certain characters such as the ASCII linefeed (decimal 10) have a different meaning.

In the following example, the conversion table is used to exchange the meaning of the linefeed character and the record separator (RS) character.

The 9831A interprets a decimal 10 as a linefeed, but decimal 30 has no special meaning. In the example, the subscript to the array variable is the incoming code, and the assigned value is the 9831A's ASCII equivalent code.

Program –

```
10 DIM BI[128],L$[80]
20 FOR X=1 TO 128
30 B[X]=X
40 NEXT X
50 B[10]=30
60 B[30]=10
70 ENTER (3,*,B)L$
80 PRINT L$
90 GOTO 70
100 END
```

Lines 50 and 60 define the conversion table B. Note that the conversion table must be dimensioned as an integer-precision array, as in line 10. Line 70 accesses the conversion table and the data is entered into L$. Whenever a decimal 30 is entered from the paper tape, the 9831A will convert this to a linefeed, or decimal 10, and vice-versa.

The RBYTE Function

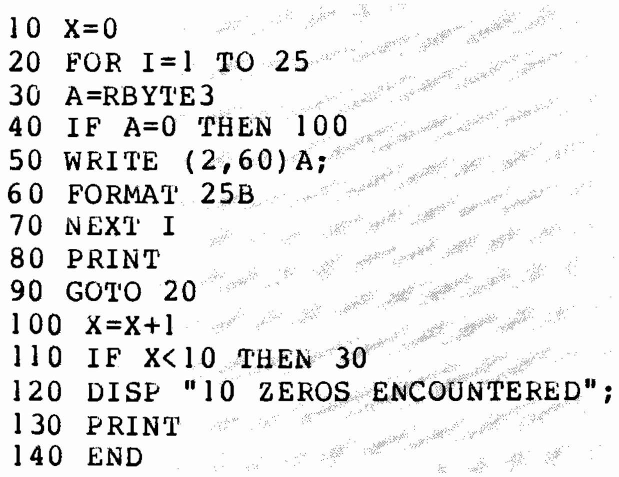
RBYTE select code

To enter data which has no delimiters and is unformatted, the RBYTE (Read Byte) function is used. Although RBYTE is slower than ENTER, any 8-bit data structure can be entered as the decimal equivalent of binary values.

In the following example, the paper tape contains alphanumeric text without delimiters. The information on the tape is to be printed in 25 character segments. This is the information on the tape in ASCII code –

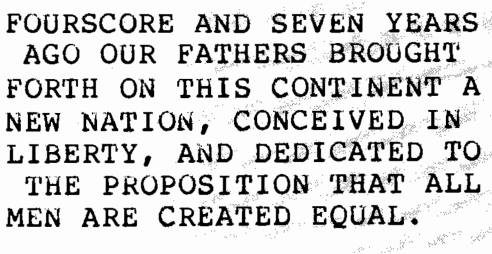
FOURSCORE AND SEVEN YEARS AGO OUR FATHERS BROUGHT FORTH ON THIS CONTINENT A NEW NATION, CONCEIVED IN LIBERTY, AND DEDICATED TO THE PROPOSITION THAT ALL MEN ARE CREATED EQUAL.

Program –



```
10 X=0
20 FOR I=1 TO 25
30 A=RBYTE3
40 IF A=0 THEN 100
50 WRITE (2,60)A;
60 FORMAT 25B
70 NEXT I
80 PRINT
90 GOTO 20
100 X=X+1
110 IF X<10 THEN 30
120 DISP "10 ZEROS ENCOUNTERED";
130 PRINT
140 END
```

Print out –



```
FOURSCORE AND SEVEN YEARS
 AGO OUR FATHERS BROUGHT
FORTH ON THIS CONTINENT A
NEW NATION, CONCEIVED IN
LIBERTY, AND DEDICATED TO
 THE PROPOSITION THAT ALL
MEN ARE CREATED EQUAL.
```

In line 30, one character is entered from the tape reader at select code 3. In line 50, each non-zero character entered in line 10 is printed. The program continues until 10 zeros are encountered.

Entering Programs From Paper Tape

BASIC language programs can be entered using the PTAPE command. The syntax is –

PTAPE select code
or
PTA select code

When the program transfer is complete, press STOP to halt the command. PTAPE is not programmable.

The PTAPE command inputs program lines written in HP 9831A or HP 9830 Basic language and transmitted in ASCII code only. For example, store this short program in the 9831A.

If a tape punch is at select code 9, then LIST #9 will store the program on paper tape. At a later time, the program can be re-entered into the 9831A from the 9883A Tape Reader at any select code. If select code 7 is used, then execute PTAPE 7.

Chapter 11

HP 9884A Tape Punch

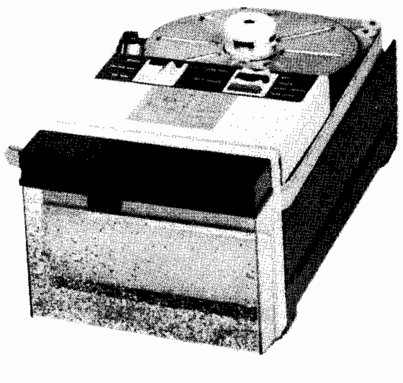
Introduction

This chapter describes the general I/O operations used to control the 9884A Tape Punch: the WRITE, LIST and Write Binary (WBYTE). A brief explanation of the operations is given in this chapter, refer to Chapter 2 for more information on the operations.

The 9884A Tape Punch is connected to the 9831A via the 98032A Option 184 Interface. Installation for this system is explained in the Installation and Service Note (P/N 09831-90051). Use the System Test Booklet and System Test Cartridge to verify proper operation of this system.

Description

The HP 9884A Tape Punch provides the user with a fast and reliable method of transferring the 9831A's output directly onto punched tape. The output can be data or programs. Under program control, the 9884A can punch 5, 6, 7 or 8 level codes at speeds up to 75 characters per second. These codes may be punched onto paper, plastic or mylar tape. If required, the 9884A can be set to punch a parity bit.



HP 9884A Tape Punch

Here are the statements and functions used to control the 9884A Tape Punch –

WRITE — Outputs text and numbers to any output device. Items are output in either the PRINT format or in the form specified by a referenced FORMAT statement.

LIST — Outputs program lines to the device specified by the select code.

WBYTE — Used in WRITE and OUTPUT statements to output a single character specified by a decimal value.

Data Output

The WRITE, FORMAT and LIST statements as well as the WBYTE (write byte) function are used to control the tape punch. Their syntax are reviewed briefly here. Example uses of each operation are shown later in the chapter.

The WRITE Statement

WRITE (select code , * or line number) [text and/or expressions]

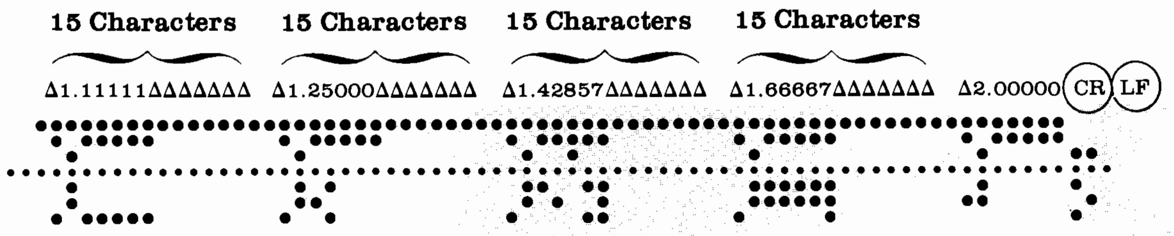
The list of text and expressions is output to the specified device. Remember that the select code range is from 2 thru 15.

The * specifies the print format. The TAB, SPA (space) and LIN (linefeed) functions can be used, and commas or semicolons determine the spacing between output items with the print format. A line number in place of the * references a FORMAT statement for more complete output formatting. The line number must be a constant. When a FORMAT statement is referenced, each output item must be separated by a comma. A semicolon at the end of the WRITE statement suppresses the automatic CR/LF (carriage-return linefeed) sent after the last item.

For example, this program sequence punches five numbers on tape –

```
10 A=10
20 FIXED 5
30 WRITE (9,*)A/9,A/8,A/7,A/6,A/5
40 END
```

Here is the output. The Δ indicates a blank space.

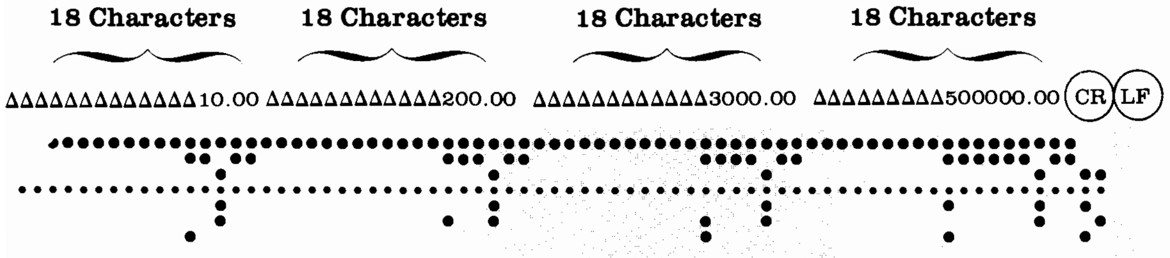


```
Δ1.11111ΔΔΔΔΔΔΔ Δ1.25000ΔΔΔΔΔΔΔ Δ1.42857ΔΔΔΔΔΔΔ Δ1.66667ΔΔΔΔΔΔΔ Δ2.00000
```

As shown, the print format causes each number to appear in a 15-character field. The FIXED 5 statement determines the output form. A CR/LF is output after each five items and after the last item. For more examples of the print format, refer to Chapter 2.

This next sequence punches the four numbers calculated in line 30. The FORMAT statement (line 20) is referenced by the WRITE statement. As shown the format statement provides complete control of the WRITE statement.

```
10 A=10
20 FORMAT 4F18.2
30 WRITE (9,20)A,2*A^2,3*A^3,5*A^5
40 END
```

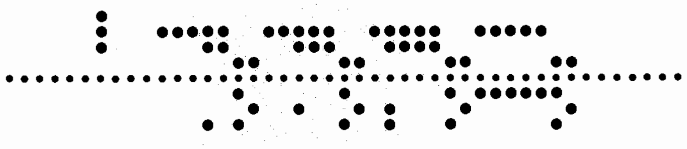


```
ΔΔΔΔΔΔΔΔΔΔΔΔΔ10.00 ΔΔΔΔΔΔΔΔΔΔΔΔ200.00 ΔΔΔΔΔΔΔΔΔΔΔ3000.00 ΔΔΔΔΔΔΔΔΔ500000.00
```

FORMAT 4F18.2 outputs the four numbers with 2 digits after the decimal point in a field of 18 characters. The last output items are the CR/LF.

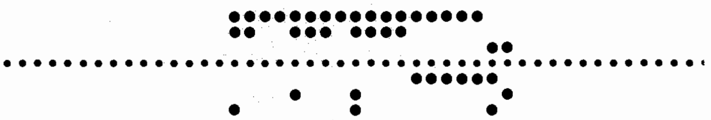
Changing the FORMAT statement (line 20) to FORMAT F5.0 gives the following output. Each value is output in a 5-character field without a decimal point. Since the entire FORMAT statement is referenced for each number, a CR/LF is output after each. Notice that the fourth number is too large for the 5-character field width and $$$$$ are output.

ΔΔΔ10 (CR) (LF) ΔΔ200 (CR) (LF) Δ3000 (CR) (LF)$$$$$ (CR)(LF)



Changing the FORMAT statement (line 20) to FORMAT 4F5.0 uses a repeat factor; only one CR/LF is output.

ΔΔΔ10ΔΔ200Δ3000$$$$$ (CR) (LF)



String Variables

Any length alphanumeric string can be punched when the PRINT format is referenced.

For example, suppose that these two strings are in memory.

A$ = BROWN,ΔE.ΔΔΔΔΔΔΔ90,95,76 (LF)

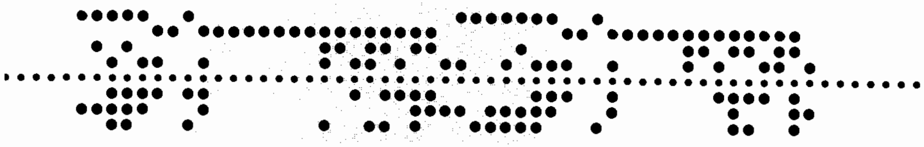
B$ = JACKSON,ΔA.ΔΔΔΔΔ80,75,87 (LF)

25 Characters

To output the entire strings on tape using PRINT format, use this line –

```
30 WRITE(9,*)A$,B$
```

BROWN,ΔE.ΔΔΔΔΔΔΔ90,95,76 (LF) JACKSON,ΔA.ΔΔΔΔΔ80,75,87 (LF)



The LIST Command

LIST [# select code ;] [first line number [; last line number]]

The LIST command outputs program lines to the tape punch specifying its select code. The optional line numbers are used to list a limited number of program lines. To output a program in memory to a tape punch responding to select code 9, execute –

```
LIST #9
```

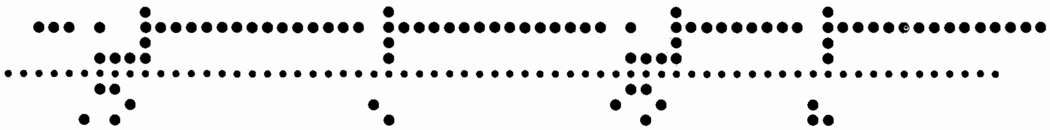
The WBYTE Function

WBYTE decimal value

This statement outputs the 16-bit, binary equivalent of each item in the list. The usable range for each expression is an integer from −32768 through 32767. Since the tape punch handles 8-bit code (normal mode) only the eight least-significant bits of each integer are punched. This reduces the usable range to integer values from −256 thru 255.

Here is a program sequence which uses WBYTE to punch 5-character data items separated by commas (decimal 44). The values for each variable are computed in lines 10 and 20. The program halts when X equals 100. A sample output is shown below.

```
10 X=Y=0
20 X=X+1
30 Y=Y+1
40 WRITE (9,*)"x",WBYTEX,"y",WBYTEY,WBYTE44
50 IF X<100 THEN 20
60 END
```



Parity Generation

A parity bit is a hole punched in one row of each character (frame) to maintain either an even or odd number of holes in each character. This extra bit is then used as a check when reading the tape, since an incorrect number of holes would indicate an error in the character. For odd parity, the controller expects to see an odd number of holes in each frame. If an even number of holes are seen, the controller halts.

The tape punch is preset at the factory to punch 8-bit (8 level) code, without a parity bit. This is called the normal mode. The tape punch can be set to punch 7-bit data and automatically add an eighth bit for either even or odd parity. The required setting is explained in the 9884A Operating and Service Manual.

Appendix

Basic Syntax
Statements, Functions and Commands

BIAND (decimal value$_1$, decimal value$_2$)

Binary AND function 30

ENTER (select code , * or line number [conversion table]) variable$_1$[, variable$_2$,...]

Enter statement 19, 27, 37, 46, 67, 118

FORMAT specification$_1$[, specification$_2$,...]

Format statement 12, 20, 68

INOR (decimal value$_1$, decimal value$_2$)

Inclusive OR function 31

LIST [# select code] [, first line number [, last line number]]

List command 25, 52, 84, 112, 127

OUTPUT (select code or string name , * or line number[, conversion table])
expression or text$_1$, [expression or text$_2$...]

Output statement 15

PRINT [any combination of text and expressions]

Print statement 12, 53, 85, 112

PTAPE select code

Ptape command 25, 39, 81, 121

RBYTE select code

Read byte function 23, 75, 120

ROT (decimal value , number of places)

Rotate (right) function 31

STAT select code

Status function 24, 56, 81, 87, 109

STDPR select code

Standard printer statement 52, 85, 108

WBYTE value

Write byte function 22, 127

WRITE (select code , * or line number) [any combination of text and expressions]

Write statement 12, 48, 67, 85, 124

ASCII Character Codes

ASCII	EQUIVALENT FORMS		
Char.	Binary	Octal	Dec
NULL	00000000	000	0
SOH	00000001	001	1
STX	00000010	002	2
ETX	00000011	003	3
EOT	00000100	004	4
ENQ	00000101	005	5
ACK	00000110	006	6
BELL	00000111	007	7
BS	00001000	010	8
HT	00001001	011	9
LF	00001010	012	10
VTAB	00001011	013	11
FF	00001100	014	12
CR	00001101	015	13
SO	00001110	016	14
SI	00001111	017	15
DLE	00010000	020	16
DC_1	00010001	021	17
DC_2	00010010	022	18
DC_3	00010011	023	19
DC_4	00010100	024	20
NAK	00010101	025	21
SYNC	00010110	026	22
ETB	00010111	027	23
CAN	00011000	030	24
EM	00011001	031	25
SUB	00011010	032	26
ESC	00011011	033	27
FS	00011100	034	28
GS	00011101	035	29
RS	00011110	036	30
US	00011111	037	31

ASCII	EQUIVALENT FORMS		
Char.	Binary	Octal	Dec
space	00100000	040	32
!	00100001	041	33
''	00100010	042	34
#	00100011	043	35
$	00100100	044	36
%	00100101	045	37
&	00100110	046	38
´	00100111	047	39
(	00101000	050	40
)	00101001	051	41
*	00101010	052	42
+	00101011	053	43
,	00101100	054	44
–	00101101	055	45
.	00101110	056	46
/	00101111	057	47
Ø	00110000	060	48
1	00110001	061	49
2	00110010	062	50
3	00110011	063	51
4	00110100	064	52
5	00110101	065	53
6	00110110	066	54
7	00110111	067	55
8	00111000	070	56
9	00111001	071	57
:	00111010	072	58
:	00111011	073	59
<	00111100	074	60
=	00111101	075	61
>	00111110	076	62
?	00111111	077	63

ASCII	EQUIVALENT FORMS		
Char.	Binary	Octal	Dec
@	01000000	100	64
A	01000001	101	65
B	01000010	102	66
C	01000011	103	67
D	01000100	104	68
E	01000101	105	69
F	01000110	106	70
G	01000111	107	71
H	01001000	110	72
I	01001001	111	73
J	01001010	112	74
K	01001011	113	75
L	01001100	114	76
M	01001101	115	77
N	01001110	116	78
O	01001111	117	79
P	01010000	120	80
Q	01010001	121	81
R	01010010	122	82
S	01010011	123	83
T	01010100	124	84
U	01010101	125	85
V	01010110	126	86
W	01010111	127	87
X	01011000	130	88
Y	01011001	131	89
Z	01011010	132	90
[	01011011	133	91
\	01011100	134	92
]	01011101	135	93
ˆ	01011110	136	94
_	01011111	137	95

ASCII	EQUIVALENT FORMS		
Char.	Binary	Octal	Dec
`	01100000	140	96
a	01100001	141	97
b	01100010	142	98
c	01100011	143	99
d	01100100	144	100
e	01100101	145	101
f	01100110	146	102
g	01100111	147	103
h	01101000	150	104
i	01101001	151	105
j	01101010	152	106
k	01101011	153	107
l	01101100	154	108
m	01101101	155	109
n	01101110	156	110
o	01101111	157	111
p	01110000	160	112
q	01110001	161	113
r	01110010	162	114
s	01110011	163	115
t	01110100	164	116
u	01110101	165	117
v	01110110	166	118
w	01110111	167	119
x	01111000	170	120
y	01111001	171	121
z	01111010	172	122
{	01111011	173	123
¦	01111100	174	124
}	01111101	175	125
~	01111110	176	126
DEL	01111111	177	127

Subject Index

a

Array Elements 26, 119
ASCII Character Codes 130

b

Bar Chart 60, 63
Basic Syntax 11
Beep 48, 74
BIAND Function 30
Binary Coding 7
Brackets 11

c

Card Reader 9869A 3,65
Carriage Return/Linefeed 12, 17
Character 11
Character Plotting 56, 99
Control Characters 66,117
Conversion Tables 26, 119
  With ENTER 27
  With OUTPUT 27

d

Delimiters 37
Digitizer 9864A 2, 43
Disk ROM 4
Dot Matrix 11

e

Edit Specs 14
EIA Codes 27
ENTER 19, 37, 67
Expression 11
External Tape Memory 9877A 103

f

Flexible Disk Drive 9885 4
FOR, with ENTER 19
FORMAT 12, 20, 68
Free-field 20
Function Codes 90

h

Hold Function 45

i

INOR Function 31
Input FORMAT Specs 20
  Edit Specs 22
I/O Expander 9878A 4
Input/Output Scheme 5
Interface 98032A 5
  Status 24, 56, 87
Interrupt 7

l

Linefeeds with Carriage Return . 12, 17
LIST 25, 84, 112

m

Matrix/Plotter ROMs 2

n

Numeric Specs 13

o

Origin Setting ... 44
OUTPUT ... 15

p

Plotting with Characters ... 56, 99
Plotters:
  9862A ... 2
  9872A ... 2
Printers:
  9866A/B ... 3, 51
  9871A ... 3, 83
  9881A ... 3, 107
Print Format ... 12, 53, 85
  Line Width ... 53, 55
PTAPE ... 25, 81, 121

r

RBYTE ... 23, 75, 120
ROT Function ... 31

s

Sample Mode ... 44, 45
Sample Rate ... 46
Select Code ... 6, 11
  HP-IB ... 7, 101
Setting the Origin ... 44
Standard Printer ... 6, 12, 52, 85
Status Function ... 24, 56, 87
Strings ... 15, 38, 115, 126
Suppressing CR/LF's ... 12, 17
Suppressing Leading Blanks ... 18
Syntax Basic ... 11

t

Tape Punch 9884A ... 2, 123
Tape Readers:
  9863A ... 1, 35
  9883A ... 1, 113
TAPE Statement ... 104
Text ... 11, 14

v

Variable ... 11

w

WRITE ... 12, 48, 67, 85
WBYTE ... 22, 127

Error Messages

ERROR	Meaning
1	Plug-in ROM missing – attempt to run a program without having the ROM installed.
2	Insufficient memory – the 9831A needs more memory than is available.
3	Statement cannot be executed from the keyboard.
4	Missing line number, or integer missing or out of range – caused by pressing STORE instead of EXEC, or by using a variable where an integer constant must be used.
5	Statement or command not recognized – caused by pressing STORE instead of EXEC, or vice versa.
6	Improper arithmetic expression. Also, missing number or expression.
7	Extra characters or parameters not allowed.
8	Missing punctuation in program statement.
9	Invalid command unless in special function key mode.
10	Special function key is undefined.
11	Exponent is out of range.
12	Two decimal points in number.
13	Sign given without number.
14	Missing comma.
15	Missing left parenthesis.
16	Missing right parenthesis.
17	Missing subscript.
18	String operation not permitted.
19	No opening quote or missing string variable.
20	No closing quote.
21	Missing or improper function name.

22 Missing function parameter.

23 Missing or incorrect DATA item.

24 Improper IF statement.

25 Missing OF in computed GOTO statement.

26 Missing variable.

27 Missing or improper FOR variable.

28 Missing TO in FOR statement.

29 Missing STEP, or illegal characters following FOR statement.

30 Missing assignment (=) operator.

31 Missing or improper assignment.

32 Improper FORMAT specification.

33 COM statement rules not followed.

34 Improper COM declaration.

35 Array or string variable is doubly dimensioned.

36 Precision of variable is doubly defined.

37 Inconsistent dimensions are given.

38 Array has unknown dimensions.

39 Dimensions are either ≤0 or too large.

40 Variable of function is undefined – often caused by using a variable which does not have a value.

41 Array or string has not been initialized – check for COM or DIM statement.

42 Array or string subscript exceeds bounds.

43 Select code out of range of 2 thru 15. The range is extended to 1 thru 15 for TAPE and 0 thru 15 for ENTER and RBYTE.

44 Line not found – often caused by incorrect branching statement.

45 Improper statement type referenced.

46 Improper statement nesting in multiline function.

47 Improper RETURN.

48 FOR statement has no matching NEXT. Also incorrect FOR nesting.

49 Out of DATA.

50 Last statement is not END.

51 LOG or LGT of negative number.

52 SQR of negative number.

53 Zero raised to zero power.

54 Non-integer power of negative number.

Tape Cartridge Errors –

55 Syntax error in tape cartridge statement.

56 Wrong file or file not found.

57 Improper operation on secured (SEC) program.

58 Tape cartridge status error: no cartridge in transport; tape is write protected (RECORD slide); external tape drive is switched off.

59 Tape verification error – tape head is dirty or tape is damaged. Also, tape position is unknown (execute REWIND or FIND to re-establish position). ERROR 59 also indicates a data verification error during PRINT# and MAT PRINT# (flexible disk) operations.

60 Incorrect file size. Also caused by an attempt to STORE DATA without an allocated memory area (COM).

61 Wrong precision or data type.

62 Wrong file type.

63 Cartridge LOAD or MERGE operation would overlay new program over old one – operation not performed.

String Variable Errors –

64 Incomplete IF statement.

65 Incorrect LEN, POS, or VAL syntax.

66 Current string length exceeded.

67 Operation is on a non-continuous string. Substring requested is beyond the logical boundary for the string and is undefined.

68 Maximum string length exceeded. Additional string length must be specified in the DIM statement.

69 Illegal DATA encountered during READ statement execution. Character data found; numeric data expected.

I/O Errors –

72 End of data reached or data contains more than ten blanks in a row.

73 Invalid FORMAT specification.

74 Numeric input has syntax error: multiple decimal points; more than one E; other non-numeric input.

75 Conversion table or code not found. Check for integer initialization in DIM or COM statement.

76 Select code does not match interface card. For example, select code without HP-IB address code addressed to HP-IB Interface, or vice versa. Also, I/O operation not allowed with select code 1 (internal tape cartridge).

77 Interface card not connected.

Flexible Disk Drive Errors –

78 I/O interrupt. For example, an interface card is plugged in while power is on.

79 All disk drives not switched on.

80 Disk drive door open.

81 Disk not installed or specified drive number not set.

82 Write-protected disk.

83 Disk drive record header error.

84 Disk track not found.

85 Disk data checksum error.

86 Disk drive hardware failure. Press RESET to regain system control.

87 Read-data error: try to reprint the data.

Flexible Disk ROM Errors –

88 Miscellaneous Disk ROM syntax error. For example, storing an incorrect IF END# statement.

89 Incorrect disk drive number or select code. Also, incorrect record pointer or word pointer.

90 Incorrect disk file name or file not found.

91 Available disk file space exceeded. Directory or availability table is full.

92 File name already exists on drive.

93 EOF (end of file) mark reached or physical end of file encountered.

94 Disk file format error. For example, a multirecord string not intact.

Recoverable Errors –

100 Numeric overflow (assumes + or − ∞).

101 Numeric underflow (assumes 0).

102 LOG or LGT of zero (assumes − ∞).

103 Division by zero (assumes + or − ∞).

104 Zero to negative power (assumes + ∞).

105 Integer variable overflow (assumes + or −32767).

106 Split variable overflow (assumes + or −9.99999E+63).

107 Split variable underflow (assumes 0).

NOTE

The machine approximates + and − infinity (∞) by 9.99999999999E+99 and −9.99999999999E+99, respectively.

Matrix Errors –

366 Matrix must be square for the attempted operation.

368 Matrix has no inverse. The data contained in the matrix does not have a solution.

369 Incompatible dimensions. Dimensions of added, subtracted, multiplied, copied, or transposed matrices must agree.

Plotter Errors –

370 Attempt to execute a 9872A Plotter operation on the 9862A Plotter.

Invalid plotter select code. 9862A requires a select code between 2 and 15. The 9872A Plotter uses a 3 or 4 digit select code with the format ccdd, where cc is a number from 1 thru 15 designating the interface select code and dd is a number from 00 thru 30 designating the device number on the HP-IB.

371 Instruction not recognized. The plotter has received an illegal character sequence. (9872A only.)

372 Wrong number of parameters. Too many or too few parameters have been sent with an instruction. (9872A only.)

373 Bad parameter. The parameters sent to the plotter with an instruction are out of range for that instruction. (9872A only.)

374 Illegal character. The character specified as a parameter is not in the allowable set for that instruction. (9872A only.)

375 Unknown character set. A character set out of the range 0 thru 4 has been designated as either the standard or alternate character set. (9872A only.)

376 Position overflow. An attempt to draw a character or perform a cplot that is located outside the plotters numeric limits. (9872A only.)

377 Transmission error. The computer has received an illegal ASCII input from the plotter. (9872A only.)

378 Attempt to scale on 9872A when P1 (lower left) is not less than P2 (upper right). (9872A only.)

379 Stop key pressed during plot execution. (9872A only.)

380 Attempt to execute an AXIS, OFFSET, PLOT, or IPLOT statement before executing a SCALE statement.

381 9862A only.

Character height specification in a LABEL statement is greater than 18.4% of the height of the plotting area.

Aspect ratio in a LABEL statement specifies a character width greater than 18.4% of the height of the plotting area.

The X or Y parameter in a CPLOT statement requires a pen movement greater than 18.4% of the height of the plotting area.

382 Attempt to execute an AXIS statement with the specified start point outside of the plotting area.

Attempt to execute an AXIS statement with the tic mark spacing too small (i.e., space between tics is less than 1/9999 of the maximum width or height of the plotting area). (9862A only.)

HEWLETT PACKARD

SALES & SERVICE OFFICES

AFRICA, ASIA, AUSTRALIA

ANGOLA
Telectra
Empresa Técnica de
Equipamentos
Eléctricos, S.A R.L
R. Barbosa Rodrigues, 42-I°OT.
Caixa Postal, 6487
Luanda
Tel: 35515/6
Cable: TELECTRA Luanda

AUSTRALIA
Hewlett-Packard Australia
Pty. Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
P.O. Box 36
Doncaster East, Victoria 3109
Tel: 89-6351
Telex: 31-024
Cable: HEWPARD Melbourne

Hewlett-Packard Australia
Pty. Ltd.
31 Bridge Street
Pymble
New South Wales, 2073
Tel: 449-6566
Telex: 21561
Cable: HEWPARD Sydney

Hewlett-Packard Australia
Pty. Ltd.
153 Greenhill Road
Parkside, S.A., 5063
Tel: 272-5911
Telex: 82536 ADEL
Cable: HEWPARD ADELAIDe

Hewlett-Packard Australia
Pty.Ltd.
141 Stirling Highway
Nedlands, W.A. 6009
Tel: 86-5455
Telex: 93859 PERTH
Cable: HEWPARD PERTH

Hewlett-Packard Australia
Pty. Ltd.
121 Wollongong Street
Fyshwick, A.C.T. 2609
Tel: 95-2733
Telex: 62650 Canberra
Cable: HEWPARD CANBERRA

Hewlett Packard Australia
Pty. Ltd.
5th Floor
Teachers Union Building
495-499 Boundary Street
Spring Hill, 4000 Queensland
Tel: 229-1544
Cable: HEWPARD Brisbane

GUAM
Medical/Pocket Calculators Only
Guam Medical Supply, Inc.
Jay Ease Building, Room 210
P.O. Box 8947
Tamuning 96911
Tel: 646-4513
Cable: EARMED Guam

HONG KONG
Schmidt & Co (Hong Kong) Ltd
P.O. Box 297
Connaught Centre
39th Floor
Connaught Road, Central
Hong Kong
Tel: H-255291-5
Telex: 74766 SCHMC HX
Cable: SCHMIDTCO Hong Kong

INDIA
Blue Star Ltd
Kasturi Buildings
Jamshedji Tata Rd.
Bombay 400 020
Tel: 29 50 21
Telex: 2156
Cable: BLUEFROST

Blue Star Ltd
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
Bombay 400 025
Tel: 45 78 87
Telex: 4093
Cable: FROSTBLUE

Blue Star Ltd
Band Box House
Prabhadevi
Bombay 400 025
Tel: 45 73 01
Telex: 3751
Cable: BLUESTAR

Blue Star Ltd.
14/40 Civil Lines
Kanpur 208 001
Tel: 6 88 82
Telex: 292
Cable: BLUESTAR

Blue Star Ltd
7 Hare Street
P.O. Box 506
Calcutta 700 001
Tel: 23-0131
Telex: 7655
Cable: BLUESTAR

Blue Star Ltd
7th & 8th Floor
Bhandari House
91 Nehru Place
New Delhi 110024
Tel: 634770 & 635166
Telex: 2463
Cable: BLUESTAR

Blue Star Ltd.
Blue Star House
11/11A Magarath Road
Bangalore 560 025
Tel: 55668
Telex: 430
Cable: BLUESTAR

Blue Star Ltd.
Meeakshi Mandiran
xxx/1678 Mahatma Gandhi Rd
Cochin 682 016
Tel: 32069, 32161, 32282
Telex: 046-514
Cable: BLUESTAR

Blue Star Ltd
1-1-117/1
Sarojini Devi Road
Secunderabad 500 003
Tel: 70126, 70127
Cable: BLUEFROST
Telex: 459

Blue Star Ltd
2/34 Kodambakkan High Road
Madras 600034
Tel: 82056
Telex: 041-379
Cable: BLUESTAR

Blue Star Ltd.
Nathraj Mansions
2nd Floor Bistupur
Jamshedpur 831 001
Tel: 7383
Cable: BLUESTAR
Telex: 240

INDONESIA
BERCA Indonesia P.T
P.O. Box 496
1st Floor JL. Cikini Raya 61
Jakarta
Tel: 56038, 40369, 49886
Telex: 42895
Cable: BERCACON

BERCA Indonesia P.T
63 JL. Raya Gubeng
Surabaya
Tel: 44309

ISRAEL
Electronics & Engineering Div
of Motorola Israel Ltd.
17, Kremenetski Street
P.O. Box 25016
Tel-Aviv
Tel: 38973
Telex: 33569
Cable: BASTEL Tel-Aviv

JAPAN
Yokogawa-Hewlett-Packard Ltd
Ohashi Building
59-1 Yoyogi 1-Chome
Shibuya-ku, **Tokyo** 151
Tel: 03-370-2281/92
Telex: 232-2024YHP
Cable: YHPMARKET TOK 23-724

Yokogawa-Hewlett-Packard Ltd
Seiko Ibaraki Building
2-8 Kasuga 2-chome, Ibaraki-shi
Osaka,567
Tel: (0726) 23-1641
Telex: 5332-385 YHP OSAKA

Yokogawa-Hewlett-Packard Ltd
Nakamo Building
24 Kami Sasajima-cho
Nakamura-ku, **Nagoya**, 450
Tel: (052) 571-5171

Yokogawa-Hewlett-Packard Ltd
Tanigawa Building
2-24-1 Tsuruya-cho
Kanagawa-ku
Yokohama, 221
Tel: 045-312-1252
Telex: 382-3204 YHP YOK

Yokogawa-Hewlett-Packard Ltd
Mito Mitsu Building
105, Chome-1,San-no-maru
Mito, Ibaragi 310
Tel.: 0292-25-7470

Yokogawa-Hewlett-Packard Ltd.
Inoue Building
1348-3, Asahi-cho, 1-chome
Atsugi, Kanagawa 243
Tel: 0462-24-0452

Yokogawa-Hewlett-Packard Ltd
Kimura Building
3rd Floor 20
2-chome, Tsukuba
Kumagaya, Saitama 360
Tel: 0485-24-6563

KENYA
Technical Engineering Services
(E.A.)Ltd..
P.O. Box 18311
Nairobi
Tel: 557726/556762
Cable: PROTON

Medical Only
International Aeradio(E.A.)Ltd..
P.O. Box 19012
Nairobi Airport
Nairobi
Tel: 336055/56
Telex: 22201/22301
Cable: INTAERID Nairobi

KOREA
Samsung Electronics Co., Ltd.
20th Fl. Dongbang Bldg.250.2-KA
C.P.O. Box 2775
Taepyung-Ro, Chung-Ku
Seoul
Tel: (23) 6811
Telex: 22575
Cable: ELEKSTAR Seoul

MALAYSIA
Teknik Mutu Sdn. Bhd.
2 Lorong 13/6A
Section 13
Petaling Jaya,**Selangor**
Tel: 54994/54916
Telex: MA 37605

Protel Engineering
P.O. Box 1917
Lot 259, Satok Road
Kuching, **Sarawak**
Tel: 2400
Cable: PROTEL ENG

MOZAMBIQUE
A.N. Goncalves, Lta.
162, 1 Apt. 14 Av. D. Luis
Caixa Postal 107
Lourenco Marques
Tel: 27091, 27114
Telex: 6-203 NEGON Mo
Cable: NEGON

NEW ZEALAND
Hewlett-Packard (N.Z.) Ltd.
P.O. Box 9443
Courtney Place
Wellington
Tel: 877-199
Telex: NZ 3839
Cable: HEWPACK Wellington

Hewlett-Packard (N.Z.) Ltd.
Pakuranga Professional Centre
267 Pakuranga Highway
Box 51092
Pakuranga
Tel: 569-651
Telex: NZ 3839
Cable: HEWPACK.Auckland

Analytical/Medical Only
Medical Supplies N.Z. Ltd.
Scientific Division
79 Carlton Gore Rd., Newmarket
P.O. Box 1234
Auckland
Tel: 75-289
Telex: 2958 MEDISUP
Cable: DENTAL Auckland

Analytical/Medical Only
Medical Supplies N.Z. Ltd.
P.O. Box 1994
147-161 Tory St.
Wallington
Tel: 850-799
Telex: 3858
Cable: DENTAL, Wellington

Analytical/Medical Only
Medical Supplies N.Z. Ltd.
P.D. Box 309
239 Stanmore Road
Christchurch
Tel: 892-019
Cable: DENTAL, Christchurch

Analytical/Medical Only
Medical Supplies N.Z. Ltd
303 Great King Street
P.O. Box 233
Dunedin
Tel: 88-817
Cable: DENTAL, Dunedin

NIGERIA
The Electronics
Instrumentations Ltd
N6B/770 Oyo Road
Oluseun House
P.M.B. 5402
Ibadan
Tel: 61577
Telex: 31231 TEIL Nigeria
Cable: THETEIL Ibadan

The Electronics Instrumentations Ltd
144 Agege Motor Road, Mushin
P.O. Box 6645
Lagos
Cable: THETEIL Lagos

PAKISTAN
Mushko & Company, Ltd.
Oosman Chambers
Abdullah Haroon Road
Karachi-3
Tel: 511027, 512927
Telex: KR894
Cable: COOPERATOR Karachi

Mushko & Company, Ltd
38B, Satellite Town
Rawalpindi
Tel: 41924
Cable: FEMUS Rawalpindi

PHILIPPINES
The Online Advanced Systems
Corporation
6th Floor, Yujui Co. Bldg
560 Quintin Paredes St
Binondo, **Manila**
Tel: 48-71-49, 48-68-53
In Makati: 85-35-81, 85-34-91
Telex: 3274 ONLINE

RHODESIA
Field Technical Sales
45 Kelvin Road North
P.O. Box 3458
Salisbury
Tel: 705231 (5 lines)
Telex: RH 4122

SINGAPORE
Hewlett-Packard Singapore
(Pte.) Ltd.
3841F, Jalan Bukit Merah
Blk. 2, 6th Floor, Jalan
Redhill Industrial Estate
Alexandra P.O. Box 58,
Singapore 3
Tel: 633022
Telex: HPSG RS 21486
Cable: HEWPACK, Singapore

SOUTH AFRICA
Hewlett-Packard South Africa
(Pty.), Ltd.
Private Bag Wendywood
Sandton, Transvaal 2144
Hewlett-Packard Center
Daphne Street, Wendywood,
Sandton, Transvaal 2144
Tel: 802-104018
Telex: 8-4782
Cable: HEWPACK JOHANNESBURG

Service Department
Hewlett-Packard South Africa
(Pty.), Ltd.
P.O. Box 39325
Gramley, Sandton, 2018
451 Wynberg Extension 3,
Sandton, 2001
Tel: 636-8188/9
Telex: 8-2391

Hewlett-Packard South Africa
(Pty.), Ltd
P.O. Box 120
Howard Place, Cape Province, 7450
Pine Park Center, Forest Drive,
Pinelands, Cape Province, 7405
Tel: 53-7955 thu 9
Telex: 57-0006

Service Department
Hewlett-Packard South Africa
(Pty.), Ltd
P.O. Box 37099
Overport, Durban 4067
Braby House
641 Ridge Road
Durban, 4001
Tel: 88-7478
Telex: 6-7954

TAIWAN
Hewlett-Packard Far East Ltd.,
Taiwan Branch
39 Chung Shiao West Road
Sec. 1, 7th Floor
Taipei
Tel: 3819160-4
Telex: 21824 HEWPACK
Cable: HEWPACK TAIPEI

Hewlett-Packard Far East Ltd.
Taiwan Branch
68-2, Chung Cheng 3rd. Road
Kaohsiung
Tel: (07) 242318-Kaohsiung

Analytical Only
San Kwang Instruments Co., Ltd.,
No. 20, Yung Sui Road
Taipei
Tel: 3715171-4 (5 lines)
Telex: 22894 SANKWANG
Cable: SANKWANG TAIPEI

TANZANIA
Medical Only
International Aeradio (E.A.), Ltd.
P.O. Box 861
Dar as Salaam
Tel: 21251 Ext. 265
Telex: 41030

THAILAND
UNIMESA Co., Ltd.
Elcom Research Building
Bangjak Sukumvit Ave.
Bangkok
Tel: 932387, 930338
Cable: UNIMESA Bangkok

UGANDA
Medical Only
International Aeradio(E.A.), Ltd..
P.O. Box 2577
Kampala
Tel: 54388
Cable: INTAERIO Kampala

ZAMBIA
R.J. Tilbury (Zambia) Ltd
P.O. Box 2792
Lusaka
Tel: 73793
Cable: ARJAYTEE, Lusaka

OTHER AREAS NOT LISTED, CONTACT:
Hewlett-Packard Intercontinental
3200 Hillview Ave.
Palo Alto, California 94304
Tel: (415) 493-1501
TWX: 910-373-1267
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

CANADA

ALBERTA
Hewlett-Packard (Canada) Ltd
11620A - 168 Street
EdmontonT5M 3T9
Tel: (403) 452-3670
TWX: 610-831-2431 EDTH

Hewlett-Packard (Canada) Ltd
915-42 Ave S.E. Suite 102
Calgary T2G 1Z1
Tel: (403) 287-1672
Twx: 610-821-6141

BRITISH COLUMBIA
Hewlett-Packard (Canada) Ltd
837 E. Cordova Street
Vancouver V6A 3R2
Tel: (604) 254-0531
TWX: 610-922-5059 VCR

MANITOBA
Hewlett-Packard (Canada) Ltd
513 Century St
St. James
Winnipeg R3H OL8
Tel: (204) 786-7581
TWX: 610-671-3531

NOVA SCOTIA
Hewlett-Packard (Canada) Ltd
800 Windmill Road
P.O. Box 9331
Dartmouth B2Y 3Z6
Tel: (902) 469-7820
TWX: 610-271-4482 HFX

ONTARIO
Hewlett-Packard (Canada) Ltd
1020 Morrison Dr
Ottawa K2H 8K7
Tel: (613) 820-6483
TWX: 610-513-1636

Hewlett-Packard (Canada) Ltd
6877 Goreway Drive
Mississauga L4V 1M8
Tel: (416) 678-9430
TWX: 610-492-4246

QUEBEC
Hewlett-Packard (Canada) Ltd
275 Hymus Blvd
Pointe Claire H9R 1G7
Tel: (514) 697-4232
TWX: 610-422-3022
TLX: 05-821521 HPCL

FOR CANADIAN AREAS NOT LISTED:
Contact Hewlett-Packard (Canada)
Ltd. in Mississauga

CENTRAL AND SOUTH AMERICA

ARGENTINA
Hewlett-Packard Argentina
S.A.
Av. Leandro N. Alem 822 - 12
1001**Buenos Aires**
Tel: 31-6063,4,5,6 and 7
Telex: Public Booth Nº 9
Cable: HEWPACK ARG

BOLIVIA
Stambuk & Mark (Bolivia) Ltda.
Av. Mariscal, Santa Cruz 1342
La Paz
Tel: 40626, 53163, 52421
Telex: 3560014
Cable: BUKMAR

BRAZIL
Hewlett-Packard do Brasil
I.e.C. Ltda.
Avenida Rio Negro, 980
Alphaville
06400 Barueri **Seo Paulo**
Tel: 429-2148/9;429-2118/9

Hewlett-Packard do Brasil
I.e.C. Ltda.
Rua Padre Chagas, 32
90000-**Pôrto Alegre**-RS
Tel: (0512) 22-2998, 22-5621
Cable: HEWPACK porto Alegre

Hewlett-Packard do Brasil
I.E.C. Lttda.
Rua Siqueira Campos, 53
Copacabana
20000-**Rio de Janeiro**
Tel: 257-80-94-DOD (021)
Telex: 391-212-1905 HEWP-BR
Cable: HEWPACK
Rio de Janeiro

CHILE
Calcagni y Metcalfe Ltda
Alameda 580-Of. 807
Casilla 2118
Santiago, 1
Tel: 398613
Telex: 3520001 CALMET
Cable: CALMET Santiago

COLOMBIA
Instrumentacidn
Henrik A. Langebaek & Kier S.A.
Carrera 7 No. 48-75
Apartado Aéreo 6287
Bogoté, I.D.E.
Tel: 69-88-77
Cable: AARIS Bogotá
Telex: 044-400

COSTA RICA
Cientifica Costarricense S.A.
Calle Central, Avenidas 1 y 3
Apartado 10159
San José
Tel: 21-86-13
Cable: GALGUR San José

ECUADOR
Medical Only
A.F. Viscaíno Compañia Ltda
Av. Rio Amazonas No. 239
P.O. Box 2925
Quito
Tel: 242-150,247-033/034
Cable: Astor Quito

Calculators Only
Computadoras y Equipos
Electrdnicos
P.O. Box 2695
990 Toledo (y Cordero)
Quito
Tel: 525-982
Telex: 02-2113 Sagita Ed
Cable: Sagita-Quito

EL SALVADOR
Instrumentacion y Procesamiento
Electronico de el Salvador
Bulevar de los Heroes II-48
San Salvador
Tel: 252787

GUATEMALA
IPESA
Avenida La Reforma 3-48,
Zona 9
Guatemala City
Tel: 63627, 64786
Telex: 4192 Teletro Gu

MEXICO
Hewlett-Packard Mexicana,
S.A. de C.V.
Torres Adalid No. 21, 11 Piso
Col. del Valle
Mexico 12, D.F
Tel: (905) 543-42-32
Telex: 017-74-507

Hewlett-Packard Mexicana,
S.A. de C.V.
Ave. Constitucidn No. 2184
Monterrey, N.L.
Tel: 48-71-32, 48-71-84
Telex: 038-843

NICARAGUA
Roberto Terán G.
Apartado Postal 689
Edificio Terán
Manague
Tel: 25114, 23412,23454
Cable: ROTERAN Managua

PANAMA
Electrdnico Balboa, S.A.
P.O. Box 4929
Calle Samuel Lewis
Cuidad de Panama
Tel: 64-2700
Telex: 3431103 Curunda,
Canal Zone
Cable: ELECTRON Panama

PARAGUAY
Z.J. Melamed S.R.L
Oivisidn: Aparatos y Equipos
Médicos
Oivisidn: Aparatos y Equipos
Cientlficos y de Investigacidn
P.O Box 676
Chile-482, Edificio Victoria
Asunción
Tel: 91-271, 91-272
Cable: RAMEL

PERU
Compañia Electro Médica S.A.
Los Flamencos 145
San Isidro Casilla 1030
Lima 1
Tel: 41-4325
Cable: ELMEO Lima

PUERTO RICO
Hewlett-Packard Inter-Americas
Puerto Rico Branch Office
Calle 272.
No. 203 Urb. Country Club
Carolina 00924
Tel: (809) 762-7255
Telex: 345 0514

URUGUAY
Pablo Ferrando S.A.
Comercial e Industrial
Avenida Italia 2877
Casilla de Correo 370
Montevideo
Tel: 40-3102
Cable: RADIUM Montevideo

VENEZUELA
Hewlett-Packard de Venezuela
C.A.
P.O. Box 50933
Caracas 105
Los Ruices Norte
3a Transversal
Edificio Segre
Caracas 107
Tel: 35-00-11 (20 lines)
Telex: 25146 HEWPACK
Cable: HEWPACK Caracas

FOR AREAS NOT LISTED, CONTACT:
Hewlett-Packard
Inter-Americas
3200 Hillview Ave
Palo Alto, California 94304
Tel: (415) 493-1501
TWX: 910-373-1260
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

HEWLETT hp PACKARD

PART NO. 09831-90020
MICROFICHE NO. 09831-99020

PRINTED IN U.S.A.
April 1, 1977

Manual Changes

HP 9831A Desktop Computer
Peripheral Control Manual

(For Manual P/N 09831-90020, Dated April 1, 1977)

Page 30:

Please correct the second line in the paragraph titled "Binary Functions".

The usable range of each value is from −32767 thru 32767.

Page 48:

Delete page 48.

The digitizer tone is not accessible with the standard 9831A. The firmware to control this feature is contained in the 98218A Flexible Disk ROM. If you have this ROM, the following syntax will sound the tone.

WCTL select code ; 0
WCTL select code ; 1

November 1977
Supplement A to 09831-90020