Hewlett-Packard System 45 Desktop Computer

Programmer's Introduction

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

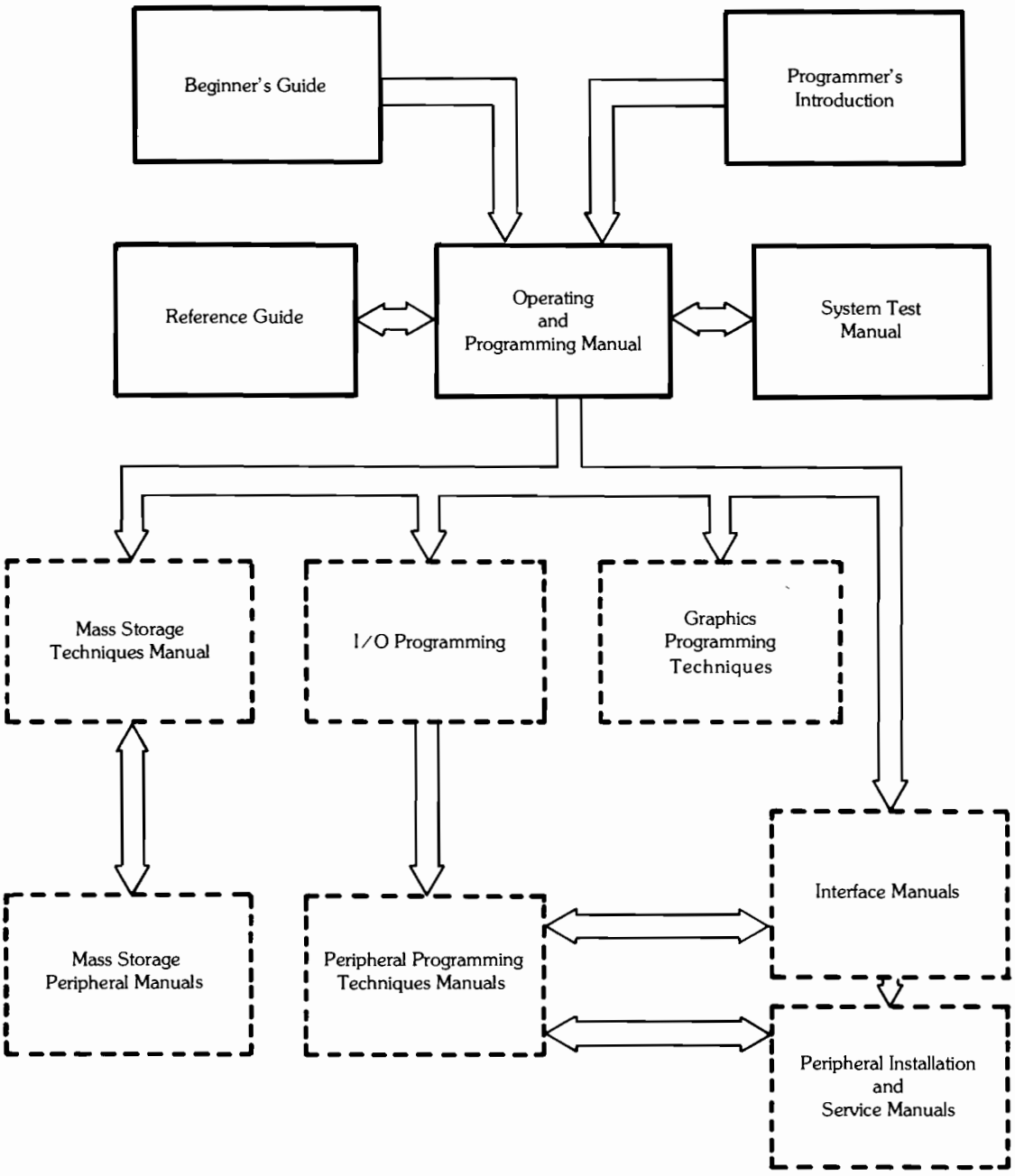
Programmer's Introduction

HP System 45 Desktop Computer

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

Table of Contents

System 45 Documentation



Preface

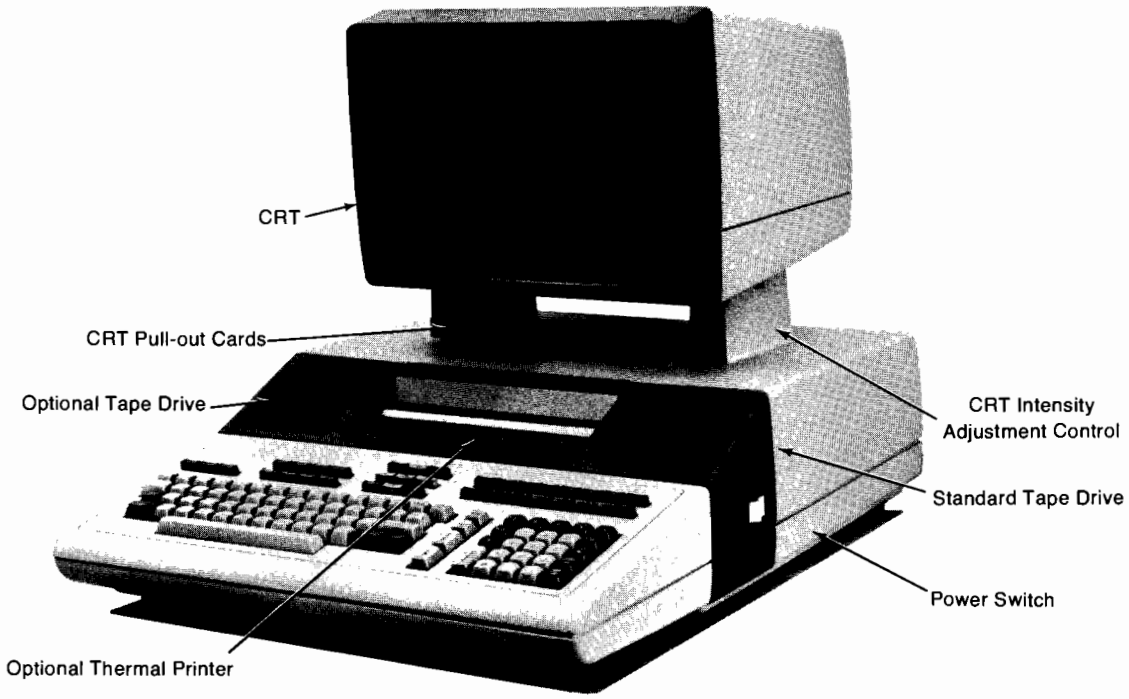
The Programmer's Introduction is a preview of the System 45 to acquaint you with the computer's unique features and capabilities that will help you in your programming applications.

Because it is a preview, this manual contains brief discussions and examples emphasizing the above-mentioned features and capabilities. For more information, you should refer to the Operating and Programming Manual which is a comprehensive reference for the System 45.

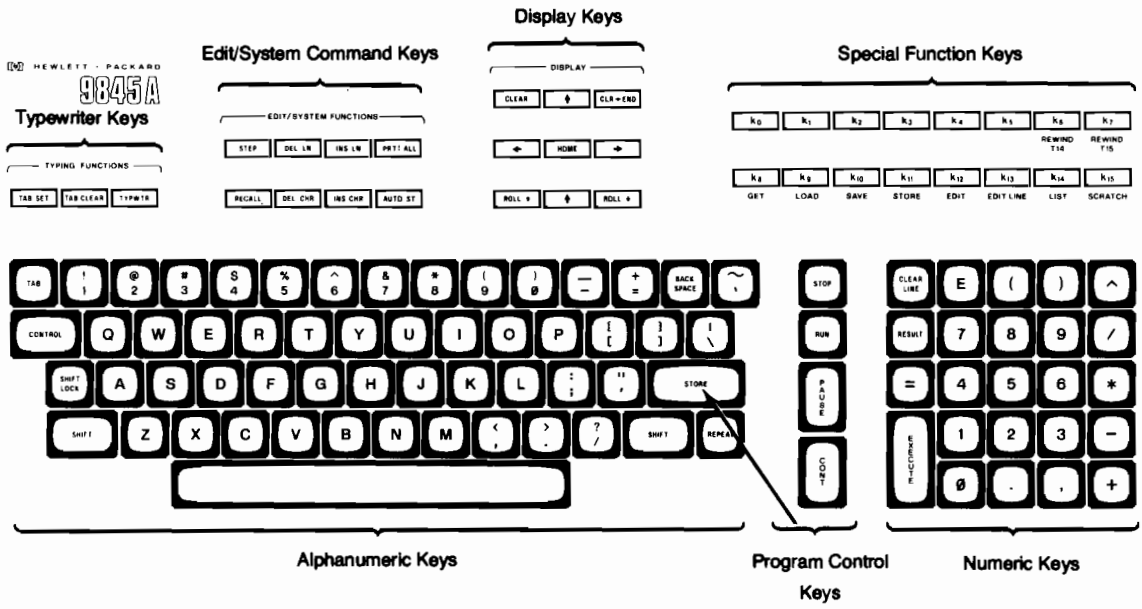
Those who are inexperienced in programming or would like to review fundamental programming principles should refer to the Beginner's Guide. This manual teaches flowcharting, discusses the System 45 operating procedures and provides tutorial aid through the use of problems and corresponding solutions.

To help you determine which manual you need at any given point, the System 45 documentation scheme and suggested progression is block diagrammed on the opposite page. The dotted-line borders indicate manuals that are shipped with specific options. Solid borders indicate those manuals that are provided with each standard System 45.

After you finish reading this manual, please complete and return the postage-paid comments card located in the back. Your opinions and comments will help us evaluate the manual's direction and effectiveness and thereby improve future manuals.



System 45



Keyboard

Chapter 1
System 45 Overview

The System 45, shown in the photograph on the opposite page, includes the CRT and two tape cartridge drives (the drive located on the righthand side is standard, the one on the lefthand side is optional). Also shown is the built-in thermal line printer which is offered as an option. In program examples throughout this manual, it is assumed that the thermal line printer is included, although several types of HP printers can be used with the System 45. The Operating and Programming Manual describes these printers and other peripherals that can be interfaced to the System 45.

General Keyboard

The System 45 keyboard comprises seven blocks of keys (see lower photograph, opposite page). The alphanumeric block is basically the same as a standard typewriter keyboard, except all output is in "reverse typewriter mode"—letters are upper case unless the SHIFT key is pressed, in which case, they are output lower case.

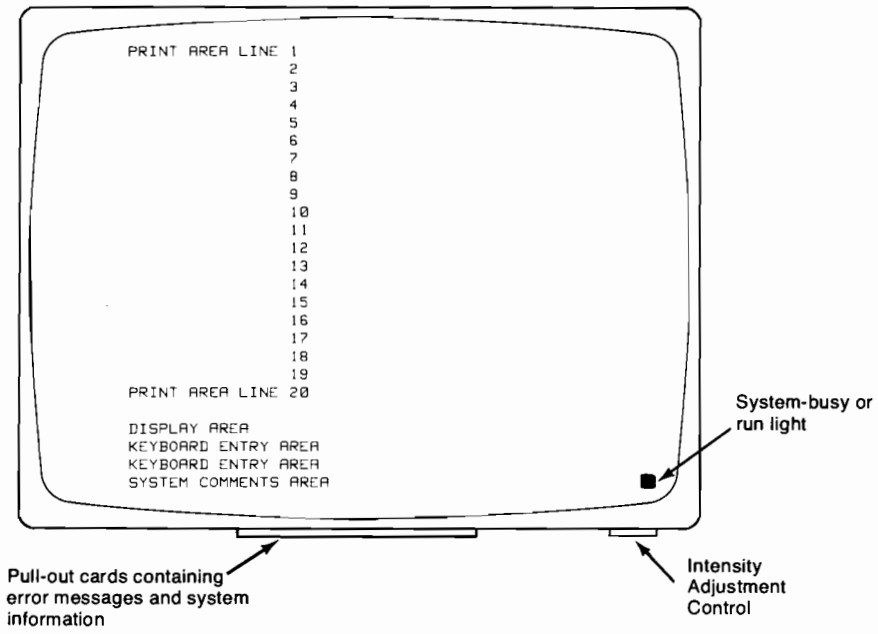
The System 45 can be switched to regular typewriter mode for applications such as text processing by pressing the TYPWTR key (located in the typewriter key block). The TAB SET and TAB CLR keys also situated in that block operate the same as tabulation keys on the standard typewriter.

Numbers and relational operators in the numeric key block perform the same functions as the corresponding keys on the alphanumeric block and are also interchangeable. You can, for example, enter a 2 from the numeric block, a ∧ from the alphanumeric block, a 3 from the alphanumeric block and press EXEC. The answer is the same as if you had made entries exclusively from one block.

The remaining blocks of keys are introduced and explained in the following chapters of this manual.

CRT Display

The System 45 CRT is a 24-line display capable of containing up to 80 characters per line. The illustration below, showing a line-by-line breakdown of the CRT, contains 25 lines; line 21 is a blank and serves as a display separator only, leaving 24 functional lines.



CRT

After you have checked to make sure the System 45 is properly connected, turn it on by setting the switch, located on the righthand side, to "1".

A small flashing line, referred to as a cursor, will appear on the lower lefthand side of the screen (line 23).

The following program is included here to demonstrate the CRT and help familiarize you with the keyboard. Before you enter this program,

- press the TYPWTR key (TYPWTR mode is indicated on the righthand side of the CRT)
- type in EDIT LINE,
- press the gold-colored EXEC key,

The System 45 is now in **edit line mode**. Type in the following program lines, pressing the gold-colored [STORE] key after each line is entered. (The REMark and comment delimiter statements, introduced below, don't affect a program and need not be entered.)

```
10   DISP "This is Hewlett-Packard's System 45."
20   REM          DISP displays that in quotes
30   REM          REM represents REMark; used for easy-to-follow documentation
40                ! The ! is a comment delimiter; does the same thing as REM
50    BEEP
60    WAIT 2100    ! Notice ! is used after statement as well as after
                     the line number.
70    BEEP
80    DISP "Lines 50 and 70 cause the computer to BEEP;"
90    WAIT 2100
100   DISP "lines 60 and 90 provide for a 2100-millisecond delay."
110               ! After line 120 is STOREd, press these keys:
                    LIST , EXECUTE and RUN.
120   END         ! Pressing LIST and EXECUTE returns the remaining
                    bytes of memory, available to you.
```

Edit Line and Auto Modes

If you prefer to alter the line numbering system that starts at 10 and increments by 10 automatically, type in EDIT LINE, the starting line and the incremental value and press EXECUTE. For example typing in and executing EDIT LINE 100, 20 would result in line numbers 100, 120, 140 and so forth.

Another way to enter program lines is by typing in AUTO (for automatic line numbering) and pressing [EXECUTE], which displays a 10 on line 23 of the CRT. With AUTO, however, only one program line at a time is displayed (unless the [PRT ALL] key, located in the edit/system functions block, is latched[1]). Another disadvantage is that the AUTO mode does not permit full use of the display keys for editing programs.

Display Control

When the CRT is in edit line mode, the display keys let you move lines up or down, move the cursor left or right and also eliminate whole or partial lines or the entire display. For example, press the [CLEAR] key; the previous program disappears from the display but remains in read/write memory.

1 [PRT ALL] outputs all operations as they are performed and is particularly useful in debugging procedures, discussed in a subsequent chapter.

To return the program to the CRT and into the edit line mode, execute EDIT LINE.

Press ↓ to move program lines down one at a time—the flashing cursor is positioned at the end of the line currently under edit line control. Pressing ↑ scrolls up one line at a time, → moves the cursor one character to the right and ← moves the cursor one position to the left. Firmly pressing an arrow key causes repetition of that direction. For a longer program, you can scroll five lines at a time, up or down, in edit line mode, by pressing ROLL↓ or ROLL↑, respectively.

HOME moves the cursor to the far-left character position, and the CLR→END key clears that part of a line to the right of the cursor position.

CLEAR LINE, located in the numeric key block, eliminates a line from the CRT no matter where the cursor is positioned in that particular line; however, the line still remains in memory.

Editing Capability

Several keys in the edit/system command block also let you edit programs. To delete a program line in edit line mode, for example, press the DEL LN key. To insert a line, scroll to the line that **follows** the location of the line you want to add and press the INS LN key.

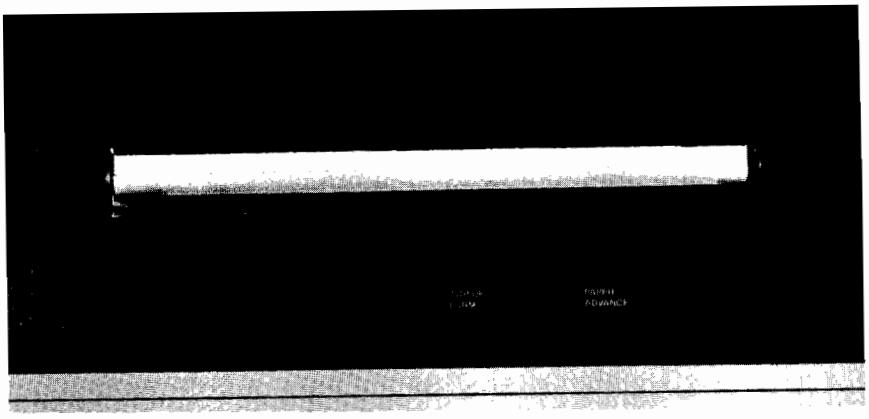
To insert a character, press the INS CHR key. An inverse video, rectangular cursor envelops the character previously indicated by the flashing line cursor. You then insert a character or characters to the **left** of the cursor and press INS CHR a second time to exit that mode. To delete a character, move the line cursor to that character and press DEL CHR.

Thermal Line Printer

A closeup of the optional internal printer is shown in the photograph below with explanations of the paper-control keys.

Top of Form Key advances unperforated paper 12 inches at one time; perforated paper is advanced to the top margin of each form

Paper Advance Key moves paper forward; to stop advance, release key



To access this printer, type in PRINTER IS 0 and press [EXEC]. Anything printed thereafter is output to the internal printer until another printer is accessed.

The number 0 is a select code and is explained in more detail in Chapter 1, "Owners Information", of the Operating and Programming Manual. At power on, the standard printer is automatically set, or defaulted, to the CRT (select code 16).

PRINTER IS statements can also be incorporated into a program. For demonstration purposes, enter the following program after you've executed SCRATCH A to erase the previous program from memory.

(Incidentally, there are several types of SCRATCH commands to eliminate different parts of memory. See Chapter 4, "Programming Information", in the Operating and Programming Manual for the list.)

```
10   PRINTER IS 0
20     J=.08
30      FOR I=1 TO 30
40      PRINT I;FNX(I,J),                ! FNX references user-defined function
50      NEXT I
60   DEF FNX(A,B)=A*(1+A)^B/((1+A)^B-1) ! DEFines single-line function X
70   END
```

[RUN]

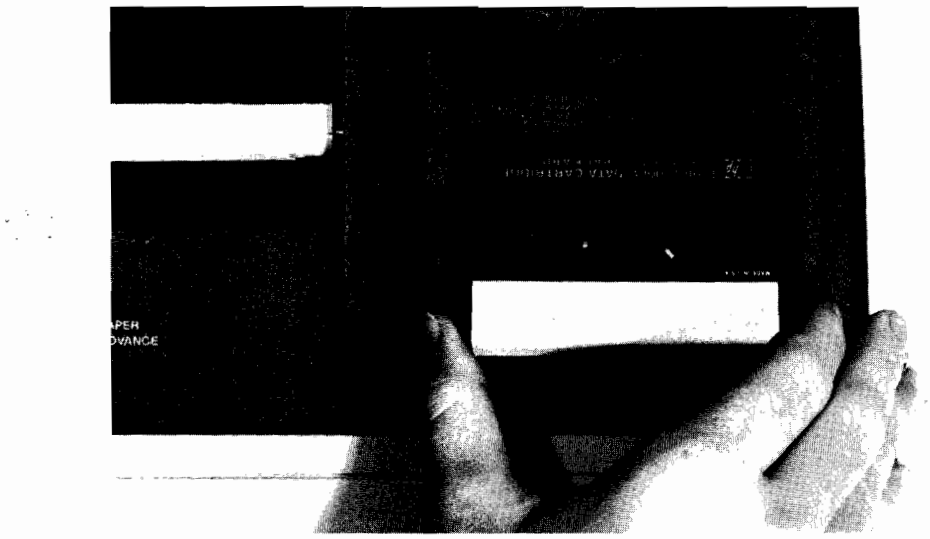
```
 1  18.5383087529    2  23.7706269544    3  28.5782522202    4  33.1096532349
 5  37.4416190334    6  41.6201806615    7  45.6756012482    8  49.6290863472
 9  53.4962062635   10  57.288810655    11  61.0161771846   12  64.6857388866
13  68.3035655941   14  71.874694152    15  75.4033618403   16  78.8931757523
17  82.3472383852   18  85.7682425466   19  89.1585443413   20  92.5202202333
21  95.8551120117   22  99.1648629004   23  102.450946782   24  105.714692019
25  108.957301118   26  112.179867117   27  115.38338725    28  118.568774522
29  121.7368675     30  124.888438695
```

Tape Drive and Mass Storage Structure

Before you store a program on a tape cartridge, or on any System 45 mass storage medium for that matter, you must first initialize it. This not only erases data and programs on a used medium but also establishes records, tables and directories on new and used media.

For the standard tape drive and cartridge, initialize by:

- sliding the RECORD tab on the cartridge in the direction of the arrow,
- inserting the cartridge as shown in the photograph below,
- typing in and executing `INITIALIZE ":T15"`



The initialize command contains the mass storage unit specifier (**msus**) which addresses the mass storage device. In this case the msus is `":T15"` where `T` is the device type and `15` is the select code. If you are using the optional tape drive, the msus is `":T14"`.

After the tape stops moving and the run light disappears from the CRT, retype the previous program or `LIST` it to make sure you haven't erased it, then type in `STORE "MYPROG"` and press EXEC.

Any ASCII character except `"`, `:`, a blank and the ASCII null can be used in a file name, but it must not exceed six characters in length.

When the tape once again stops moving, execute the command CAT (which represents catalog). What then appears on the CRT or hard-copy printout is a tape file directory, shown below.

For a complete explanation of the directory contents, see Chapter 2, "Getting Started", of the Mass Storage Techniques Manual.

To rewind the standard tape cartridge, type in REWIND ":T15" and press EXEC. You can also press Special Function key k7 if it contains its power-on definition of REWIND ":T15". (Special Function keys are discussed in Chapter 2 of this manual.)

The following table briefly explains some mass storage and retrieval commands, including STORE and LOAD.

Command	File Type	File-type Abbreviation	Explanation
STORE	program	PROG	stores program from memory
LOAD			retrieves program and loads into computer memory
RE-STORE			replaces program with revised version
SAVE	data	DATA	saves program from memory as string data
GET			retrieves saved programs and copies into computer memory
LINK			retrieves and copies saved program, retaining current values of variables
RE-SAVE			replaces program with revised version
			(STORE and LOAD are faster than SAVE and GET)
STORE KEY	key	KEYS	stores definitions of Special Function keys
LOAD KEY			retrieves and loads definitions
STORE ALL	storeall	ALL	stores entire read/write memory
LOAD ALL			retrieves and loads memory
STORE BIN	binary	BPRG	stores binary routines
LOAD BIN			retrieves and loads binary routines

Other commands enable you to RENAME files, COPY files from one medium to another, PURGE or eliminate files and PROTECT files against accidental erasure. For DATA files you can CREATE specifically defined records, ASSIGN files to file numbers and read and write in either serial or random mode with READ # and PRINT #.

The System 45 incorporates a unified mass storage structure which, simply stated, means that you write to other media such as flexible disks and fixed discs with the same set of previously mentioned commands. For example, if your larger storage device is an HP 9885M Flexible Disk Drive (and assuming you had the proper interface and ROM), to store the previous program, you would either execute MASS STORAGE IS ":F", where F is the device type for the flexible disk and execute STORE "MYPROG" or type in STORE "MYPROG: F" and press EXEC.

Detailed information concerning all mass storage commands and capabilities of the System 45 is available in the Mass Storage Techniques Manual.

Chapter 2
Language and Keyboard Highlights

The System 45's BASIC language is designed so that you can use multicharacter variables in programs. This means that in the following program, for example, .01 can be assigned to Payperday instead of a single-character variable such as P. Such designation of understandable names to variables simplifies debugging procedures for programmers and also helps clarify programs.

```
10   Payperday=.01  ! Variable names can't be longer than 15 characters.
20       FOR Day=1 TO 30
30       PRINT "If this is day ";Day;" you owe me $";Payperday
40       Accumulate=Accumulate+Payperday
50                  ! Numeric variables are preinitialized to zero.
60       Payperday=Payperday*2
70       NEXT Day
80   PRINT LIN(2),"This is the total --$";Accumulate
90                  ! LIN indicates carriage return-line feed
100  IF Accumulate>1000 THEN PRINT "... not bad for one month!"
110                 ! Branching is reduced with IF-THEN statements such
                      as that in line 100.
120  END            ! Up to 160 characters can be entered for one line number.
```

RUN

```
If this is day  1  you owe me $ .01
If this is day  2  you owe me $ .02
If this is day  3  you owe me $ .04
If this is day  4  you owe me $ .08
If this is day  5  you owe me $ .16
If this is day  6  you owe me $ .32
If this is day  7  you owe me $ .64
If this is day  8  you owe me $ 1.28
If this is day  9  you owe me $ 2.56
If this is day  10  you owe me $ 5.12
        ⋮                  ⋮
If this is day  23  you owe me $ 41943.04
If this is day  24  you owe me $ 83886.08
If this is day  25  you owe me $ 167772.16
If this is day  26  you owe me $ 335544.32
If this is day  27  you owe me $ 671088.64
If this is day  28  you owe me $ 1342177.28
If this is day  29  you owe me $ 2684354.56
If this is day  30  you owe me $ 5368709.12
This is the total --$ 10737418.23
... not bad for one month!
```

Space Dependent Mode

The preceding program appears to have been entered in TYPWTR mode but was actually entered in space dependent mode. You access space dependent by holding down the CONTROL key (located in the alphanumeric block) and pressing the TYPWTR key; you exit from this mode the same way.

One advantage of using this mode is that variables, line labels and subprogram names entered in uppercase letters are interpreted as lower case except for the first letter which remains upper case. (Line labels are introduced in the next section and subprograms are discussed in Chapter 7 of this manual.)

Because space dependent has no effect on data in quotes and comments, that information in the program was entered using the SHIFT key. In addition, space dependent has no effect on items in DATA statements.

When the System 45 is in space dependent mode, keywords such as PRINT, IF and FOR remain upper case after being stored. However, they must have a space separating them from the rest of the statement—thus the term, space dependent. In space dependent mode, output functions LIN, TAB and SPA must also be entered with a separating space or parentheses. If neither is included, the functions are considered to be unassigned variables and return 0's as shown in the following example.

```
10   PRINT "Hewlett-Packard"
20   PRINT Lin1
30   PRINT Tab3,"System 45"
40   END
```

RUN

```
Hewlett-Packard
 0
 0                    System 45
```

Other space dependent characteristics that you should be aware of are explained in Chapter 4, "Programming Information", of the Operating and Programming Manual.

Line Labels

The BASIC language of the System 45 also lets you use self-explanatory labels such as Start_letter in line 10 of this next program to identify lines.

```
10 Start_letter:  PRINT PAGE,"To our customers and associates:"
20   PRINT LIN(2),SPA(4),"We here at XXXX Labs wish you and "
30   PRINT "your family a happy holiday and thank "
40   PRINT "you for your patronage."
50   PRINT TAB(5),"We hope to continue to serve you "
60   PRINT "in the new year."
70         ! TAB tabs to the specified column; SPA skips
             the specified number of spaces.
80   PRINT LIN(3),TAB(25),"Best Regards,",LIN(2),TAB(25);
90   PRINT "President,",LIN(1),TAB(25),"XXXX Labs Inc.";
120  INPUT "If you want another letter, press 1 and the CONT key",A
130  IF A=1 THEN Start_letter
140  END
```

By referencing line labels in branching statements such as that shown in line 130, you eliminate line-number guess work. You also ensure that if program lines are later edited, references to lines won't have to be changed to reflect altered line numbers.

(The underscore you see in line 10 is used to separate words in labels instead of the dash because the dash is interpreted as a minus sign.)

Another CONTROL Capability

The CONTROL key is also used to access special highlighting features such as inverse video, blinking and continuous underlining. To enter this message, for example:

PRINT "DO NOT ERASE PROGRAM"

type in PRINT and ", hold down the CONTROL key, press Special Function key k2 (underline mode), release the CONTROL key and k2 and type the remaining message. Release the underline mode before typing in the last " by holding down CONTROL and pressing k2.

To display the message in all three modes, hold down the CONTROL key and press k0, k1 and k2. If PRINTER IS 0, the hard-copy message shows underline mode only.

Remember to exit these modes after you have STORE d a line. Also, when you use any or all of the highlighting modes in a REM statement or after a comment delimiter (!), exit the specific mode(s) and then space one character position over before storing that line. Otherwise the remainder of your program listing will be highlighted in that particular mode or modes.

Special Function Keys

The Special Function key block, situated in the upper righthand corner of the keyboard, contains 16 Special Function keys with 32 functions – 16 with shift and 16 without.

These keys have a variety of uses, two of which have been mentioned: predefined typing aids (LIST – k14, EDIT LINE – k13, REWIND ":T15" – k7) and access keys to special CRT features (with the CONTROL key, inverse video – k0, blinking – k1, underlining – k2).

Listing Special Function Keys

For a list of all 32 key definitions (shown below) that are available at power on or SCRATCH A,

- type in LIST KEY #0 (If the internal printer is not included in your unit, execute LIST KEY #16.)
- press EXEC

The result is :

```
KEY 0-Undefined
KEY 1-Undefined
KEY 2-Undefined
KEY 3-Undefined
KEY 4-Undefined
KEY 5-Undefined
KEY 6
   -Clear line
    REWIND ":T14"
   -Execute

KEY 7
   -Clear line
    REWIND ":T15"
   -Execute

KEY 8
   -Clear line
    GET

KEY 9
   -Clear line
    LOAD

KEY10
   -Clear line
    SAVE

KEY11
   -Clear line
    STORE

KEY12
   -Clear line
    EDIT

KEY13
   -Clear line
    EDIT LINE

KEY14
   -Clear line
    LIST

KEY15
   -Clear line
    SCRATCH

KEY16-Undefined
KEY17-Undefined
KEY18-Undefined
KEY19-Undefined
KEY20-Undefined
KEY21-Undefined
KEY22-Undefined
KEY23-Undefined
KEY24-Undefined
KEY25-Undefined
KEY26-Undefined
KEY27-Undefined
KEY28-Undefined
KEY29-Undefined
KEY30-Undefined
KEY31-Undefined
```

Defining Procedure

You can define keys 0-5 and 16-31 (or redefine keys 6-15) to contain your own special functions. To define [k5], for instance, to contain a comment delimiter on the 30th space:

- press [k12] to enter edit key mode
- press [k5]
- space over 30 character positions using the space bar
- press and hold [SHIFT] and press [!/1]
- press [EXEC],
- press [k5] again to store the definition.

Now instead of spacing 30 character positions to start a comment on a program line, all you do is press [k5] after you've entered the statement. Your comments will then be easier to enter.

Editing Special Function Keys

Some editing and display keys are output literally when entered into a Special Function key definition and cannot be used to edit definitions. For example, [↑] is output as Up arrow. To edit definitions, use the [←], [→], [INS CHR] and [DEL CHR] keys. By holding down the [CONTROL] key and pressing each of these four keys, you can enter their respective literal definitions also.

An advantage of using such literal meanings is demonstrated in the following where [k9] is redefined as:

```
KEY 9
   -Clear line
    LOAD " "
   -Left arrow
   -Insert character
```

} entered using [CONTROL]

Now you can easily LOAD a program from a mass storage medium into the System 45 by just pressing [k9], entering the file name and pressing [EXEC].

To abort the editing process and clear the display, press the [STOP] key.

Interrupt and Priority

Special Function keys can also be used to interrupt program segments for branching purposes as shown in this next program. In line 10 an interrupt is declared by Special Function key [k1]. This means that at any time while the program is running, you can press [k1] and branch out of the loop. The random number that is generated at the time [k1] is pressed is then displayed.

```
10   ON KEY #1 GOTO Pick_a_number
20   RANDOMIZE       ! Selects random-number starting point
30   FOR I=1 TO 500
40   A=INT(RND*100) ! INT is integer function; RND is random number function
50   DISP A
60   NEXT I
70   STOP
80 Pick_a_number: DISP "***";A;"is your number"
90   END
```

The following program demonstrates interrupt **and** priority capability using Special Function keys.

```
10   ON KEY #5,4 GOSUB Firstpriority
20   ON KEY #2,3 GOSUB Secondpriority
30   ON KEY #1,2 GOSUB Thirdpriority
40   ON KEY #4,1 GOSUB Lastpriority
41   PRINTER IS 0
60   DISP "Press 1,2,4,5 in any order, at any speed."
61   GOTO 60
100  STOP
110 Firstpriority:  PRINT "Result from Key 5"
111      FOR I=1 TO 5
120      PRINT TAB(I);"*"
130      NEXT I
140      RETURN
150 Secondpriority: PRINT "Result from Key 2"
151      FOR J=8 TO 1 STEP -1
160      PRINT TAB(J);"**"
170      NEXT J
180      RETURN
190 Thirdpriority:  PRINT "Result from Key 1"
200      FOR H=1 TO 10
210      PRINT TAB(H);"* *"
220      NEXT H
230      RETURN
240 Lastpriority:  PRINT "Result from Key 4"
250      FOR K=30 TO 15 STEP -1
260      PRINT TAB(K*2);"*"
270      NEXT K
280      RETURN
290  END
```

The output of the preceding program depends on key-access speed and sequence.

All 32 Special Functions keys can be used for interrupt capability, but there are only 15 levels of priority available, with 15 being the highest. The key assigned the highest priority, in this case [k5], interrupts those assigned lower priority numbers- [k2], [k1] and [k4]; [k2] interrupts [k1] and [k4] and so forth.

For more information concerning interrupt and priority, see Chapter 4, "Programming Information" and Chapter 6, "Branching and Subroutines" of the Operating and Programming Manual.

Chapter 3
Built-in Functions

Many mathematical functions that are frequently used are built into the System 45. The following list gives the syntax and describes the function of each. Parentheses enclose arguments.

The program examples that immediately follow the list show how functions can be used in programs. To calculate a function from the keyboard, enter the function and argument(s) and then press [EXEC].

General Functions

ABS (X)	Returns absolute value of X
DROUND (X, Y)	Digit round—returns X to the number of significant digits indicated by Y
FRACT (X)	Returns fractional part of X
INT (X)	Returns integer part of X
MAX (X1, X2, X3, X4)	Returns greatest value of the series
MIN (X1, X2, X3, X4)	Returns smallest value of the series
PI	Represents approximate value of π − 3.1415926536
PROUND (X, Y)	Power-of-ten round—returns X rounded to the power-of-ten position indicated by Y
RND	Random number returns a random value greater than 0 and less than 1
SGN (X)	Returns a 1 if X is positive, −1 if X is negative and 0 if X is 0
SQR (X)	Returns the square root of a positive X

Logarithmic and Exponential Functions

EXP (X)	Returns the value of the Napierian constant *e* (2.71828182846) raised to power of X
LGT (X)	Returns common log (base 10) of a positive X
LOG (X)	Returns natural log (base *e*) of a positive X

Trigonometric Functions and Statements

angular units	DEG	Degree mode
	GRAD	Grad mode (more common in Europe)
	RAD	Radian mode (status at power on, on when SCRATCH A, or (RUN) is executed)
ACS (X)		Returns value of arccosine of X in angular units; X must be in the range of −1 to 1
ASN (X)		Returns value of arcsine of X; X must be in the range of −1 to 1
ATN (X)		Returns value of arctangent of X
COS (X)		Returns the cosine of the angle X
SIN (X)		Returns the sine of angle X
TAN (X)		Returns the tangent of angle X

Program example #1, using FIXED format:

```
10                  ! Line 30 indicates FIXED POINT format.
20                  ! In this case, 5 digits appear to the right of the decimal.
30  FIXED 5         ! Number-of-digits parameter can range from 0 to 12.
40  DEG             ! Sets DEGREE mode
50 Enter:   INPUT "Enter a positive integer, 1 to 5",Number
60   ON Number GOTO One,Two,Three,Four,Five
70                  ! Computed GOTO branches to line 90, 110, 130, 150 or 170
80                  ! depending on value of Number.
90 One:  PRINT "Arccosine of 1 =";ACS(Number)
100     GOTO Enter
110 Two:  PRINT "e^2 =";EXP(Number)
120     GOTO Enter
130 Three:     PRINT "Common log of 3 =";LGT(Number)
140     GOTO Enter
150 Four:      PRINT "Tangent of 4 =";TAN(Number)
160     GOTO Enter
170 Five:      PRINT "Natural log of 5 =";LOG(Number)
180     GOTO Enter
190 END              !  Press CONT to execute each entry.
```

(RUN) (example entries)

```
Natural log of 5 = 1.60944
Common log of 3 = .47712
Tangent of 4 = .06993
Arccosine of 1 = 0.00000
e^2 = 7.38906
```

Program example #2, using FLOAT format:

```
10   FLOAT 11      ! Indicates FLOATING POINT format; parameter range is 0-11
20   DEG
30      A=TAN(18)
40      B=COS(18)
50      C=SIN(18)
60   RAD           ! Sets RADIAN mode
70      D=LGT(6)
80      E=COS(18)
90   PRINTER IS 0,WIDTH(65)
100                ! WIDTH(65) limits the output width to 65 characters.
110  PRINT "A =";A;LIN(-1);"B =";B;LIN(-2);"C =";C;LIN(-2);
111                ! Negative number in LIN function suppresses carriage return
120  PRINT "D =";D;LIN(-1);"E =";E
130  STANDARD      ! Sets STANDARD format; up to 12 significant digits output
140  PRINT SPA(44);LIN(-1);"PI =";PI
150  END
```

RUN

```
A = 3.24919696231E-01
                              B = 9.51056516303E-01
                                                          C = 3.09016994376E-01
D = 7.78151250382E-01
                              E = 6.60316708200E-01
                                                          PI = 3.1415926536
```

Chapter 4
Formatted Output

As previously mentioned, PAGE, LIN, TAB and SPA are some of the functions used with PRINT to format output. For more format control, however, you can use the PRINT USING and IMAGE statements (similar to FORTRAN's WRITE and FORMAT) with specifiers.

These specifiers, listed on page 23, are symbol or letter codes. The following program shows how some of these can be used with PRINT USING and IMAGE to format information on the first line of a check.

```
5    DIM Name$[25]
10   INPUT "Enter name",Name$,"Enter amount",Amount
20     PRINT USING 30;Name$,Amount
30     IMAGE 10X,K,20X,3DC3D.2D
40   END
```

(RUN) (example entry where amount is entered as 1890.77)

Pay to the
order of _____ J. J. Henry _____ $ 1,890.77

(Name$ in lines 5, 10 and 20 is a string variable which is discussed in Chapter 5 of this manual.)

The PRINT USING statement in line 20 references the corresponding IMAGE statement in line 30 and specifies the variable names in the **print using list**. The IMAGE statement contains the **format string** that controls the output. In this case it allows for 10 blank spaces (10X), defines an output field for Name$ (K), and skips 20 spaces (20X). The dollar amount is formatted by 3DC3D.2D where 3D allows for 3 digits, C specifies a comma, . specifies a decimal and 2D allows for 2 digits after the decimal.

Implicit IMAGE

A PRINT USING statement can contain its implicit image as indicated in line 30 of the next financial summary program where the format string for the years is "3X6D5X".

```
10   PRINT USING 20;"ABC Imports",51499
20   IMAGE K,/,K,2/      ! / specifies carriage return-line feed
30   PRINT USING "3X6D5X";1965,1967,1970,1973,1977
40   IMAGE "$" 3DC3D.2D,3X
50   PRINT USING 40;7394.32,9934.87,88256.75,9793.69,130275.54
60   END
```


```
ABC Imports
51499

     1965          1967          1970          1973          1977
  $7,394.32     $9,934.87    $88,256.75     $9,793.69   $130,275.54
```

Notice that lines 10 and 50 can also contain implicit images as shown in the following revised program. To include literals such as "$" in a format string, an IMAGE statement must be used with a corresponding PRINT USING statement as shown in the original program.

```
10   PRINT USING "K,/,K,2/";"ABC Imports",51499
30   PRINT USING "3X6D5X";1965,1967,1970,1973,1977
50   PRINT USING "K,3DC3D.2D3X";"$",7394.32,"$",9934.87,"$",88256.75,"$",9793.69
,"$",130275.54
60   END
```


```
ABC Imports
51499
     1965          1967          1970          1973          1977
$  7,394.32   $  9,934.87   $ 88,256.75   $  9,793.69   $130,275.54
```

Specifiers

,	separates two specifiers
/	separates two specifiers; also outputs carriage return-line feed
@	separates specifiers; also indicates new page
X	outputs blank space
A	indicates single string character
D	specifies digit position—leading zeros replaced with blank spaces
Z	specifies digit position—leading zeros replaced with 0's
*	specifies digit position—leading zeros replaced with *'s
.	specifies decimal point as radix indicator[2]
R	specifies comma as radix indicator (more common in Europe)[2]
M	specifies a blank if a number is positive, − if negative
S	specifies + if a number is positive, − if negative
C	specifies comma as digit separator
P	specifies period as digit separator (more common in Europe)
E	causes output of E and two-digit exponent
K	defines field for numeric or string output
#	suppresses carriage return-line feed
+	suppresses line feed (on CRT display, suppresses output)
−	suppresses carriage return

2 Only one radix specifier allowed in an IMAGE or an implicit image.

Chapter 5
Strings

Literals or text within quotes are referred to as strings in System 45 BASIC and are assigned to single or multicharacter variables, with the variable followed by a $ such as:

```
Name$="Triple X Labs"
```

Quotes designate that which is text and are not part of the string.

Dimension

If a string is more than 18 characters long, its string variable name must be specified or dimensioned within a program by a DIM (dimension) statement or a COM (common) statement. (See Chapter 5, "Using Variables", of the Operating and Programming Manual for an explanation of COM.)

A string that contains 18 characters or less is implicitly dimensioned and can be entered into a program without a corresponding DIM statement as shown in the following example.

```
10   Name$="TRIPLE X LABS INC."
20   PRINT Name$
30   END
```

RUN

```
TRIPLE X LABS INC.
```

This next program shows how strings can be used in a mailing list application.

```
10   DIM Name$[25],Position$[25],Company$[25],Address$[35],City$[20],Referenceco
de$[25],State$[20]      ! Strings are dimensioned in brackets.
20   INPUT "How many entries?",X
30   PRINTER IS 0
40   FOR Entry=1 TO X
50    INPUT "Enter name",Name$,"Position?",Position$,"Company?",Company$,"Addres
s?",Address$,"City?",City$,"State?",State$,"Zipcode?",Zipcode
60    PRINT USING "4(K,/),3(K),4X,K";Name$,Position$,UPC$(Company$),Address$,Cit
y$;",";State$,Zipcode
70    Referencecode$=City$[1,4]&State$
80    PRINT USING "2/8XK";Referencecode$
90   NEXT Entry
100 END                  ! Press the CONT key after typing each entry.
```

(RUN) (example entry)

```
J. J. Henry
Order Coordinator
ABC IMPORTS LTD.
P.O. Box 73
Gaithersburg,Maryland    20760

        GaitMaryland
```

Notice that the string variable name, `Name$`, is dimensioned to contain no more than 25 characters. Should you enter a name that exceeds that dimension (or make an entry for any string variable name that exceeds its dimension) and then press (CONT), `ERROR 18` is displayed. You can then either change the dimension number in line 10 or edit the entry to contain 25 characters.

Concatenation

Strings can be concatenated, or joined, with an `&` as shown in line 70 of the previous program. In that example the first four letters of the city, which are indicated by the **substring specifiers** `[1, 4]` and comprise a **substring**, are concatenated to the state (without a separating space) and stored in another string variable, `Referencecode$`.

String Functions

The UPC$ in line 60 is a string function that converts the string expression, Company$, to upper case letters.

Another string function, CHR$, returns an ASCII character that corresponds to the numeric argument. (See Appendix A of this manual for the ASCII Character Code table.) The following program shows an application using CHR$ to access and exit inverse video, underline and blinking.

```
10   DIM A$[60]
20   A$="PLEASE "&CHR$(135)&"DO NOT ERASE PROGRAM"&CHR$(128)
30   DISP A$
40   END
```

(RUN) (boxed area indicates inverse video, underline and blinking)

```
PLEASE DO NOT ERASE PROGRAM
```

Other string functions that let you manipulate string contents include:

LWC$(X$)	Converts uppercase letters to lower case		
		LWC$("OHIO")	ohio
REV$(X$)	Reverses order of characters		
		REV$("Ohio")	oihO
RPT$(X$,Y)	Repeats X$, Y times		
		RPT$("*",7)	*******
TRIM$(X$)	Deletes leading and trailing blanks		
		TRIM$(" OHIO ")&"STATE"	OHIOSTATE
VAL$(X)	Returns a **string** representing numeric value of the numeric X		
		VAL$(4*16)	64

These string functions return numeric results:

LEN(X$)	Returns number of characters LEN("Ohio") 4
NUM("XYZ")	Returns ASCII decimal equivalent of **first** character NUM("&+") 38
POS(Xy$,y$)	Determines numeric position of substring y within string X POS("Columbus,Ohio","Ohio") 10
VAL(X$)	Returns numeric value of string of digits for calculations VAL("9872.6") 9872.6

CHR$ can also be used to access special printer capabilities that let you

- generate up to nine of your own specially defined characters,
- increase hard-copy character height 150%, and
- enter the plotting mode

Procedures for these and other capabilities are explained in Appendix C, "Advanced Printing Techniques" of the Operating and Programming Manual.

String Arrays

The System 45 also lets you use collections of strings known as string arrays. An example of a string array is shown in the following where four elements of a one-dimensional string array are declared by A$(1:4) and each element is dimensioned to contain not more than 60 characters.

```
10   DIM A$(1:4)[60]
20     PRINT "ALL THOSE REGISTERED",,"CALL THESE IN PRECINCT 24"
30     PRINT RPT$("=",80)
40 Inputdata:INPUT "Enter name, county, precinct, telephone #",A$(*)
50     PRINT A$(*)
60     B$=UPC$(A$(1))
70     IF A$(3)="24" THEN PRINT USING "40XK3X,K";B$,A$(4)
80   GOTO Inputdata
90     END
```

RUN (example entries)

```
          ALL THOSE REGISTERED              CALL THESE IN PRECINCT 24
          ========================  ======================================
A$(1)     J. J. Henry
A$(2)     Delta
A$(3)     36
A$(4)     699-3254


A$(1)     P. H. Foster
A$(2)     Smith
A$(3)     24
A$(4)     543-6588

                                        B$          A$(4)
                                    P. H. FOSTER    543-6588
```

Related comments concerning this program:

1. The two commas you see in line 20 cause two 20-character fields to be output and separate the two headings.
2. The asterisks in lines 40 and 50 are array identifiers and are explained in the next chapter.
3. String array elements can be manipulated just as simple strings are as demonstrated in lines 60 and 70.

Chapter 6

Arrays

The System 45 can handle arrays that have up to six dimensions. The following program defines two elements in a two-dimensional array and prints the array. Its size is DIMensioned by subscripts that are enclosed in parentheses and preceded by the array name.

```
10   DIM Alpha(2,3)
20               ! Alpha is supposedly dimensioned to
                 six elements.
30         Alpha(1,2)=9926
40         Alpha(2,3)=7003
50   PRINT USING "/K3X,K5X,K/";"Individual elements:",Alpha(1,2),Alpha(2,3)
60   PRINT "Printed 2X3 array",Alpha(*)
70               ! Line 60 contains an array identifier, (*), which is
                   discussed in the next section.
80   END
```

RUN

```
Individual elements:   9926     7003
Printed 2X3 array
 0                  0                  0                  0

 0                  0                  9926               0

 0                  0                  0                  7003
```

Obviously the array contains 12 elements, not six as specified in the subscripts. The reason for this is that arrays have a lower bound of 0 which is a default condition of the System 45 at power on. In this mode the elements of Alpha are represented as:

0	(0,0)	0	(0,1)	0	(0,2)	0	(0,3)
0	(1,0)	0	(1,1)	9926	(1,2)	0	(1,3)
0	(2,0)	0	(2,1)	0	(2,2)	7003	(2,3)

Bounds

To set the lower bound to 1 and eliminate the extra six elements, insert this statement in the program **before** the DIM statement: OPTION BASE 1. (OPTION BASE 1 must also precede the COM statement and the INTEGER, REAL and SHORT statements which are discussed later.)

To reset the lower bound to 0, DELete the OPTION BASE 1 statement by typing in DEL and the line number and then pressing (EXEC). Or, for documentation purposes, edit the statement to read OPTION BASE 0.

You can also specify other upper and lower bounds such as Alpha(-2:1, 2:4) which dimensions a 4X3 array. These type of bounds are convenient when you are using formulas that have negative indexing, for example.

Array Statements and Operations

Statements such as READ, INPUT and PRINT—usually associated with nonarray operations—are also used with arrays but must be prefaced by MAT or followed by an array identifier (*). With (*) you can READ, INPUT and PRINT both arrays and nonarray items in one line as shown in lines 150 and 350 of the next program.

```
10   OPTION BASE 1
30     DIM Numberlist(8)
50      DATA 2,36,5,12,6,9,31,18
130     MAT READ Numberlist
150     PRINT "Original data";LIN(2);Numberlist(*);
170           FOR I=1 TO 7 ! This starts the sort routine which
                             utilizes a nested FOR-NEXT.
190           L=I+1
210            FOR N=L TO 8
230            IF Numberlist(N)>=Numberlist(I) THEN Gonextn
250            Temporary=Numberlist(I)
270            Numberlist(I)=Numberlist(N)
290            Numberlist(N)=Temporary
310 Gonextn:     NEXT N
330           NEXT I
350     PRINT "Data sorted";LIN(2);Numberlist(*);
470     END
```

(RUN)

```
Original data

 2  36  5  12  6  9  31  18

Data sorted

 2  5  6  9  12  18  31  36
```

If a MATPRINT were used in line 150, for example, only arrays could be specified in that line, and an additional line containing PRINT "Original data" would be necessary.

Arrays can also be manipulated for applications such as solving simultaneous equations. X and Y in these equations, for example:

$$19X + 8Y = 223$$
$$16X + 7Y = 194$$

can be easily computed using the MATrix INVerse statement and checked by the DETerminent statement as shown in this program:

```
10   OPTION BASE 1
20   DIM Alpha(2,2),Beta(2,1),Delta(2,2),Answer(2,1)
30            ! The matrix to be inverted must be square.
40        DATA  19,8,16,7,223,194
50        READ Alpha(*),Beta(*)
60        FIXED 4
70           PRINT "DETerminant of ALPHA is";DET(Alpha),LIN(2)
80        MAT Delta=INV(Alpha)
90        MAT Answer=Delta*Beta
100       PRINT "X and Y are, respectively:";SPA(3),Answer(*)
110  END
```

RUN

```
DETerminant of ALPHA is 5.0000


X and Y are, respectively:
 1.8000

 23.6000
```

Other array-manipulation statements let you:

Assign the value of one to each element in an array	`MAT A=CON`
Assign zero to each element of an array	`MAT A=ZER`
Assign a constant value to each element of an array	`MAT A=(Z)`
Perform arithmetic or relational operations with each element of an array using a constant scalar (X)	`MAT A=B+(X)`
Perform arithmetic or relational operations with **corresponding elements** of two arrays	`MAT A=B. C` (the multiplication operator in this case is . not *)
Copy elements of an array into another array	`MAT A=B`
Specify identity matrix where elements of the main diagonal have value of 1; other elements are 0's	`MAT A=IDN`
Multiply two matrices (not the same as MAT A=B . C)	`MAT A=B*C`
Find the sum of columns	`MAT A=CSUM(B)`
Find the sum of rows	`MAT A=RSUM(B)`
Find the sum of all elements of an array (B)	`C=SUM(B)`
Determine number of rows in an array (B)	`C=ROW(B)`
Determine number of columns in an array (B)	`C=COL(B)`
Redimension working size of array	`REDIM A (X,Y)`
Determine the dot product of two **vectors**	`DOT(A,B)`
Transpose column elements to rows and row elements to columns	`MAT A=TRN(B)`
Use mathematical functions with arrays	`MAT A=LOG(B)`

Type Declaration

The INTEGER, SHORT and REAL statements, mentioned in a previous section, also dimension array variables and simple variables. The following chart shows the number of bytes reserved in read/write memory by each type.

Type	Simple Variable	Array Variable
INTEGER	4 bytes	4 bytes + 4 bytes per dimension + 2 bytes per element
SHORT (Short precision)	6 bytes	4 bytes + 4 bytes per dimension + 4 bytes per element
REAL (Full precision)	10 bytes	4 bytes + 4 bytes per dimension + 8 bytes per element

Chapter 7
Subroutines and Subprograms

Another feature of the System 45 is transfer of program control through the use of subroutines and subprograms.

Subroutines

A **subroutine** in System 45 BASIC consists of a group of statements, **within** a program, that can be accessed as many times as necessary by the statement GOSUB and its line identifier. After the subroutine is completed, computed values and program control are returned to the line following the GOSUB statement, as demonstrated in the Special Function key interrupt and priority example in Chapter 2 of this manual.

Subprograms

A **subprogram** consists of several statements that perform a specific computation or computations and are physically located after the main or calling program segment. Because they are independent of the calling program, subprograms can be entered in modules which simplifies development and debugging procedures.

Subprograms are convenient because they permit repetition of an operation without code duplication; they also offer flexibility in that different values can be substituted each time a subprogram is accessed.

One type of subprogram is **subroutine subprogram** (similar to FORTRAN's subroutine subprogram) which can compute several values. An example of a BASIC subroutine subprogram is shown in the following program where CALL and a subprogram name access the subprogram, SUB and the subprogram name indicate the first line and SUBEND (or SUBEXIT) terminates the subprogram and returns control to the calling program.

Calling program

```
10   INPUT "HOUR?",Hours,"MINUTES?",Minutes
20   INPUT "AM OR PM?",Am_pm$
30     CALL Timeconversion(Hours,Minutes,Am_pm$,Militarytime)
40               ! The CALL statement contains the pass
                   parameter list.
50     PRINT Hours;":";Minutes;Am_pm$;" is";Militarytime;"hours"
60      END
```

Subprogram

```
70        SUB Timeconversion(H,I,J$,K)
80               ! The  SUB statement contains the formal
                   parameter list.
90        K=100*H+I
100       IF J$="AM" THEN SUBEXIT
110              ! SUBEXIT transfers control to main program
                   before SUBEND is executed.
120              ! It's an optional statement, not required
                   with each and every subroutine subprogram.
130       K=K+1200
140       SUBEND
```

RUN (example entry)

```
10 : 17 AM is 1017 hours
```

The other type of subprogram is the **multiple-line function subprogram** which is comparable to the function subprogram in FORTRAN. It defines a numeric or string function and returns a **single** value to the calling program.

An example of a multiple-line function subprogram is shown below where FN and the subprogram name reference the subprogram, DEF FN and the function name indicate the first line, RETURN and an expression return the function value to the calling program and FNEND terminates the subprogram.

```
5  B=9
10   A=B+FNChange(B,C)    ! Contains pass parameter list
20   PRINT "A =";A
30   END
40   DEF FNChange(Y,Z)    ! Contains formal parameter list
45 Y=6
46 Z=3
60   V=Y/Z
70   RETURN V             ! Specifies value returned to
                            calling program
80   FNEND
```

RUN

```
A = 8
```

Parameters

Pass parameters, indicated in the preceding examples, pass values from the calling program to the subprogram by **reference** or by **value**. In the last program, for example, parameters are passed by reference. If line 10 was edited to:

```
10 A=B+FNChange((B),(C))
```

the parameter would be passed by value and A would equal 11.

The pass parameter list can contain numeric and string variables, constants, array identifiers (*) and mass storage file numbers.

Formal parameter lists define subprogram variables according to the pass parameter list. Formal parameter type and position must correspond to pass parameter type and position but variable names can differ.

Recursion

In System 45 BASIC subprograms have the capability of calling themselves as demonstrated in this program which calculates factorials for numbers 1 to 50.

```
10   FOR Number=1 TO 50
20   N=FNFactorial(Number)
30   PRINT Number;"! =";N
40   NEXT Number
50   END
60     DEF FNFactorial(A)
70     IF A<=1 THEN RETURN 1
80     RETURN FNFactorial(A-1)*A
90     FNEND
```

RUN

```
1 ! = 1
2 ! = 2
3 ! = 6
4 ! = 24
5 ! = 120
6 ! = 720
7 ! = 5040
8 ! = 40320
```

⋮

```
44 ! = 2.65827157479E+54
45 ! = 1.19622220866E+56
46 ! = 5.50262215984E+57
47 ! = 2.58623241512E+59
48 ! = 1.24139155926E+61
49 ! = 6.08281864037E+62
50 ! = 3.04140932019E+64
```

For more information on parameter passing and subprogram considerations such as temporary defaults, deactivated interrupts and deletion of subprograms, see Chapter 7, "Subprograms", of the Operating and Programming Manual.

Chapter 8
Debugging

The System 45's debugging capability is a powerful tool for tracing logic and data flows of programs. In fact debugging is accessed by a variety of TRACE statements, two of which are demonstrated by this next program[3].

The output following the listing is obtained by latching the PRT ALL key before pressing RUN.

```
10   PRINT ALL IS 0
20   TRACE ALL
30   TRACE WAIT 500
40   FOR I=13 TO 2 STEP -2
50   PRINT I;LOG(I)
60   NEXT I
70   END
```

RUN

```
TRACE--LINE 40, I = 13
 13  2.56494935745
TRACE--LINE 60, I = 11
TRACE--FROM 60 TO 50
 11  2.39789527279
TRACE--LINE 60, I = 9
TRACE--FROM 60 TO 50
 9  2.19722457733
TRACE--LINE 60, I = 7
TRACE--FROM 60 TO 50
 7  1.94591014905
TRACE--LINE 60, I = 5
TRACE--FROM 60 TO 50
 5  1.60943791243
TRACE--LINE 60, I = 3
TRACE--FROM 60 TO 50
 3  1.09861228866
TRACE--LINE 60, I = 1
```

The TRACE ALL statement traces all logic and variables to let you see what the entire program is doing. TRACE WAIT, used in conjunction with another TRACE statement, delays a specified number of milliseconds after each line of output to give you time to examine that line.

3 The STEP key can also be used for simple debugging/monitoring. Each time STEP is pressed, one line of the program is executed.

Other TRACE statements let you:

- trace some or all of a program; indicates a line-to-line trace when a branch is encountered; statement is TRACE
  example output: TRACE--FROM 60 TO 50

- pause before a specified line is executed to verify that that line has been reached; execution is continued by pressing (CONT); statement is TRACE PAUSE followed by line identifier

  example statement: TRACE PAUSE 40
  example output: TRACE LINE 30, I=1
  40 FOR J=1 to 4

- trace changes in values of variables for numerics, strings, arrays and subprograms (only variables passed by reference); statement is TRACE VARIABLES followed by variables list; up to five variables can be indicated in the list

  example statement: TRACE VARIABLES J
  example output: TRACE--LINE 40, J=1
  TRACE--LINE 60, J=2

- trace all variables with line numbers indicated; statement is TRACE ALL VARIABLES

  example output: TRACE--LINE 30, I=1
  TRACE--LINE 40, J=1

To cancel all tracing operations, execute NORMAL or any one of five SCRATCH commands.

Tracing logic flow with TRACE statements temporarily defaults the System 45 to SERIAL mode if OVERLAP has been previously indicated. These two conditions are discussed in Chapter 10 of this manual.

Chapter 9
Error Testing and Recovery

The System 45 has the capability of trapping and recovering from run-time errors in programs, such as division by zero, with the ON ERROR statement. Without error trapping and recovery, a program containing run-time errors is halted during execution.

The following program example demonstrates the trapping capability and a recovery procedure when a negative number is input as a factor in a square root calculation.

```
10   INPUT "A?",A,"B?",B
20   ON ERROR GOTO Fix_it
30   C=SQR(A*B)
40   PRINT "Square root of ";A;"*";B;"=";C
50   STOP
60 Fix_it:    IF (ERRN<>30) OR (ERRL<>30) THEN Other_error
70            C=SQR(ABS(A*B))
80            PRINT LIN(1),"With absolute value substitution,"
90            GOTO 40
100 Other_error:  DISP ERRM$
110  END
```

RUN (example entries)

```
A?
123658
B?
56547
Square root of  123658 * 56547 = 83621.1033531

A?
-6587
B?
589

With absolute value substitution,
Square root of -6587 * 589 = 1969.70632328
```

The ON ERROR statement in line 20 declares that when an error is encountered, a branch to the recovery routine, Fix_it, occurs. In line 60 the error number function, ERRN, tests for the error number 30 (SQR of negative number) and the error line function, ERRL, tests for line

30. If one of these conditions is true – ERRN ≠ 30 or ERRL ≠ 30 – then another error has occurred or ERROR 30 has occurred in another line.

If this happens, DISP ERRM$ in line 100 displays the error message as a string. With ERRM$ you have the added capability of manipulating the error message using string functions.

If one condition is false – error number is 30 or an error occurs in line 30 – the program continues with line 70.

Other branches that are allowed in ON ERROR statements besides GOTO are GOSUB and CALL. Refer to Chapter 8, "Editing and Debugging", of the Operating and Programming Manual for information concerning subroutines and subprograms in error testing and recovery procedures.

Chapter 10
Overlapped Processing

At power on, SCRATCH or SCRATCH A the System 45 is in SERIAL mode which means computation and I/O occur sequentially. In some programs, however, you may want computation to occur simultaneously with I/O to one or more devices. This is possible with the OVERLAP statement.

To observe the difference between SERIAL and OVERLAP for yourself, enter this program (which will be in SERIAL mode, initially) and press (RUN).

```
10   J=3.56876
20   FOR I=1 TO 26
30     A=RND*COS(1/(2*PI*SQR(1*I^2)))*LOG(ATN(J^2*LGT(I)*J+J^2))
40     PRINTER IS 0
50     PRINT I,A,
60     PRINTER IS 16
70     PRINT I,A,
80   NEXT I
90   END
```

(RUN)

```
1                .268134998272      2                .162754236309
3                .181202601424      4                .219297957036
5                .324994363923      6                .223270985035
7                .261168923145      8                .170341889434
9                .391337074017      10               .35717523396
11               .385827097857      12               .431957379226
13               .178636352197      14               .139196321779
15               1.71804272031E-02  16               .308255129046
17               .183452901088      18               .381376354299
19               .438371686396      20               3.25199494241E-02
21               .346384578751      22               .395610232103
23               8.86396886068E-02  24               .121067263004
25               .347236191966      26               .404844079188
```

Now type in OVERLAP and press (EXEC) and then (RUN). The calculations are output much faster than when SERIAL is indicated. (OVERLAP and SERIAL can be statements in programs or commands entered from the keyboard.)

OVERLAP remains in effect until SERIAL is executed or SCRATCH or SCRATCH A is executed.

With some programs it is advisable not to access the OVERLAP mode such as when ON ERROR is used or where you have only a few I/O operations and many computations or vice versa.

For more information on OVERLAP and its effect on mass storage operations, see Chapter 4, "Data Transfer", of the Mass Storage Techniques Manual.

Appendix A

ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec	ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec	ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec	ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec
NUL	00000000	000	0	space	00100000	040	32	@	01000000	100	64	`	01100000	140	96
SOH	00000001	001	1	!	00100001	041	33	A	01000001	101	65	a	01100001	141	97
STX	00000010	002	2	"	00100010	042	34	B	01000010	102	66	b	01100010	142	98
ETX	00000011	003	3	#	00100011	043	35	C	01000011	103	67	c	01100011	143	99
EOT	00000100	004	4	$	00100100	044	36	D	01000100	104	68	d	01100100	144	100
ENQ	00000101	005	5	%	00100101	045	37	E	01000101	105	69	e	01100101	145	101
ACK	00000110	006	6	&	00100110	046	38	F	01000110	106	70	f	01100110	146	102
BEL	00000111	007	7	'	00100111	047	39	G	01000111	107	71	g	01100111	147	103
BS	00001000	010	8	(	00101000	050	40	H	01001000	110	72	h	01101000	150	104
HT	00001001	011	9	)	00101001	051	41	I	01001001	111	73	i	01101001	151	105
LF	00001010	012	10	*	00101010	052	42	J	01001010	112	74	j	01101010	152	106
VT	00001011	013	11	+	00101011	053	43	K	01001011	113	75	k	01101011	153	107
FF	00001100	014	12	,	00101100	054	44	L	01001100	114	76	l	01101100	154	108
CR	00001101	015	13	–	00101101	055	45	M	01001101	115	77	m	01101101	155	109
SO	00001110	016	14	.	00101110	056	46	N	01001110	116	78	n	01101110	156	110
SI	00001111	017	15	/	00101111	057	47	O	01001111	117	79	o	01101111	157	111
DLE	00010000	020	16	ø	00110000	060	48	P	01010000	120	80	p	01110000	160	112
DC_1	00010001	021	17	1	00110001	061	49	Q	01010001	121	81	q	01110001	161	113
DC_2	00010010	022	18	2	00110010	062	50	R	01010010	122	82	r	01110010	162	114
DC_3	00010011	023	19	3	00110011	063	51	S	01010011	123	83	s	01110011	163	115
DC_4	00010100	024	20	4	00110100	064	52	T	01010100	124	84	t	01110100	164	116
NAK	00010101	025	21	5	00110101	065	53	U	01010101	125	85	u	01110101	165	117
SYN	00010110	026	22	6	00110110	066	54	V	01010110	126	86	v	01110110	166	118
ETB	00010111	027	23	7	00110111	067	55	W	01010111	127	87	w	01110111	167	119
CAN	00011000	030	24	8	00111000	070	56	X	01011000	130	88	x	01111000	170	120
EM	00011001	031	25	9	00111001	071	57	Y	01011001	131	89	y	01111001	171	121
SUB	00011010	032	26	:	00111010	072	58	Z	01011010	132	90	z	01111010	172	122
ESC	00011011	033	27	;	00111011	073	59	[	01011011	133	91	{	01111011	173	123
FS	00011100	034	28	<	00111100	074	60	\	01011100	134	92	¦	01111100	174	124
GS	00011101	035	29	=	00111101	075	61	]	01011101	135	93	}	01111101	175	125
RS	00011110	036	30	>	00111110	076	62	^	01011110	136	94	~	01111110	176	126
US	00011111	037	31	?	00111111	077	63	_	01011111	137	95	DEL	01111111	177	127

Appendix B

Error Messages

1	Missing ROM or configuration error
2	Memory overflow
3	Line not found or not in current program segment
4	Improper return
5	Abnormal program termination
6	Improper FOR/NEXT matching
7	Undefined function or subroutine
8	Improper parameter matching
9	Improper number of parameters
10	String value required
11	Numeric value required
12	Attempt to redeclare variable
13	Array dimensions not specified
14	Multiple OPTION BASE statements or OPTION BASE statement preceded by variable declarative statements
15	Invalid bounds on array dimension or string length in memory allocation statement
16	Dimensions are improper or inconsistent
17	Subscript out of range
18	Substring out of range or string too long
19	Improper value
20	Integer precision overflow
21	Short precision overflow

22	Real precision overflow
23	Intermediate result overflow
24	TAN (N*π/2), when N is odd
25	Magnitude of argument of ASN or ACS is greater than 1
26	Zero to negative power
27	Negative base to non-integer power
28	LOG or LGT of negative number
29	LOG or LGT of zero
30	SQR of negative number
31	Division by zero
32	String does not represent valid number or string response when numeric data required
33	Improper argument for NUM, CHR$, or RPT$ function
34	Referenced line is not IMAGE statement
35	Improper format string
36	Out of DATA
37	EDIT string longer than 160 characters
38	I/O function not allowed
39	Function subprogram not allowed
40	Improper replace, delete or REN command
41	First line number greater than second
42	Attempt to replace or delete a busy line or subprogram
43	Matrix not square
44	Illegal operand in matrix transpose or matrix multiply
45	Nested keyboard entry statements
46	No binary in STORE BIN or no program in SAVE
47	Subprogram COM declaration is not consistent with main program

48	Recursion in single line function
49	Line specified in ON declaration not found
50	File number less than 1 or greater than 10
51	File not currently assigned
52	Improper mass storage unit specifier
53	Improper file name
54	Duplicate file name
55	Directory overflow
56	File name is undefined
57	Mass Storage ROM is missing
58	Improper file type
59	Physical or logical end-of-file found
60	Physical or logical end-of-record found in random mode
61	Defined record size is too small for data item
62	File is protected or wrong protect code specified
63	The number of physical records is greater than 32767
64	Medium overflow (out of user storage space)
65	Incorrect data type
66	Excessive rejected tracks during a mass storage initialization
67	Mass storage parameter less than or equal to 0
68	Invalid line number in GET or LINK operation
69 – 79	See Mass Storage ROM errors
80	Cartridge out or door open
81	Mass storage device failure
82	Mass storage device not present
83	Write protected
84	Record not found

85	Mass storage medium is not initialized
86	Not a compatible tape data cartridge
87	Record address error
88	Read data error
89	Check read error
90	Mass storage system error
91-99	See Mass Storage ROM errors
100	Item in print using list is string but image specifier is numeric
101	Item in print using list is numeric but image specifier is string
102	Numeric field specifier wider than printer width
103	Item in print using list has no corresponding image specifier
104-109	Unused
110-119	See Graphics ROM errors

SYSTEM ERROR
System Error octal number

These two errors indicate an error in the machine's firmware system; they are fatal errors. If reset does not bring control back, the machine must be turned off, then on again. If the problem persists, contact your Sales and Service Office.

I/O Device Errors

Two error messages can occur when attempting to direct an operation to an I/O device that is not ready for use. A printer which is out of paper is an example. The first message is –

I/O ERROR ON SELECT CODE select code

If the condition is not correct, the machine beeps intermittently and the following message replaces the first –

DEVICE TIMEOUT ON SELECT CODE select code

The I/O device can be made usable by correcting the error (loading paper for example), then executing the READY# command –

READY# select code

This command readies the I/O device and the operation which was attempted is completed.

Mass Storage ROM Errors

69	Format switch off
70	Not a disc interface
71	Disc interface power off
72	Incorrect controller address, or controller power off
73	Incorrect device type in mass storage unit specifer
74	Drive missing or power off
75	Disc system error
76	Incorrect unit code in mass storage unit specifier
77-79	Unused
91-99	Unused

Graphics ROM Errors

110	Plotter specifications not recognized.
111	Plotter not previously specified.
112	CRT Graphics hardware not installed.
113	LIMIT specifications out of range.
114-119	Unused

Subject Index

a

b

c

d

e

f

g

h

i

k

l

m

n

o

p

q

r

s

t

u

v

w

HEWLETT hp PACKARD

SALES & SERVICE OFFICES

AFRICA, ASIA, AUSTRALIA

ANGOLA
Telectra
Empresa Técnica de Equipamentos Eléctricos, S A R L
R Barbosa Rodrigues, 42-1 DT
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 HEWPARO Melbourne

Hewlett-Packard Australia Pty Ltd
31 Bridge Street
Pymble
New South Wales, 2073
Tel 449-6566
Telex 21561
Cable HEWPARO Sydney

Hewlett-Packard Australia Pty Ltd
153 Greenhill Road
Parkside, S A , 5063
Tel 272-5911
Telex 82536 ADEL
Cable HEWPARO ADELAIDE

Hewlett-Packard Australia Pty Ltd
141 Stirling Highway
Nedlands, W A 6009
Tel 86-5455
Telex 93859 PERTH
Cable HEWPARO PERTH

Hewlett-Packard Australia Pty Ltd
121 Wollongong Street
Fyshwick, A C T 2609
Tel 95-2733
Telex 62650 Canberra
Cable HEWPARO CANBERRA

Hewlett Packard Australia Pty Ltd
5th Floor
Teachers Union Building
495-499 Boundary Street
Spring Hill, 4000 Queensland
Tel 229-1544
Cable HEWPARO 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 SCHMIOTCO Hong Kong

INDIA
Blue Star Ltd
Kasturi Buildings
Jamshedji Tata Rd
Bombay 400 020
Tel 29 50 21
Telex 001-2156
Cable BLUEFROST

Blue Star Ltd
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
Bombay 400 025
Tel 45 78 87
Telex 011-4093
Cable FROSTBLUE

Blue Star Ltd
Band Box House
Prabhadevi
Bombay 400 025
Tel 45 73 01
Telex 011-3751
Cable BLUESTAR

Blue Star Ltd
7 Hare Street
P O Box 506
Calcutta 700 001
Tel 23-0131
Telex 021-7655
Cable BLUESTAR

Blue Star Ltd
7th & 8th Floor
Bhandari House
91 Nehru Place
New Delhi 110024
Tel 634770 & 635166
Telex 031-2463
Cable BLUESTAR

Blue Star Ltd
Blue Star House
11/11A Magarath Road
Bangalore 560 025
Tel 55668
Telex 043-430
Cable BLUESTAR

Blue Star Ltd
Meenakshi Mandiran
xxx/1678 Mahatma Gandhi Rd
Cochin 682 016
Tel 32069,32161,32282
Telex 0885-514
Cable BLUESTAR

Blue Star Ltd
1-1-117/1
Sarojini Devi Road
Secunderabad 500 003
Tel 70126, 70127
Cable BLUEFROST
Telex 015-459

Blue Star Ltd
2/34 Kodambakkam High Road
Madras 600034
Tel 82056
Telex 041-379
Cable BLUESTAR

INDONESIA
BERCA Indonesia P T
P O Box 496/Jkt
JL.Abdul Muis 62
Jakarta
Tel 40369, 49886,49255,356038
JKT 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 MARKET
TOK 23-724
Cable YHPMARKET

Yokogawa-Hewlett-Packard Ltd
Chuo Bldg., 4th Floor
4-20, Nishinakajima 5-chome
Yodogawa-ku, Osaka-shi
Osaka,532
Tel 06-304-6021

Yokogawa-Hewlett-Packard Ltd
Nakamo Building
24 Kami Sasajima-cho
Nakamura-ku, **Nagoya**, 450
Tel (052) 571-5171

Yokogawa-Hewlett-Packard Ltd
Tangawa 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
Kumagaya Asahi
Hackijuni Building
4th Floor
3-4, 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 INTAERIO 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
Courtenay Place
Wellington
Tel 877-199
Cable HEWPACK Wellington

Hewlett-Packard (N Z) Ltd.
Pakuranga Professional Centre
267 Pakuranga Highway
Box 51092
Pakuranga
Tel 569-651
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
Cable OENTAL Auckland

Analytical/Medical Only
Medical Supplies N Z Ltd
P O Box 1994
147-161 Tory St
Wellington
Tel 850-799
Telex 3858
Cable OENTAL, Wellington

Analytical/Medical Only
Medical Supplies N Z Ltd
P O Box 309
239 Stanmore Road
Christchurch
Tel 892-019
Cable OENTAL, Christchurch

Analytical/Medical Only
Medical Supplies N Z Ltd
303 Great King Street
P O Box 233
Dunedin
Tel 88-817
Cable OENTAL, 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 2894
Cable COOPERATOR Karachi

Mushko & Company, Ltd
38B, Satellite Town
Rawalpindi
Tel 41924
Cable FEMUS Rawalpindi

PHILIPPINES
The Online Advanced Systems Corporation
Rico House
Amorsolo cor Herrera Str
Legaspi Village, Makati
Metro **Manila**
Tel 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
1150 Depot Road
Alexandra P O Box 58
Singapore 4
Tel 270-2355
Telex HPSG RS 21486
Cable HEWPACK, Singapore

SOUTH AFRICA
Hewlett-Packard South Africa (Pty), Ltd
Private Bag Wendywood
Sandton, Transvaal 2144
Hewlett-Packard Centre
Daphne Street, Wendywood,
Sandton, Transvaal 2144
Tel 802-10408
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 Provnce, 7450
Pine Park Centre, 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 Hsiao West Road
Sec. 1, 7th Floor
Taipei
Tel: 3819160-4
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 es Salaam
Tel 21251 Ext. 265
Telex 41030

THAILAND
UNIMESA Co., Ltd.
Elcom Research Building
2538 Sukumvit Ave
Bangkok
Tel: 3932387, 3930338
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

CANADA

ALBERTA
Hewlett-Packard (Canada) Ltd
11620A - 168th Street
EdmontonT5M 3T9
Tel (403) 452-3670
TWX 610-831-2431

Hewlett-Packard (Canada) Ltd
210 7220 Fisher St S E
Calgary T2H 2H8
Tel (403) 253-2713
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

MANITOBA
Hewlett-Packard (Canada) Ltd
513 Century St.
St James
Winnipeg R3H 0L8
Tel (204) 786-7581
TWX 610-671-3531

NOVA SCOTIA
Hewlett-Packard (Canada) Ltd.
800 Windmill Road
Dartmouth B3B 1L1
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 610-563-1636

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 122443 AR CIGY
Cable HEWPACK ARG

BOLIVIA
Casa Kavlin S A
Calle Potosi 1130
P O Box 500
La Paz
Tel 41530,53221
Telex CWC BX 5298 ITT 3560082
Cable KAVLIN

BRAZIL
Hewlett-Packard do Brasil I e C Ltda
Avenida Rio Negro 980
Alphaville
06400**Barueri** SP
Tel 429-3222

Hewlett-Packard do Brasil I e C Ltda
Rua Padre Chagas, 32
90000-**Porto Alegre**-RS
Tel (0512) 22-2998, 22-5621
Cable HEWPACK Porto Alegre

Hewlett-Packard do Brasil I E C Ltda
Rua Siqueira Campos, 53
Copacabana
20000-**Rio de Janeiro**
Tel 257-80-94-DDD (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
Bogotá, I D E
Tel 69-88-77
Cable AARIS Bogotá
Telex 044-400

COSTA RICA
Cientifica Costarricense S A
Avenida 2, Calle 5
San Pedro de Montes de Oca
Apartado 10159
San Jose
Tel 24-38-20, 24-08-19
Telex 2367 GALGUR CR
Cable GALGUR

ECUADOR
Calculators Only
Computadoras y Equipos Electrdnicos
P O Box 6423 CCI
Eloy Alfaro #1824,3 Piso
Quito
Tel 453482
Telex 02-2113 Sagita Ed
Cable Sagita-Quito

EL SALVADOR
Instrumentacion y Procesamiento Electronico de el Salvador
Bulevar de los Heroes 11-48
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
Av Periférico Sur No 6501
Tepepan, Xochimilco
Mexico 23, D.F.
Tel 905-676-4600

Hewlett-Packard Mexicana, S.A. de C.V
Ave. Constitucidn No 2184
Monterrey, N L
Tel 48-71-32, 48-71-84
Telex 038-410

NICARAGUA
Roberto Terán G
Apartado Postal 689
Edificio Terán
Managua
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 3483103 Curunda, Canal Zone
Cable ELECTRON Panama

PERU
Compañia Electro Médica S A
Los Flamencos 145
San Isidro Casilla 1030
Lima 1
Tel: 41-4325
Cable: ELMED 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