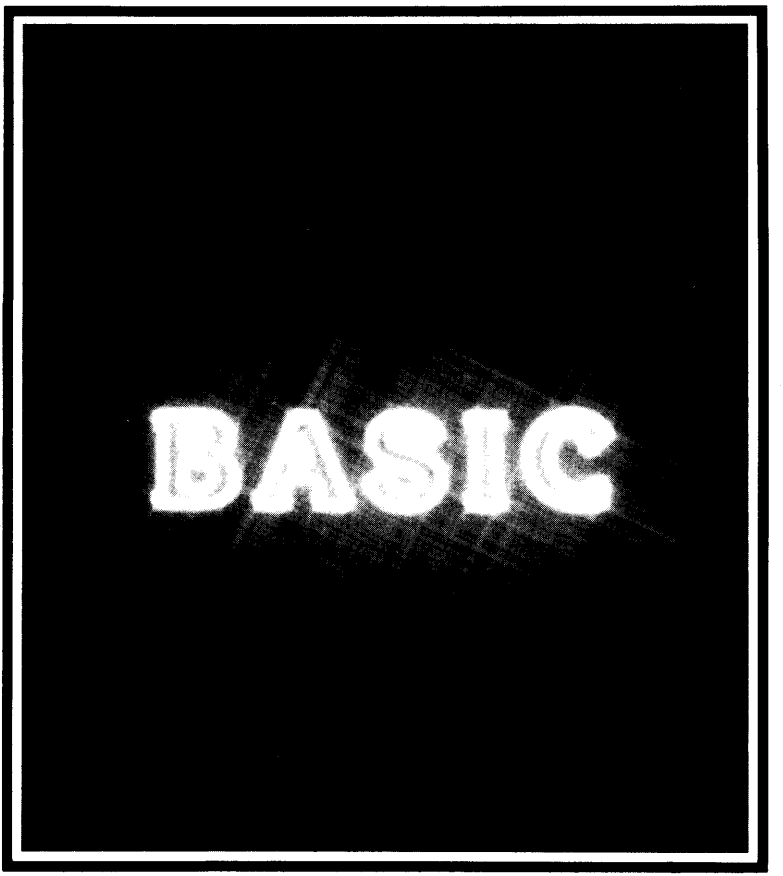
HP 9000 Series 200/300 Computers

HEWLETT PACKARD

BASIC 4.0 User's Guide



BASIC 4.0 User's Guide

for HP 9000 Series 200/300 Computers

Manual Reorder No. 98613-90041

Hewlett-Packard Company
3404 East Harmony Road, Fort Collins, Colorado 80525

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

June 1985...Edition 1
August 1985...Update
September 1985...Edition 1 with update merged.

98613-90041, rev. 9 85

Table of Contents

Chapter 4: Using Your Keyboard With BASIC

Chapter 5: Working With BASIC

Chapter 6: Talking to Peripherals With BASIC

Chapter 7: Special BASIC Features

Appendix A: Task Reference

The First Sixty Minutes

Chapter 1

The next few pages show how to install and configure BASIC in seven easy-to-follow steps. Follow the directions carefully and you will be doing productive work in sixty minutes or less (somewhat longer if you have a hard disc).

To proceed, you should have:

- Set up your computer according to its *Installation Card* or *Installation Guide.*
- Connected all peripherals, and installed all interface and accessory cards according to the instructions in the *Peripheral Installation Guide.*
- Filled out the BASIC System Worksheet provided with the *Peripheral Installation Guide.*

If you have trouble following any of the steps, lift the tab marked *Help!* for assistance.

Note

HP Series 200/300 Computers support three different keyboards. Keys for all three keyboards are shown in this order: HP 46020A, HP 98203B, and HP 98203A (e.g., Return, EXECUTE, or EXEC).

Press the key that is on **your** keyboard.

Now, let's move on.

Step 1:

Booting BASIC

Description

The process of loading a language system from a disc into your computer's memory is called **booting**.

This procedure describes how to boot BASIC.

Note

If you have the HP 98603 BASIC 4.0 ROM system, begin this procedure at step 5.

Procedure

1. Turn your computer off.
2. If your disc drive is not built into your computer, turn your disc drive on.
3. Find the disc labeled *System Disc*. If you don't know how to handle and care for flexible discs, lift the *Help!* tab.
4. Insert the *System Disc* into one of your flexible disc drives:
   - If using a micro flexible disc drive, insert the disc with the label side up and metal end forward.
   - If using a mini flexible disc drive, insert the disc with the label side up and the media window forward. Close the door.
5. Turn your computer on. (See the Special Instructions on the next page if you have another system program besides BASIC.)
6. Wait for the message:

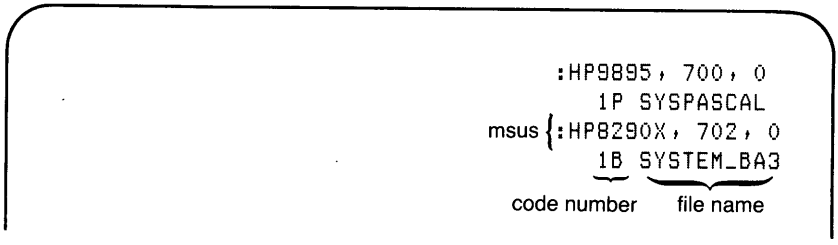
```
BASIC Ready 4.0  (c) Copyright HP 1985
```

BASIC has now been booted.

Special Instructions

If more than one system program is accessible to your computer, and you have boot ROM revision 3.0 or later, follow this procedure:

1. After turning your computer on in step 5, press the space bar a few times and wait for the list of available systems to appear on the right-hand side of the screen.
2. Choose the BASIC system from the list by typing its code number:
   - If you have only one BASIC system, its code number is probably **1B**.
   - If you have more than one BASIC system, choose the one you want by typing its code number. Distinguish between BASIC systems by their associated msuses, and by their file names.



Step 2:

Tailoring BASIC to Your Computer System

Description

The BASIC Language System includes a set of **binaries** that extend the capabilities of the system. Most binaries are not loaded when BASIC is booted; instead, you can selectively load binaries after booting to create a custom BASIC system. This segmented structure conserves memory–you choose only the binaries that you need.

Some of these binaries, called **drivers**, allow BASIC to access various interfaces and peripheral devices (e.g., HP-IB, screen, disc drive, etc.). Only two drivers are loaded when BASIC is booted: CRTA which allows BASIC to use a non-bit-mapped screen, and CRTB which allows BASIC to use a bit-mapped screen.

In this procedure, you will use the LOAD BIN statement to load the remaining drivers your system requires. In addition, if you are using only one type of screen–either bit-mapped or non-bit-mapped–you will remove the unnecessary CRT driver with the SCRATCH BIN statement.

Binaries in the BASIC 4.0 ROM

The HP 98603 BASIC 4.0 ROM system contains all drivers *except*:

- DCOMM
- SRM

Thus, if you have a ROM system and your hardware configuration does **not** require the DCOMM or SRM binaries, then you can skip this entire section.

Procedure

Note

If you have a non-US ASCII Keyboard, lift the *Help!* tab and read "Using Non-US ASCII Keyboards" in the *Help!* section for Step 2 before beginning this procedure.

1. If you have only one type of screen, either bit-mapped or non-bit-mapped, type:

   `SCRATCH BIN` (Return), (EXECUTE), or (EXEC)

   This removes the unneeded CRT driver.
2. Get the BASIC System Worksheet that you filled out while working through the *Peripheral Installation Guide.*
3. Find the disc labeled *Drivers Disc* (*Language Extensions and Drivers Disc* if you have double-sided micro-discs).
4. Remove the BASIC *System Disc* from the disc drive and insert the *Drivers* (or *Language Extensions and Drivers*) *Disc.*
5. Type:

   `LOAD BIN "`**driver**`"` (Return), (EXECUTE), or (EXEC)

   where **driver** is the name of the first driver in the "Drivers" column of the BASIC System Worksheet.

   Example: To execute this statement, just substitute the name of your first driver into the statement above. For example, suppose `DISC` is the name of the first driver on your worksheet. Then you would type:

   `LOAD BIN "DISC"` (Return), (EXECUTE), or (EXEC)
6. Wait for the following message to appear at the bottom of the screen:

   `BASIC` **driver** `4.0`

   where **driver** is the name of the driver you just loaded.
7. Repeat step 5 to load the remaining drivers. If a driver appears more than once on your worksheet, load it only once.

Step 3:

Tailoring BASIC to Your Programming Tasks

Description

Language extension files are another type of binary. Each language extension file contains a set of BASIC statements that extend the capabilities of the language. For example, the language extension file called GRAPH contains statements used in graphics programming. Without this file, programs containing these statements cannot be designed or run.

In this procedure, you will select the language extension files required for your particular programming tasks. Then you will load them using the LOAD BIN statement.

Binaries in the BASIC 4.0 ROM

The HP 98603 BASIC 4.0 ROM system contains all language extensions *except*:

- KNB2_0

Thus, if you have a ROM BASIC system and you do **not** need the KNB2_0 binary, then you can skip this entire section.

Procedure

1. Look over the list of language extension files given on the following two pages. Circle the files required for your programming tasks.
2. Find the disc labeled *Language Extensions Disc* (*Language Extensions and Drivers Disc* if you have double-sided micro-discs).
3. If you have separate *Language Extensions* and *Drivers* discs, remove the *Drivers Disc* from the disc drive and insert the *Language Extensions Disc*. If you have a combined *Language Extensions and Drivers Disc*, leave it in the disc drive.
4. Type:

   `LOAD BIN "`**filename**`"` Return, EXECUTE, or EXEC

   where **filename** is the name of the first language extension file you circled.

   Example: To execute this statement, just substitute the name of the first language extension file you circled into the statement above. For example, suppose you will be doing graphics programming, and you circled the `GRAPH` language extension file. Then you would type:

   `LOAD BIN "GRAPH"` Return, EXECUTE, or EXEC

5. Wait for the following message to appear at the bottom of the screen:

   `BASIC` **filename** `4.0`

   where **filename** is the name of the language extension file you just loaded.

7. Repeat step 4 to load the remaining language extension files that you circled.

Note

We recommend loading the ERR language extension file. It provides textual descriptions of errors, rather than just error numbers, and will save you the trouble of looking up errors every time they occur.

Language Extensions Files

The following language extension files reside on the *Language Extensions Disc* (with double-sided disc media, they are on the *Language Extensions and Drivers Disc*; with ROM BASIC systems, all language extensions except KNB2_0 are in the ROM and need not be explicitly loaded):

File Name **Definition**

CLOCK The TIMEDATE function is in core BASIC. CLOCK provides these functions and statements:

DATE	ON DELAY	TIME
DATE$	OFF DELAY	TIME$
ON CYCLE	ON TIME	
OFF CYCLE	OFF TIME	

ERR Provides text messages for errors.

GRAPH Provides the RATIO function and these commonly used graphics statements:

ALPHA ON	GRAPHICS ON	PEN
ALPHA OFF	GRAPHICS OFF	PENUP
AXES	GRID	PDIR
CLIP	GSTORE	PIVOT
CSIZE	IDRAW	PLOT
DRAW	IMOVE	PLOTTER IS
DUMP DEVICE IS	IPLOT	RPLOT
DUMP GRAPHICS	LABEL	SHOW
FRAME	LDIR	VIEWPORT
GCLEAR	LINE TYPE	WINDOW
GINIT	LORG	
GLOAD	MOVE	

GRAPHX Provides these more advanced graphics statements for graphics input, color plotting, and area filling:

AREA COLOR	PLOT (*)	SET LOCATOR
AREA INTENSITY	POLYGON	SET PEN COLOR
AREA PEN	POLYLINE	SET PEN INTENSITY
DIGITIZE	READ LOCATOR	SYMBOL
GESCAPE	RECTANGLE	TRACK IS ON
GRAPHICS INPUT IS	RPLOT (*)	TRACK IS OFF
IPLOT (*)	SET ECHO	WHERE

IO Provides the PPOLL and SPOLL functions and these statements used to interface with IO resources:

ABORT	OFF INTR	REMOTE
BREAK	ON SIGNAL	REQUEST
CLEAR	OFF SIGNAL	RESET
DISABLE INTR	PASS CONTROL	SEND
ENABLE INTR	PPOLL CONFIGURE	SIGNAL
LOCAL	PPOLL RESPONSE	TRIGGER
LOCAL LOCKOUT	PPOLL UNCONFIGURE	
ON INTR		

KBD Provides these statements for advanced use of the keyboard, including softkeys. KBD is required to use any HP-HIL device with the HP 46020A keyboard:

EDIT KEY	LOAD KEY	SCRATCH KEY
LIST KEY	RESTORE KEY	STORE KEY

LEX Provides this statement for changing collate sequences, and is required to convert keycodes to characters for local language keyboards:

LEXICAL ORDER IS

MAT Provides these statements and functions which let you handle arrays, or matrices:

BASE	MAT REORDER	RANK
DET	MAT SORT	REDIM
DOT	MAX	SIZE
MAT	MIN	SUM

MS Extends the CAT statement. It also provides these additional mass memory statements:

CHECKREAD	READ LABEL
PRINT LABEL	

PDEV Provides these commands and statements which are useful for writing and debugging programs:

CHANGE	LOADSUB FROM	TRACE ALL
COPYLINES	MOVELINES	TRACE OFF
FIND	SECURE	TRACE PAUSE
INDENT		

SRM Required for the HP 98629 Shared Resource Manager interface. DCOMM driver also required. SRM must be loaded before DCOMM **if** loading from SRM network. SRM also provides an alternate form of the PROTECT statement appropriate for the SRM. Loading SRM sets the system clock on computers that do not have the powerfail or battery-backed real-time clock feature. Provides these statements:

CREATE DIR	LOCK	UNLOCK

TRANS Enables use of buffers and provides these statements which support the TRANSFER statement:

ABORTIO	ON EOT	WAIT FOR EOR
ON EOR	OFF EOT	WAIT FOR EOT
OFF EOR	TRANSFER	

XREF Provides the XREF command which obtains a cross-reference listing of identifiers in a program or subprogram.

XNB2_0 When loaded, this program causes KNOBX to function as it did in BASIC 2.0/2.1. Do not load this BIN file until you fully understand its purpose and operation. Refer to the Knob section in Chapter 15 of the *BASIC Programming Techniques* manual for more information.

Step 4:

Initializing Discs

Description

All new discs–both flexible and hard–must be prepared for use by a process called **initialization**.

This procedure describes how to initialize your discs with BASIC.

Note

If you have a double-sided micro disc drive, the following procedure initializes discs using the default formatting option that has been established for double-sided micro disc drives. If you are initializing a single-sided flexible disc in your double-sided disc drive, or if you want to use a different formatting option, you need to override this default. Lift the *Help!* tab for instructions.

CAUTION

INITIALIZING DESTROYS ANY DATA AND PROGRAMS STORED ON A DISC. BE CERTAIN THAT YOU DO NOT INITIALIZE A BASIC SYSTEM DISC, OR ANY OTHER DISC CONTAINING VALUABLE FILES.

IF INITIALIZING A HARD DISC, DO NOT INTERRUPT THE INITIALIZATION PROCESS, AS THIS MAY SEVERELY DAMAGE THE DRIVE.

Procedure

1. **Remove all flexible discs from your disc drives.**
2. Select the disc to be initialized:
   - If initializing a flexible disc, make sure the disc is write-enabled (see *Help!* for this Step), then insert the disc into the disc drive. Look up the disc drive's msus and interleave factor in the Mass Storage Devices section of the BASIC System Worksheet.
   - If initializing a hard disc, look up its msus and interleave factor in the Mass Storage Devices section of the BASIC System Worksheet.

3. Type the appropriate INITIALIZE statement:

   - If the worksheet entry for interleave factor is blank, type:

     `INITIALIZE ":msus"` (Return), (EXECUTE), or (EXEC)

   - If the worksheet entry for interleave factor contains a value, type:

     `INITIALIZE ":msus",interleave` (Return), (EXECUTE), or (EXEC)

   where **msus** is the msus of the disc drive containing the disc to be initialized, and **interleave** is the recommended interleave factor.

   Example: To execute this statement, just substitute your msus and interleave factor into the statement above. For example, suppose you look on the BASIC System Worksheet and find:

   - The msus of your drive is `,700,1`.
   - The interleave factor for your drive is blank.

   Then you would type:

   `INITIALIZE ":,700,1"` (Return), (EXECUTE), or (EXEC)

   Another Example: If you look on your worksheet and find:

   - The msus of your drive is `,700`.
   - The interleave factor for your drive is `7`.

   Then you would type:

   `INITIALIZE ":,700",7` (Return), (EXECUTE), or (EXEC)

4. Wait several minutes (over an hour for a large hard disc) until the asterisk (*) disappears from the lower right corner of the screen, then remove the disc from the disc drive (if a flexible disc).

5. Repeat this process for each disc you want to initialize. You will need one initialized disc for each BASIC disc you received to make a back-up copy of your BASIC Language system (see the next Step).

Step 5:

Making a Back-up Copy of BASIC

Description

Protect your software investment by making a back-up copy of the BASIC Language System. Then if your working copy is destroyed, you can always make another.

You will need one blank, initialized flexible disc for each BASIC disc you received.

Two procedures are provided:

- Use Procedure 1 if you have two flexible disc drives.
- Use Procedure 2 if you have only one flexible disc drive.

Procedure 1:
Copying Discs with Two Drives

1. Select one of your BASIC discs. Make sure it is **write-protected** (see *Help!* for Step 4), then insert it into a flexible disc drive. Look up the msus of this drive in the Mass Storage Devices section of the BASIC System Worksheet. This is your **source msus**.
2. Insert a blank, initialized flexible disc into your other flexible disc drive. Look up the msus of this drive in the Mass Storage Devices section of the BASIC System Worksheet. This is your **destination msus**.
3. Type:

   `COPY ":`**source msus**`" TO ":`**destination msus**`"` Return, EXECUTE, or EXEC

   Example: To execute this statement, just substitute your source msus and destination msus into the statement above. For example, suppose you look on the BASIC System Worksheet and find:

   - the msus of the drive containing the original BASIC disc (source msus) is `,700,0`
   - the msus of the drive containing the blank, initialized disc (destination msus) is `,700,1`

   Then you would type:

   `COPY ":,700,0" TO ":,700,1"` Return, EXECUTE, or EXEC

4. Wait about one minute until the asterisk (*) disappears from the lower right corner of the screen.
5. Remove the original BASIC System disc from its disc drive. Remove the duplicate disc from its disc drive.
6. Label the duplicate disc with the same information that appears on the original disc. Write "Copy" on the label and record the date.
7. Repeat steps 1 through 6 for each original BASIC disc. Store the original discs in a safe place. Use the new duplicate discs for daily operation.

Procedure 2:
Copying Discs with One Drive

1. Type:

   `INITIALIZE ":MEMORY,0"` (Return), (EXECUTE), or (EXEC)

2. Select one of your BASIC discs. Make sure it is **write-protected** (see *Help!* for this step), then insert it into your flexible disc drive. Look up the msus of this drive in the Mass Storage Devices section of the BASIC System Worksheet.

3. Type:

   `COPY ":msus" TO ":MEMORY,0"` (Return), (EXECUTE), or (EXEC)

   where **msus** is the msus of your flexible disc drive.

   Example: To execute this statement, just substitute your msus into the statement above. For example, suppose you look on the BASIC System Worksheet and find that the msus of your drive is ,700,0. Then you would type:

   `COPY ":,700,0" TO ":MEMORY,0"` (Return), (EXECUTE), or (EXEC)

4. Wait a minute until the asterisk (*) disappears from the lower right corner of the screen.

5. Remove the original BASIC System disc from the disc drive and insert a blank, initialized disc.

6. Type:

   `COPY ":MEMORY,0" TO ":msus"` (Return), (EXECUTE), or (EXEC)

   where **msus** is the msus of your flexible disc drive.

7. Wait a minute until the asterisk (*) disappears from the lower right corner of the display.

8. Remove the duplicate disc from the disc drive and label it with the same information that appears on the original disc. Write "Copy" on the label and record the date.

9. Repeat steps 2 through 8 for each original BASIC disc. Store the original discs in a safe place. Use the new duplicate discs for daily operation.

Step 6:

Saving Your Customized BASIC System

Description

Instead of manually loading binaries each time you use BASIC, you can let BASIC load them for you. You can even create several custom configurations designed for different applications–one for general-purpose programming, another for graphics programming, as an example.

Two procedures are provided; we recommend that you:

- Use Procedure 1 if you have double-sided flexible discs, a hard disc, or an SRM connection.
- Use Procedure 2 if you have only single-sided flexible discs.

Procedure 1: Creating a System File

1. Type:

   `LIST BIN` (Return), (EXECUTE), or (EXEC)

   to get a list of all drivers and language extension files that you have loaded into your computer.
2. Check this list to make sure you have all of the drivers and language extension files you need. Load any that are missing (see Steps 2 and 3).
3. Get your disc ready:

   - If storing your custom system on a flexible disc, insert the disc into the disc drive. Look up the msus of this drive in the Mass Storage Devices section of the BASIC System Worksheet.
   - If storing your custom system on a hard disc, look up the msus of this drive in the Mass Storage Devices section of the BASIC System Worksheet.
4. Type:

   `STORE SYSTEM "SYSTEM_`**xxx:msus**`"` (Return), (EXECUTE), or (EXEC)

   where **xxx** is three or fewer letters that uniquely name your custom system, and **msus** is the msus of disc where you want to store your custom system.

   Example: To execute this statement, simply decide on a name for your system that begins with SYSTEM_[1] and look up the msus for your drive in the BASIC System Worksheet. For example, suppose you decide to name your system `SYSTEM_MY`, and the drive containing your disc has an msus of `,1400,0`. Then you would type:

   `STORE SYSTEM "SYSTEM_MY:,1400,0"` (Return), (EXECUTE), or (EXEC)

1 If you have boot ROM 3.0 or later, you may substitute SYS for SYSTEM_ in the system name, if you choose. This allows you to use seven identifying letters (instead of three) in the name to make the name more meaningful.

Examples of Custom System Names:

- **SYSTEM_IO** for an I/O programming environment (i.e., contains the IO language extension file).
- **SYSGRAPH** for a graphics programming environment on a computer with boot ROM 3.0 or later.

5. If you want to create another custom system:
   a. Type:

      SCRATCH BIN (Return), (EXECUTE), or (EXEC)

      to remove the current custom system from your computer.
   b. Load the drivers and language extension files that compose the next custom system (see Steps 2 and 3). (Note: if using both types of screen, reload either the CRTA or CRTB binary, which ever is missing. Use LIST BIN to find out.)
   c. Store this new system under a unique name.

Booting a Custom System

Boot a custom system as you do any other system:

- If you have only one custom system stored on a disc, insert the disc containing the desired system into the disc drive, and turn the computer off, then on again. The system will be booted.
- If you have more than one custom system stored on a disc, insert the disc containing the desired system into the disc drive. Turn the computer off, then on again, and press the space bar a few times. Select the custom system you want by typing its code number (see Step 1).

Procedure 2:
Creating an Autostart Program

1. Type:

   `LIST BIN` (Return), (EXECUTE), or (EXEC)

   to get a list of all drivers and language extension files that you have loaded into your computer. Write them down.
2. Check this list to make sure you have all of the drivers and language extension files you need. Load any that are missing (see Steps 2 and 3).
3. Make sure your working copy of the *System Disc* is **write-enabled** (see *Help!* for Step 4), then insert it into the same disc drive that you booted BASIC from.
4. Type:

   `EDIT` (Return), (EXECUTE), or (EXEC)

   to enter the BASIC Editor. The number `10` will appear on the screen.
5. Type the following Autostart program, pressing (Return), (ENTER), or (ENTER) after each line. (Note: The line numbers 10, 20, etc. appear automatically–you do not have to type them.)

Note

If you have BASIC on double-sided micro discs, change line 20 to:

```
20 PRINT "INSERT LANGUAGE EXTENSIONS AND DRIVERS DISC."
```

```
10 PRINT "REMOVE BASIC SYSTEM DISC."
20 PRINT "INSERT DRIVERS DISC."
30 INPUT "PRESS RETURN OR ENTER WHEN READY.",C$
40 LOAD "AUTOPROG"
50 END
```

6. Press unshifted (Stop), (PAUSE), or (PSE) to exit from the Editor.
7. Type:

   `STORE "AUTOST"` (Return), (EXECUTE), or (EXEC)

   to store the program on the *System Disc*. This AUTOST program is run automatically each time you boot BASIC.
8. Type:

   `SCRATCH` (Return), (EXECUTE), or (EXEC)

   to clear this program from your computer's memory.

9. Type:

   EDIT (Return), (EXECUTE), or (EXEC)

   to re-enter the BASIC Editor.

10. Type an AUTOPROG program similar to the one below, substituting the names of your drivers and language extension files into the appropriate program lines:

Note

If you have BASIC on double-sided micro-discs, omit lines 50, 60 and 70.

```
10 PRINT "PLEASE WAIT... LOADING DRIVERS."
20 LOAD BIN "DISC"
30 LOAD BIN "HPIB"
40 LOAD BIN "SERIAL"
50 PRINT "REMOVE BASIC DRIVERS DISC."
60 PRINT "INSERT BASIC LANGUAGE EXTENSIONS DISC."
70 INPUT "PRESS RETURN OR ENTER WHEN READY.",C$
80 PRINT "PLEASE WAIT... LOADING LANGUAGE EXTENSION FILES."
90 LOAD BIN "GRAPH"
100 LOAD BIN "ERR"
110 LOAD BIN "KBD"
120 PRINT "BASIC CONFIGURED"
130 END
```

11. Remove the *System Disc* and insert the *Drivers* (or *Language Extensions and Drivers*) *Disc*.

12. Type:

    STORE "AUTOPROG" (Return), (EXECUTE), or (EXEC)

    to store the program in a file called AUTOPROG on the *Drivers* (or *Language Extensions and Drivers*) *Disc*.

Starting BASIC with an AUTOST File

1. Boot as described in Step 1.
2. Insert and remove discs as instructed on the screen.
3. Wait for the message:

   BASIC CONFIGURED

   and remove the *Language Extensions* (or *Language Extensions and Drivers*) *Disc* from the disc drive.

Step 7:

Testing Your System

Description

Make sure that all of your peripheral devices are working properly with BASIC. The following procedures show you how to verify communication between the BASIC Language System and some common peripherals.

Testing Mass Storage Devices

You've already done this in the preceding steps. However, if you haven't tried all of your disc drives yet, here is a simple procedure to test them:

1. Look up the msus of the disc/tape drive in the Mass Storage Devices section of the BASIC System Worksheet.
2. Insert an initialized disc/tape cartridge into the drive. Remove all other disc/tape cartridges from your other drives.
3. Type:

   `MSI ":msus"` (Return), (EXECUTE), or (EXEC)

   to specify this drive as the default drive. All mass storage commands will use the default drive when no msus is specified.

   Example: To execute this statement, simply substitute the msus of your drive into the statement above. For example, suppose you look on the BASIC System Worksheet and find that the msus of your drive is ,702,1. Then you would type:

   `MSI ":,702,1"` (Return), (EXECUTE), or (EXEC)

4. Type:

   `CAT` (Return), (EXECUTE), or (EXEC)

If the drive's access light flashes and information is listed on the screen, your drive is working normally.

Testing Printers

1. Look up the device selector of your printer in the Printers section of the BASIC System Worksheet.
2. Check that your printer is turned on.
3. Type:

   `PRINTER IS` **device selector** (Return), (EXECUTE), or (EXEC)

   to specify your printer as the default printer.

   Example: To execute this statement, just substitute your printer's device selector into the statement above. For example, suppose you look on your worksheet and find that your printer's device selector is 701. Then you would type:

   `PRINTER IS 701` (Return), (EXECUTE), or (EXEC)
4. Type:

   `PRINT "HELLO"` (Return), (EXECUTE), or (EXEC)

If "HELLO" is printed on your printer, the printer is working normally.

Testing Plotters

1. Look up the plotter's device selector in the Plotters/Graphics Devices section of the BASIC System Worksheet.
2. Check that the plotter is turned on.
3. If you did not load the HP-IB and GRAPH language extension files into your custom system, do it now (see Step 3).
4. Type:

   PLOTTER IS **device selector**, "HPGL" (Return), (EXECUTE), or (EXEC)

   to specify the plotter as the default plotter.

   Example: To execute this statement, just substitute your plotter's device selector into the statement above. For example, suppose you look on your worksheet and find that your plotter's device selector is 705. Then you would type:

   PLOTTER IS 705, "HPGL" (Return), (EXECUTE), or (EXEC)

5. Type:

   PEN 1 (Return), (EXECUTE), or (EXEC)

   If the plotter picks up a pen, the plotter is working normally.

6. Type:

   PEN 0 (Return), (EXECUTE), or (EXEC)

   to replace the pen.

Testing HP-HIL Input Devices

1. If you did not load the KBD, GRAPH and GRAPHX language extension files into your custom system, load them now (see Step 3).
2. Type:

   EDIT (Return), (EXECUTE), or (EXEC)

   to enter the BASIC editor.
3. Type the following program, pressing (Return), (ENTER), or (ENTER) after each line:

```
10 GRAPHICS INPUT IS KBD, "KBD"
20 DIGITIZE X,Y
30 PRINT X,Y
40 GOTO 20
50 END
```

4. Press unshifted (Stop), (PAUSE), or (PSE) to exit from the Editor.
5. Digitize a few points:

   - If you have a mouse, move it around and occasionally press one of the buttons.
   - If you have a graphics tablet, press the stylus to the platten at various points.
   - If you have a touchscreen, touch the screen at various points.

   If two columns of numbers are listed on the screen, the device is working properly.
6. Press shifted (Stop), shifted (CLR I/O), or shifted (C I/O) to stop the program.

Testing the HP-HIL Mouse and Knob

1. If you did not load the KBD language extension file into your custom system, load it now (see Step 3).
2. Type, **without pressing** [Return], [EXECUTE], or [EXEC], the following line:

   ```
   THIS IS JUST A TEST LINE
   ```

3. Move the mouse left and right, or rotate the knob. If the cursor moves along the line, the device is working properly.
4. Press [Clear display], [SHIFT]-[CLR LN], or [CLR L] to clear the screen.

Testing the HP-HIL ID Module

1. If you did not load the KBD language extension file into your custom system, load it now (see Step 3).
2. Type:

   SYSTEM$("SERIAL NUMBER") [Return] [EXECUTE], or [EXEC]

   If the serial number is reported on the screen, the device is working properly.

Testing the HP-HIL Bar Code Reader

1. If you did not load the KBD language extension file into your custom system, load it now (see Step 3).
2. Set the switch on the bar code reader to keyboard mode. (See the bar code reader's manual for instructions).
3. Scan a sample bar code in the bar code reader's manual. If the code is displayed on the screen, the bar code reader is working properly.

HELP!

Chapter 2

Instructions for Getting Help

1. **Where did you have trouble?** Look up the number of the Step you need help with in the table below.
2. **Turn to the indicated page number.** There you'll find a list of topics for which help is provided. Choose your topic and turn to the indicated page number.

Help for Step 1:

Booting BASIC

Help is provided for:

- Disc care and handling: see page 27
- Inserting a disc into your disc drive: see pages 31, 36
- What to do if booting fails: see page 37
- Booting with multiple systems present: see page 38
- Identifying your boot ROM: see page 39

Disc Care and Handling

Your BASIC language system comes on 3½-inch flexible discs (micro-discs) or 5¼-inch flexible discs (mini-discs). The next several pages tell you how to use and care for your flexible discs. The first section is applicable to 3½-inch discs; the second section tells you about your 5¼-inch discs.

CAUTION

IF YOU ARE UNFAMILIAR WITH DISCS AND DISC HANDLING PROCEDURES, READ THE FOLLOWING INFORMATION CAREFULLY. OTHERWISE, YOU MAY DAMAGE YOUR DISCS.

The flexible disc is a thin piece of plastic enclosed in a special plastic jacket. The disc is covered with a thin oxide coating on which your program and data information are electronically stored.

When you insert the disc in the drive and close the door, the drive is ready to read information from or write information onto the disc. When the computer requests a read or write, the disc spins at a constant rate, like a phonograph record. The light on the disc drive indicates when reading or writing is taking place. Do not attempt to remove the disc when the light is on. Be sure to use only discs supplied or approved by HP. Other discs may not be of adequate quality or may damage the drive.

Unless you have a hard disc connected to your computer, you'll find that handling, inserting and removing flexible discs become an integral part of your daily routine. Even if you have a hard disc, you must deal with flexible discs initially (BASIC comes on flexible disc), and from time to time after that.

Since flexible discs represent a great deal of your money (the BASIC language system and other software you buy) and your time (the programs that you write), and contain your valuable records (data that you maintain), it is a very wise idea to treat them with considerable respect. This section tells you how to (and how not to) handle flexible discs. It also shows you how to insert and remove discs from your disc drive, which is no obvious matter if you have never done it before.

The following sections describe the handling and use of 3½-inch micro-discs and 5¼-inch mini-discs. Read only the section that pertains to your disc size.

3½-inch Micro-disc Handling and Use

This section describes the handling and use of 3½-inch micro-discs. If you're using 5¼-inch mini-discs instead, skip to the section titled "5¼-inch Mini-disc Handling and Use".

Handling 3½-inch Micro-discs

The flexible micro-disc is basically maintenance free, but is delicate and **must be handled carefully**. A good rule of thumb is to treat your disc as you would a valuable record album.

Here are some specific Do's and Don'ts to avoid loss of data or damage to your discs:

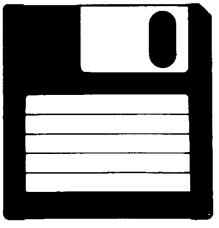
CAUTION

EVEN A LITTLE CARELESSNESS IN DISC HANDLING CAN DRAMATICALLY REDUCE THE LIFE OF A DISC.

Back Up Discs

There is always a chance of losing data anytime you are using your computer. There are many possible causes for this: a programming bug, operator error, power failure, hardware failure, or disc media failure from wearout, damage or contamination. **The only sure protection against data and program loss is to back up your discs.** Backing up discs is covered in Chapter 1.

Slide Disc Guard Over Media Window When Not in Use

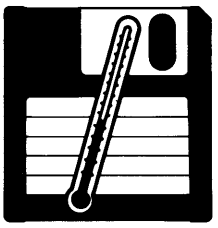


This is the single most important thing to remember about handling your disc. The disc guard protects the media against dust, fingerprints and scratches. **Never attempt to blow dust from the disc**. The disc's self-cleaning mechanism will remove most small particles from the media. When not in use the disc should be stored upright in a dust-free container, such as the box your discs were shipped in.

Operate Your Computer in a Clean Environment

Airborne contaminants and particles accidentally dropped onto the disc will cause your disc to wear out prematurely. Unreliable data storage and retrieval operations may also occur. Some of the most common contaminants are **dust, smoke, ashes, eraser crumbs and food particles. Never attempt to blow small particles from the disc.** Chemical vapors may also cause premature wearout.

Maintain Proper Temperature and Humidity

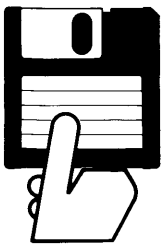


The proper operating range is 10°C (45°F) to 45°C (115°F) and 20% to 80% relative humidity. While temperature is usually easy to control, you may need to make special provisions to keep the humidity in the proper range. Although the disc will operate outside the normal humidity range, it will wear out more quickly and have a higher error rate.

Avoid Magnetic Fields

Data is stored on the disc magnetically, and can be erased by an external magnetic field. Avoid placing the disc near power transformers, magnets, large disc memories or motors.

Remove the Disc from the Drive When Not in Use



Remove the disc completely from the drive when you are through using it.

Use a Felt Tip Pen to Label Your Disc

Use a soft felt tip pen to label your disc, and be careful to write only in the label area. Using a hard tip pen, such as a ball point, can damage the media.

Don't Touch the Surface of the Disc

Be careful not to touch the media surface that is exposed through the media window. The thickness of a fingerprint is enough to lift the head off the disc and cause errors. The oil in a fingerprint will also collect dust and cause a disc to wear out prematurely.

Don't Bend or Fold the Disc

The disc is flexible but will not operate if creased. Using ball point pens, rubber bands, paper clips, etc., can crease the disc.

Don't Clean Your Disc

The inside surface of the disc jacket is covered with a special material that cleans the disc as it rotates in the drive. Any other method of cleaning may damage the media and cause data loss. **If a disc becomes dirty or scratched, immediately transfer the data to a new disc and dispose of the old one.**

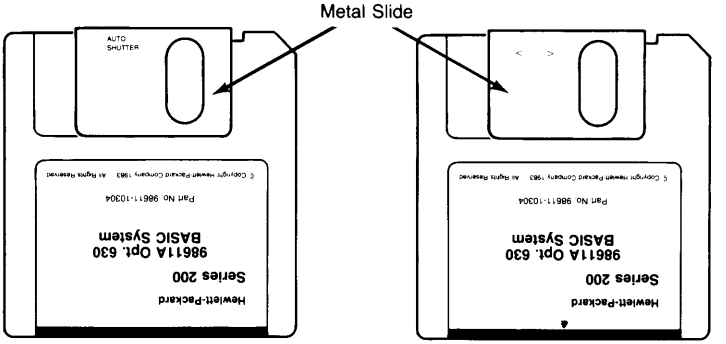
CAUTION

IF YOUR DISC DRIVE EVER DESTROYS THE MEDIA ON A DISC, STOP USING THE DRIVE UNTIL IT CAN BE SERVICED. THIS IS ESPECIALLY IMPORTANT, AS CONTINUED USE OF THE DRIVE WILL DESTROY MORE MEDIA. IMMEDIATELY CALL YOUR NEAREST HP SALES AND SERVICE OFFICE

Inserting and Removing 3½-inch Micro-discs

In this section, you will learn how to insert and remove 3½-inch micro-discs to and from your disc drive. If you're using 5¼-inch mini-discs, skip to the section titled "5¼-inch Mini-disc Handling and Use".

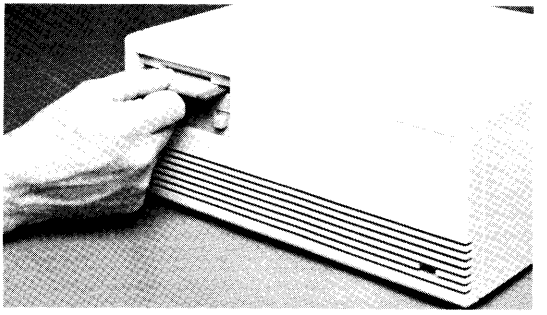
Micro-discs come in two varieties: "auto shutter" and "manual shutter." To find out which type you have, look at the metal slide. If the words "AUTO SHUTTER" appear just above the HP logo, you have an auto shutter disc. If a two-directional arrow appears, you have a manual shutter disc. See the following illustrations.



Auto Shutter Disc **Manual Shutter Disc**

Inserting Auto Shutter Micro-discs into Your Disc Drive

To insert an auto shutter disc, make sure the disc drive is turned on and slide the disc into the disc drive, label side up and metal slide facing the disc drive (try it!).



Inserting 3½-inch Micro-disc

The drive should completely "swallow" the disc and flash its light to signal acceptance. If the drive rejects the disc and pushes it back out at you, follow these steps:

1. Remove the disc and make sure you inserted it label side up. Also check that the disc drive is turned on. If you inserted the disc incorrectly, try it again.
2. If you inserted the disc correctly but the drive still refuses it, move the metal slide to the left, exposing the media. Reinsert the disc.

Note

When possible, turn the disc drive on before inserting a disc.

Inserting Manual Shutter Micro-discs into Your Disc Drive

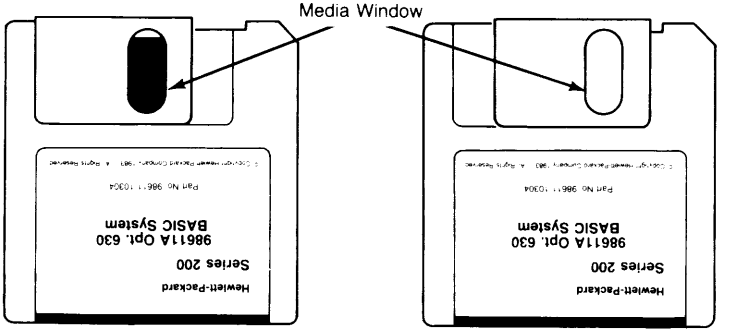
To insert manual shutter micro-discs into your disc drive, follow this procedure (try it!):

1. Make sure your disc drive is turned on.
2. Move the metal slide all the way to the left, exposing the media.
3. Insert the disc into the drive, label-side up and metal end facing the disc drive.

The drive should completely "swallow" the disc and flash its light to signal acceptance. If it pushes the disc back out at you, check that the label is facing up and the drive is turned on. Re-insert the disc.

Removing 3½-inch Micro-discs

To remove a 3½-inch disc from the disc drive, make sure the disc drive is turned on and press the disc eject button in the lower-right corner of the drive (try it!) The drive will immediately eject the disc. Close the metal slide over the media window if it is not already covered.



Media Exposed **Media Covered**

5¼-inch Mini-disc Handling and Use

This section describes the handling and use of 5¼-inch mini-discs. If you're using 3½-inch micro-discs instead, go back to the section titled "3½-inch Micro-disc Handling and Use".

Handling 5¼-inch Mini-discs

The flexible mini-disc is basically maintenance free, but is delicate and **must be handled carefully**. A good rule of thumb is to treat your disc as you would a valuable record album. Here are some specific Do's and Don'ts to avoid loss of data or damage to your discs:

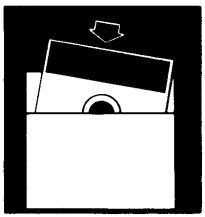
CAUTION

EVEN A LITTLE CARELESSNESS IN DISC HANDLING CAN DRAMATICALLY REDUCE THE LIFE OF A DISC.

Back Up Discs

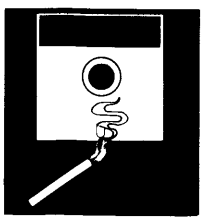
There is always a chance of losing data anytime you are using your computer. There are many possible causes for this: a programming bug, operator error, power failure, hardware failure, or disc media failure from wearout, damage or contamination. **The only sure protection against data and program loss is to back up your discs.** Backing up discs is covered in Chapter 1.

Return Disc to Storage Envelope When Not in Use



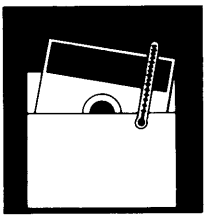
This is the single most important thing to remember about handling your disc. The storage envelope protects the media against dust, fingerprints and scratches. **Never attempt to blow dust from the disc.** The disc's self-cleaning mechanism will remove most small particles from the media. When not in use the disc should be stored upright in a dust-free container, such as the box your discs were shipped in.

Operate Your Computer in a Clean Environment



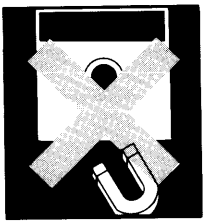
Airborne contaminants and particles accidentally dropped onto the disc will cause your disc to wear out prematurely. Unreliable data storage and retrieval operations may also occur. Some of the most common contaminants are **dust, smoke, ashes, eraser crumbs and food particles. Never attempt to blow small particles from the disc.** Chemical vapors may also cause premature wearout.

Maintain Proper Temperature and Humidity



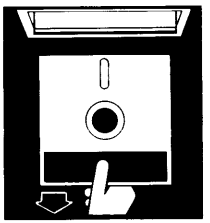
The proper operating range is 10°C (45°F) to 45°C (115°F) and 20% to 80% relative humidity. While temperature is usually easy to control, you may need to make special provisions to keep the humidity in the proper range. Although the disc will operate outside the normal humidity range, it will wear out more quickly and have a higher error rate.

Avoid Magnetic Fields



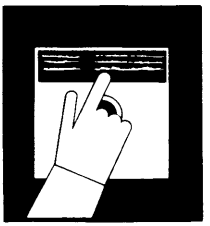
Data is stored on the disc magnetically, and can be erased by an external magnetic field. Avoid placing the disc near power transformers, magnets, large disc memories, or motors.

Remove the Disc from the Drive When Not in Use



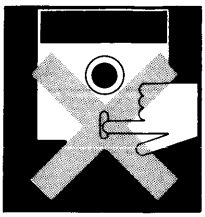
Remove the disc completely from the drive when you are through using it.

Use a Felt Tip Pen to Label Your Disc



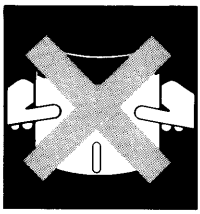
Use a soft felt tip pen to label your disc, and be careful to write only in the label area. Using a hard tip pen, such as a ball point, can damage the media. If possible, write on the large labels provided with your discs BEFORE applying them to the disc.

Don't Touch the Surface of the Disc



Be careful not to touch the media surface that is exposed through the media window. The thickness of a fingerprint is enough to lift the head off the disc and cause errors. The oil in a fingerprint will also collect dust and cause a disc to wear out prematurely.

Don't Bend or Fold the Disc



The disc is flexible but will not operate if creased. Using ball point pens, rubber bands, paper clips, etc., can crease the disc.

Don't Clean Your Disc

The inside surface of the disc jacket is covered with a special material that cleans the disc as it rotates in the drive. Any other method of cleaning may damage the media and cause data loss. **If a disc becomes dirty or scratched, immediately transfer the data to a new disc and dispose of the old one.**

CAUTION

IF YOUR DISC DRIVE EVER DESTROYS THE MEDIA ON A DISC, STOP USING THE DRIVE UNTIL IT CAN BE SERVICED. THIS IS ESPECIALLY IMPORTANT, AS CONTINUED USE OF THE DRIVE WILL DESTROY MORE MEDIA. IMMEDIATELY CALL YOUR NEAREST HP SALES AND SERVICE OFFICE.

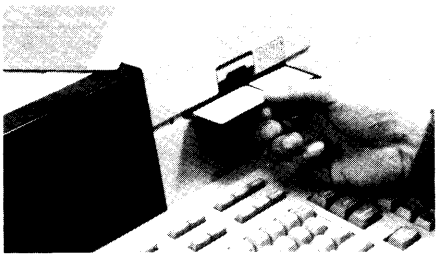
Inserting and Removing 5¼-inch Mini-discs

This section describes how to insert and remove 5¼-inch mini-discs to and from your disc drive. If you're using 3½-inch micro-discs instead, go back to the section titled "3½-inch Micro-disc Handling and Use".

Inserting 5¼-inch Mini-discs into Your Disc Drive

To insert a mini-disc into your disc drive, follow this procedure (try it!):

1. Make sure your disc drive is turned on.
2. Open the drive door by lifting the door handle. Check that there is not already a disc in the drive.
3. Insert the disc into the drive, label-side up with the media window facing the drive.



Inserting a 5¼-inch Mini-disc

4. Close the drive door. If the door will not close, push the disc farther into the drive and try again.

CAUTION

IF YOU ACCIDENTALLY INSERT ANOTHER DISC WHEN ONE IS ALREADY IN THE DRIVE, REMOVE THE BOTTOM DISC FIRST. OTHERWISE, THE READ/WRITE HEADS COULD BE DAMAGED.

Note

When possible, turn the disc drive on before inserting a disc.

Removing a 5¼-inch Mini-disc from Your Disc Drive

To remove a mini-disc from your disc drive, follow this procedure (try it!):

1. Make sure the drive is turned on.
2. Lift the drive door and carefully pull the disc out.
3. Return the disc to its protective envelope and store it upright in a dust-free box.
4. Close the drive door.

If Booting Fails...

If you do not get the message:

```
BASIC Ready 4.0
```

check for the following potential problems:

System Not Found

Your computer will keep cycling through its search routine until it finds a system. If the BASIC system fails to boot, or does not appear in the list of available systems when more than one system is present, something is preventing the computer from finding it. A few common causes of this problem are:

Problem	Solution
Drive containing system is not turned on.	Turn drive on. Computer will automatically find system.
System Disc is not inserted in drive.	Insert disc and press reset key.
Drive is not connected to computer.	Turn computer off, connect drive, and turn computer on.
Drive door is not closed.	Close drive door.

Not Enough Memory

You must have enough RAM in your computer to hold your BASIC Language System. If the computer runs out of memory while booting the system, the following message is displayed:

```
NOT ENOUGH MEMORY
```

To solve this problem, you must install another memory card as described in the *Peripheral Installation Guide.*

Booting with Multiple Systems Present

Note

The following procedure can only be done with boot ROMs of revision 3.0 and later.

The computer uses a searching sequence when looking for system programs and boots the first system it finds. If you have multiple system programs accessible to your computer, you will want to select the BASIC system to be booted.

1. Turn your computer off.
2. Insert the *System Disc* into one of your flexible disc drives:
   - If using a micro flexible disc drive, insert the disc with the label side up and metal end forward.
   - If using a mini flexible disc drive, insert the disc with the label side up and the media window forward. Close the door.
3. Turn your computer on and press the space bar a few times. This signals the computer that you want to override its default selection and choose which system to boot. This technique does not have an immediate effect, so don't be concerned if the computer doesn't seem to respond.
4. Review all systems found. The computer finds all systems it can and displays their names and where they were found in the upper right portion of the screen (see example below).

```
                                    :HP9895, 700, 0
                                       1P SYSPASCAL
                                    :HP8290X, 702, 0
                                       1B SYSTEM_BA3
```

5. Select your system. Beneath the name of the device where the system was found is a code followed by the name of the system file. This is the code you type to select the system you want. All systems except ROM-based systems have codes consisting of one or two numbers followed by a letter. The codes for ROMs are single letters.

   To choose a system, type its code. So, to boot BASIC in the example above, type `1B`. The code you type is displayed in the lower right corner of the screen. The computer boots the system and displays the following message at the bottom of the screen:

   ```
   BOOTING A SYSTEM
   ```

   When booting is completed, the following message appears at the bottom of the screen:

   ```
   BASIC Ready 4.0    (c) Copyright HP 1985
   ```

Identifying Your Boot ROM

To find out which revision of the boot ROM you have, remove all systems (including the BASIC *System Disc*) from your disc drives, and turn your computer on.

The initial display shows which boot ROM you have:

- If the initial display contains a line like this:

  `BOOTROM` **rev.**

  then you have boot ROM 3.0 or later. **rev.** tells you which revision you have, either, 3.0, 3.0L, 4.0 or 5.0.

- If the message:

  `851536 AVAILABLE BYTES` (number of bytes may vary)

  appears on the screen, you have boot ROM 1.0 or 2.0. The differences between 1.0 and 2.0 are not important for our purposes.

Help for Step 2:

Tailoring BASIC to Your Computer System

Help is provided for:

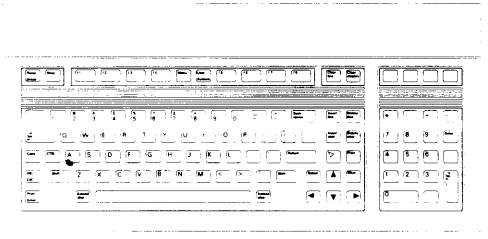
Typing and Executing Commands

Throughout this book you are told to type in a BASIC command and then press a key. For example, in Step 2 your are asked to type:

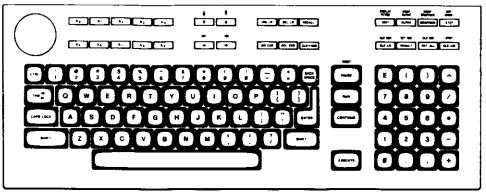
SCRATCH BIN (Return), (EXECUTE), or (EXEC)

To do this, you should type SCRATCH BIN at the keyboard, then press (Return), (EXECUTE), or (EXEC), depending on which type of keyboard you have.

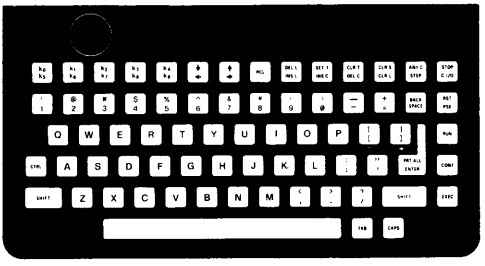
Three types of keyboards are supported on Series 200 and Series 300 Computers. They are:



HP 46020A



HP 98203B



HP 98203A

Match your keyboard to one of those shown. Then, when you see a list of keys following a command, press:

- The first one in the list if you have an HP 46020 keyboard
- The second one in the list if you have an HP 98203B keyboard
- The third one in the list if you have an HP 98203A keyboard

For example, when you see:

`SCRATCH BIN` **Return**, **EXECUTE**, or **EXEC**

Type `SCRATCH BIN` and press:

- **Return** if you have the HP 46020 keyboard
- **EXECUTE** if you have the HP 98203B keyboard
- **EXEC** if you have the HP 98203A keyboard

Removing the Unneeded CRT Driver

Two CRT drivers are loaded when BASIC is booted:

- CRTA, which allows BASIC to use a non-bit-mapped screen
- CRTB, which allows BASIC to use a bit-mapped screen

When you execute the SCRATCH BIN statement, all binaries are removed *except* the driver for the currently active screen.

SCRATCH BIN with One Screen Type

If you have only one screen, or if you have two screens but they are both of the same type (either both bit-mapped or both non-bit-mapped), SCRATCH BIN will automatically remove the CRT driver that you don't need.

SCRATCH BIN with Both Screen Types

If you have two screens, one bit-mapped and one non-bit-mapped, SCRATCH BIN will keep the CRT driver of the currently active screen, and remove the CRT driver of the inactive screen.

For example, if you are currently using the non-bit-mapped screen to issue commands and view output, SCRATCH BIN will keep the CRTA driver and remove the CRTB driver.

To reload the deleted CRT driver, do this:

1. Type:

   `LIST BIN` (Return), (EXECUTE), or (EXEC)

   to list all binaries on the screen.
2. Look over the list to find out which CRT driver is *not* currently loaded.
3. Execute the appropriate LOAD BIN statement to re-load the missing CRT driver, either:

   - `LOAD BIN "CRTA"` if CRTA is missing, or
   - `LOAD BIN "CRTB"` if CRTB is missing.

Substituting Values into Commands

In substep 5, you are told to substitute the name of your first driver into the command:

LOAD BIN "**driver**" (Return), (EXECUTE), or (EXEC)

This situation occurs frequently throughout this manual. Whenever you see a **bold** word imbedded in a command, do not type the word. Instead, substitute your own information in place of the word. The instructions immediately below the command will tell you how to make the substitution.

In substep 5, you should substitute the name of a driver (as found on your BASIC System Worksheet) in place of the bold word **driver**. If your first driver name is DISC, you would type:

LOAD BIN "DISC" (Return), (EXECUTE), or (EXEC)

Here is a good way to type such commands:

1. Type the command exactly as shown until you come to the bold word. In this example, you would type:

   ```
   LOAD BIN "
   ```

2. When you come to the bold word, make your substitution. In this example, the command would now look like this:

   ```
   LOAD BIN "DISC
   ```

3. Continue typing the command exactly as shown until you come to another bold word or you finish. In this example, there are no additional bold words, so you would finish the command:

   LOAD BIN "DISC" (Return), (EXECUTE), or (EXEC)

Responding to Error Messages

A few of the more common error messages are given below, along with their probable cause:

Error Number	Probable Cause
ERROR 56	File name is undefined. This means BASIC couldn't find the driver you just tried to load. Check that you spelled the name correctly, that the *Drivers* (or *Language Extensions and Drivers*) *Disc* is in the same drive that the *System Disc* was in, and that the drive is turned on.
ERROR 72	Drive not found or bad address. This probably means that the *Drivers* (or *Language Extensions and Drivers*) *Disc* is not in the same drive that the *System Disc* was in. Insert the *Drivers* (or *Language Extensions and Drivers*) *Disc* into the same drive that the *System Disc* was in when you booted BASIC.
ERROR 949	This symbol not allowed here. This means BASIC couldn't recognize what you typed as a valid command. Check that you typed LOAD BIN correctly, and typed the open quote before the driver name.
ERROR 985	Invalid quoted string. This means you probably forgot the close quote around the driver name.

Using Non-US ASCII Keyboards

The non-US ASCII keyboard operates as a US ASCII keyboard until the LEX binary on the *Language Extensions* (or *Language Extensions and Drivers*) *Disc* is loaded. The characters displayed correspond to U.S. keyboard positions without regard for the legend on the keycaps.

To configure your keyboard so that all characters displayed match the corresponding keycaps, you must load the LEX binary. Remove the BASIC *System Disc* and insert the *Language Extensions* (or *Language Extensions and Drivers*) *Disc*, then follow the procedure for your keyboard.

HP 46020 Keyboard

To display the "A" character with the French or Belgian keyboard, you must press [Q].

To display the "#" character with all keyboards, you must hold down the key identified by 1 in Figure 1, and press the key identified by 2.

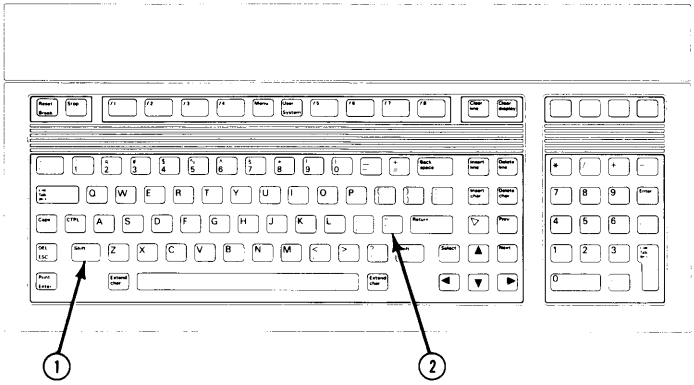


Figure 1

Type:

```
LOAD BIN "LEX" [Return]
```

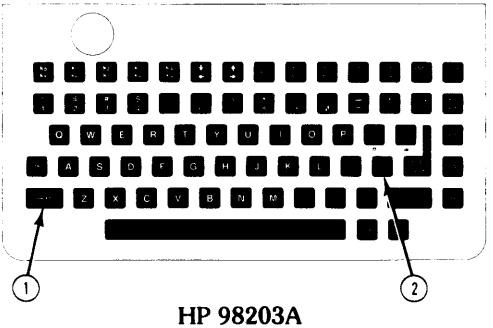
When the message:

```
BASIC LEX 4.0
```

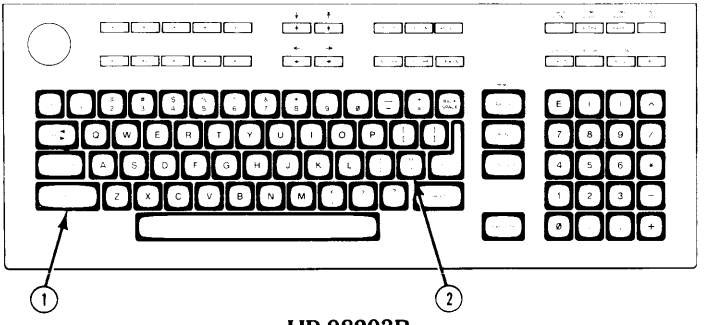
is displayed at the bottom of the screen, return to Step 2. The characters displayed will now match the legends on your keycaps.

HP 98203A/B Keyboard

To display the "[" character, you must hold down the key identified by 1 in Figure 2, and press the key identified by 2.



HP 98203A



HP 98203B

Figure 2

Type:

```
LOAD BIN "LEX"
```
EXECUTE or EXEC

When the message:

```
BASIC LEX 4.0
```

appears, return to Step 2. The characters displayed will now match the legends on your keycaps.

Loading Drivers

If you did not fill out your BASIC System Worksheet, or if you do not know which drivers to load, follow this procedure:

1. Look over the list of drivers on the next page. Circle the ones you need for your particular set of interfaces and peripherals.
2. Insert the *Drivers* (or *Language Extensions and Drivers*) *Disc* into the same drive that the *System Disc* was in when you booted BASIC.
3. For each driver you circled, type:

   `LOAD BIN "`**driver**`"` (Return), (EXECUTE), or (EXEC)

   where **driver** is the file name of one of the drivers you circled.

   Example: If the first driver you circled was `BUBBLE`, then your would begin by typing:

   `LOAD BIN "BUBBLE"` (Return), (EXECUTE), or (EXEC)

Driver Files

The following Driver files reside on the *BASIC Drivers* (or *Language Extensions and Drivers*) *Disc*:

File Name	Definition
BCD	Required for the HP 98623 Binary Coded Decimal interface.
BUBBLE	Required for the HP 98259 Magnetic Memory interface.
CRTA	Required for non-bit-mapped display. This BIN file is also part of SYSTEM_BA4 on the *BASIC System Disc* and is automatically loaded when the BASIC system is booted.
CRTB	Required for bit-mapped display. This BIN file is also part of SYSTEM_BA4 on the *BASIC System Disc* and is automatically loaded when the BASIC system is booted.
CS80	Required for CS80 disc drives. HPIB or FHPIB also required.
DCOMM	Required for the HP 98628 Datacomm or HP 98629 Shared Resource Management interface. If it's for the SRM interface, the SRM BIN file is also required. If loading **from** SRM, the SRM BIN file must be loaded before DCOMM.
DISC	Required for non-CS80 external disc drives. HPIB BIN file also required for HP 82901, HP 82902, HP 8290X, and HP 9135 drives. HPIB or FHPIB required for other disc drives.
EPROM	Required for the HP 98255 Erasable Programmable Read-Only Memory interface.
FHPIB	Required for the HP 98625 High-speed Disc interface. Achieves higher disc drive performance than HPIB BIN file when used with DISC or CS80 BIN files.
GPIO	Required for the HP 98622 General Purpose Input/Output interface.
HP9885	Required for the HP 9885 Disc Drive. The drive is connected to the HP 98622 GPIO interface.
HPIB	Required for the internal (built-in) HP-IB interface and for the HP 98624 HP-IB interface.
SERIAL	Required for the HP 98626 Asynchronous Serial Interface.

Help for Step 3:

Tailoring BASIC to Your Programming Tasks

Help is provided for:

- Choosing language extension files: see page 49
- Responding to error messages: see page 50

For Help on related topics, see:

- Typing and executing commands (see Help for Step 2)
- Substituting values into commands (see Help for Step 2)

Choosing Language Extension Files

If you are unfamiliar with Series 200/300 BASIC, you may need to do a little research before you select which language extension files to load. Here is a suggested procedure:

1. If you cannot decide whether you need a language extension file after reading the brief descriptions provided in Step 3, find the *BASIC 4.0 Language Reference* manual.
2. Look up the statements you think you might need in the *BASIC 4.0 Language Reference*. There you will find detailed information on the function of each statement.
3. If you need at least one of the statements in a language extension file, load that file using the procedure in Step 3.

Note

We recommend loading the ERR language extension file. It provides textual descriptions of errors, rather than just error numbers, and will save you the trouble of looking up errors every time they occur.

Responding to Error Messages

A few of the more common error messages are given below, along with their probable cause:

Error Number	Probable Cause
ERROR 1	This means you have tried to load the GRAPHX language extension file before loading the GRAPH language extension file. Simply LOAD BIN "GRAPH" first, then LOAD BIN "GRAPHX".
ERROR 56	File name is undefined. This means BASIC couldn't find the language extension file you just tried to load. Check that you spelled the name correctly, that the *Language Extensions* (or *Language Extensions and Drivers*) *Disc* is in the same drive that the *System Disc* was in, and that the drive is turned on.
ERROR 72	Drive not found or bad address. This probably means that the *Language Extensions* (or *Language Extensions and Drivers*) *Disc* is not in the same drive that the *System Disc* was in. Insert the *Language Extensions* (or *Language Extensions and Drivers*) *Disc* into the same drive that the *System Disc* was in when you booted BASIC.
ERROR 949	This symbol not allowed here. This means BASIC couldn't recognize what you typed as a valid command. Check that you typed LOAD BIN correctly, and typed the open quote before the language extension file name.
ERROR 985	Invalid quoted string. This means you probably forgot the close quote around the language extension file name.

Help for Step 4:

Initializing Discs

Help is provided for:

- Deciding whether to initialize a disc: see page 51
- Write-enabling and write-protecting flexible discs: see page 52
- Choosing double-sided micro disc formatting options: see page 55
- Initializing single-sided micro discs in double-sided disc drives: see page 56
- Responding to error messages: see page 57

For Help on related topics, see:

- Typing and executing commands (see Help for Step 2)
- Substituting values into commands (see Help for Step 2)

Deciding Whether to Initialize

You **must** initialize your disc if it is new, right out of the box.

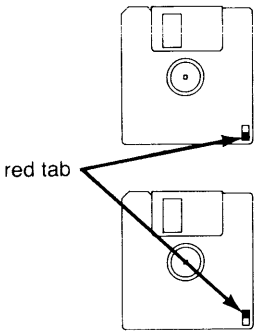
You **may** choose to initialize your disc if it contains no valuable files and you want to erase it. If it does contain some valuable files, copy them to another disc before initializing it.

You do not need to re-initialize your disc if has been previously initialized by the Series 200/300 Pascal Workstation.

Write-enabling/protecting Flexible Discs

3½-inch Double-sided Micro-discs

Gray double-sided micro discs should be write-enabled when you receive them. That means you can write data on the disc at any time. When you want to ensure that data on the disc is not accidentally overwritten or erased, you can write-protect the disc. The write-enabling/protecting mechanism works like this:



To write-enable the disc, make sure you can see through the write-protect hole in the corner of the disc–the red tab should **not** block the hole.

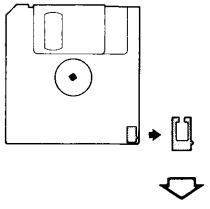
To write-protect the disc, make sure the red tab shows through the write-protect hole in the corner of the disc–you should not be able to see through the hole.

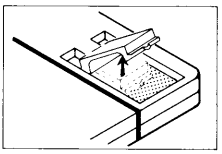
Double-sided Micro-disc Write-enabling/protecting

3½-inch Single-sided Micro-discs

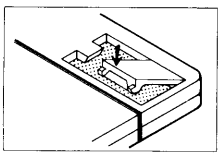
Blue single-sided micro-discs are write-enabled when you receive them. That means you can write data on the disc at any time. When you want to ensure that data on the disc is not accidentally overwritten or erased, you can write-protect the disc.

The write-enabling/protecting mechanism works like this:

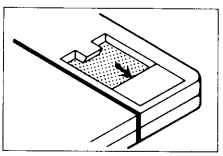




1. Weaken or score the attach point so as not to break the tab.
   Break off the write-protect tab.

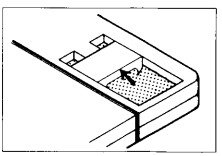


2. Align the protrusion on the tab with the groove in the disc.



3. Depress the tab into the groove–tab should fit snugly. In the position shown, the tab write-protects the disc.

Write-protect



4. To write-enable the disc, slide the tab up.

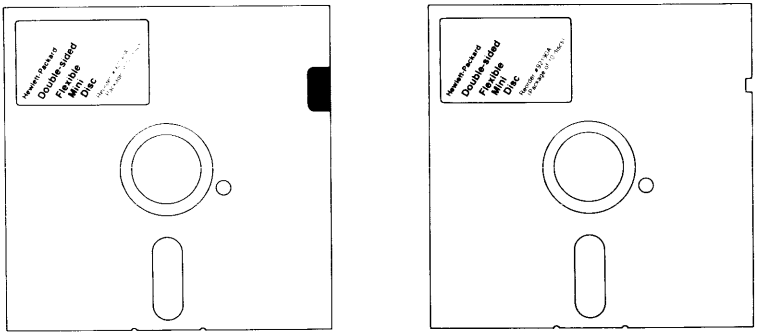
Write-enable

3½-inch Micro-disc Write-protection

5¼-inch Mini-discs

Covering or uncovering a notch in the disc jacket determines whether the disc drive can write information on the disc. When the notch is covered, it's impossible for the drive to write on the disc; thus information already on the disc is protected from being overwritten or erased.

Labels are provided with discs to allow you to cover the write-protect notch.



Write-protected **Write-enabled**

5¼-inch Mini-disc Write-protection

Choosing Double-sided Micro Disc Formatting Options

If you choose not to use the default formatting option, you can select select from among the following:

Format Option	Bytes/ Sector	Double-sided or Single-sided Formatting	Kbytes of Storage
0 (default)	256	Double-sided	630K
3[1]	1024	Double-sided	788K
4	256	Single-sided	270K

When you specify a formatting option in the INITIALIZE statement, you must also specify an interleave factor. Thus, the form of the INITIALIZE statement in this case is:

INITIALIZE ":**msus**",**interleave**,**format** (Return), (EXECUTE), or (EXEC)

where **msus** is the msus of the drive containing the disc to be initialized, **interleave** is the interleave factor, and **format** is the formatting option.

Examples:

- INITIALIZE ":**msus**",2,0 initializes the disc with interleave factor 2 and formatting option 0. Since the defaults for interleave factor and formatting option are 2 and 0, respectively, this is the same as INITIALIZE ":**msus**".
- INITIALIZE ":**msus**",2,3 initializes the disc with the default interleave factor of 2 and formatting option 3.
- INITIALIZE ":**msus**",2,4 initializes the disc with the default interleave factor and formatting option 4 (see next section).

1 Using this option will restrict the use of some BASIC statements (e.g., you cannot use TRANSFER with this option).

Initializing Single-sided Discs in Double-sided Disc Drives

To initialize a single-sided disc in a double-sided disc drive, you must use a formatting option of **4**. The INITIALIZE statement looks like this:

INITIALIZE ":**msus**",2,4 (Return), (EXECUTE), or (EXEC)

where **msus** is the msus of the drive containing the disc to be initialized.

Notice that you must also include the interleave factor in the INITIALIZE statement. In the example above, the recommended default of 2 is used.

Responding to Error Messages

A few of the more common error messages are given below, along with their probable cause:

Error Number	Probable Cause
ERROR 1	Configuration error. This means one of the drivers required to access the disc drive is missing. Check that you loaded DISC if the disc is not a CS/80 disc; check that you loaded CS80 if the disc is a CS/80 disc; check that you loaded HP9885 if using an HP 9885 disc drive.
ERROR 72	Drive not found or bad address. This means the disc drive specified by the msus in the INITIALIZE statement was not found. Check that the drive is turned on and check that you correctly typed the msus.
ERROR 80	Medium changed or not in drive. This means there is no disc in the disc drive specified by the msus in the INITIALIZE statement. Check that there is a disc in the drive, or check that you correctly typed the msus of the drive containing the disc.
ERROR 83	Write protected. This means the disc you are trying to initialize is write-protected. Write-enable the disc according to the instructions in *Help!* section for this Step.
ERROR 163	I/O interface not present. This means the driver required to access the disc drive's interface is not present. Check that you loaded HPIB if the disc is connected to a built-in HP-IB or HP 98624 HP-IB interface; check that you loaded FHPIB if the disc is connected to an HP 98625 Disc interface; check that you loaded GPIO if the disc is connected to a GPIO interface.
ERROR 949	This symbol not allowed here. This means BASIC couldn't recognize what you typed as a valid command. Check that you typed INITIALIZE correctly, and that you typed the open quote.
ERROR 985	Invalid quoted string. This means you probably forgot the close quote around the msus.

Help for Step 5:

Making a Back-up Copy of BASIC

Help is provided for:

- Responding to error messages: see page 58

For Help on related topics, see:

- Write-protecting flexible discs (see Help for Step 4).
- Typing and executing commands (see Help for Step 2)
- Substituting values into commands (see Help for Step 2)

Responding to Error Messages

A few of the more common error messages are given below, along with their probable cause:

Error Number	Probable Cause
ERROR 1	Configuration error. This means one of the drivers required to access the disc drive is missing. Check that you loaded DISC if the disc is not a CS/80 disc; check that you loaded CS80 if the disc is a CS/80 disc; check that you loaded HP9885 if the disc is an HP 9885 disc drive.
ERROR 2	Memory overflow. This means you do not have enough memory to perform this procedure. Instead, use the COPY or BACKUP utility. See the *BASIC Utilities Library Manual* for details.
ERROR 72	Drive not found or bad address. This means the disc drive specified by the msus in the COPY statement was not found. Check that the drive is turned on and check that you correctly typed the msus.
ERROR 80	Medium changed or not in drive. This means there is no disc in the disc drive specified by the msus in the COPY statement. Check that there is a disc in the drive, or check that you correctly typed the msus of the drive containing the disc.

ERROR 83 Write protected. This means you are trying to copy to a disc that is write-protected. Make sure the disc you are copying to (destination disc) is write-enabled, and check that you have not reversed the order of the source and destination msus in the COPY statement. You may be trying to copy the blank disc to the original BASIC disc.

ERROR 163 I/O interface not present. This means the driver required to access the disc drive's interface is not present. Check that you loaded HPIB if the disc is connected to a built-in HP-IB or HP 98624 HP-IB interface; check that you loaded FHPIB if the disc is connected to an HP 98625 Disc interface; check that you loaded GPIO if the disc is connected to a GPIO interface.

ERROR 949 This symbol not allowed here. This means BASIC couldn't recognize what you typed as a valid command. Check that you typed COPY correctly, and typed the open quote.

ERROR 985 Invalid quoted string. This means you probably forgot the close quote around the msus.

Help for Step 6:

Saving Your Customized BASIC System

Help is provided for:

- Choosing a System File or an Autostart File: see page 61
- Creating the AUTOPROG program (substep 10): see page 62
- Responding to error messages: see page 64

For Help on related topics, see:

- Write-protecting flexible discs (see Help for Step 4).
- Typing and executing commands (see Help for Step 2)
- Substituting values into commands (see Help for Step 2)
- Identifying your boot ROM revision (see Help for Step 1)
- Booting with multiple systems present (see Help for Step 1)

Choosing a System File or an Autostart File

A BASIC custom system file contains a custom BASIC system, including all binaries, stored in one file. It can be directly booted like any other system file.

An Autostart file contains a BASIC program that supervises the loading of binaries that compose a custom BASIC system. It performs, programmatically, all of the steps that you performed manually in Steps 2 and 3.

In general, system files are preferred over Autostart files because they load much more quickly and do not require you to swap discs. However, storing a complete custom BASIC system in a system file can require a lot of disc storage space. For this reason, you probably cannot store a custom system file on a single-sided flexible disc.

Therefore, use system files if you have enough disc space. If a STORE SYSTEM command results in the message:

```
ERROR 64  Mass storage medium overflow
```

You don't have enough room on the disc to store the system. In such instances, you will have to use the Autostart file approach.

Creating an AUTOPROG Program

When creating an AUTOPROG program, simply type the program exactly as shown in substep 10, with the following exceptions:

- Substitute the names of your drivers for the sample drivers given in lines 20 through 40.
- Substitute the names of your language extension files for the sample language extension files given in lines 90 through 110.

For example, suppose your custom system contains the following drivers:

- CRTA
- BUBBLE
- CS80
- FHPIB
- SERIAL

In addition, your system contains the following language extension files:

- ERR
- MAT

Your AUTOPROG program would look like this:

Note

If you have BASIC on double-sided micro-discs, you can omit lines 60, 70 and 80.

```
10 PRINT "PLEASE WAIT... LOADING DRIVERS."
20 LOAD BIN "BUBBLE"
30 LOAD BIN "CS80"
40 LOAD BIN "FHPIB"
50 LOAD BIN "SERIAL"
60 PRINT "REMOVE BASIC DRIVERS DISC."
70 PRINT "INSERT BASIC LANGUAGE EXTENSIONS DISC."
80 INPUT "PRESS RETURN OR ENTER WHEN READY.";C$
90 PRINT "PLEASE WAIT... LOADING LANGUAGE EXTENSION FILES."
100 LOAD BIN "ERR"
110 LOAD BIN "MAT"
120 PRINT "BASIC CONFIGURED"
130 PRINT "REMOVE BASIC LANGUAGE EXTENSIONS DISC."
140 END
```

Note that the CRTA driver is not included in the program because it is loaded automatically when BASIC is booted. There is no need to try to load it again.

Also note that you may add or delete LOAD BIN statements from the program depending on the number of binaries you need to load.

Responding to Error Messages

A few of the more common error messages are given below, along with their probable cause:

Error Number	Probable Cause
ERROR 1	Configuration error. This means one of the drivers required to access the disc drive is missing. Check that you loaded DISC if the disc is not a CS/80 disc; check that you loaded CS80 if the disc is a CS/80 disc; check that you loaded HP9885 if the disc is an HP 9885 disc drive.
ERROR 64	Mass storage medium overflow. This means there is insufficient space on your disc to store your custom system file. Either use a disc with more storage capacity, or use the Autostart file method (Procedure 2).
ERROR 72	Drive not found or bad address. This means the disc drive specified by the msus in the STORE SYSTEM statement was not found. Check that the drive is turned on and check that you correctly typed the msus.
ERROR 80	Medium changed or not in drive. This means there is no disc in the disc drive where BASIC is trying to store a file. If executing the STORE SYSTEM statement, check that there is a disc in the drive, or check that you correctly typed the msus of the drive containing the disc. If executing a STORE statement, make sure your disc is inserted into the same disc that the *System Disc* was in when you booted BASIC.

ERROR 83	Write protected. This means you are trying to store a file on a disc that is write-protected. Make sure the disc you are storing to is write-enabled.
ERROR 163	I/O interface not present. This means the driver required to access the disc drive's interface is not present. Check that you loaded HPIB if the disc is connected to a built-in HP-IB or HP 98624 HP-IB interface; check that you loaded FHPIB if the disc is connected to an HP 98625 Disc interface; check that you loaded GPIO if the disc is connected to a GPIO interface.
ERROR 949	This symbol not allowed here. This means BASIC couldn't recognize what you typed as a valid command. Check that you typed STORE SYSTEM or STORE correctly and typed the open quote.
ERROR 985	Invalid quoted string. This means you probably forgot the close quote around the file name and/or msus.

Help for Step 7:

Testing Your System

Help is provided for:

- What to do if the mass storage device test fails: see page 66
- What to do if the printer test fails: see page 67
- What to do if the plotter test fails: see page 67
- What to do if an HP-HIL device test fails: see page 67
- Responding to error messages: see page 68

For Help on related topics, see:

- Typing and executing commands (see Help for Step 2)
- Substituting values into commands (see Help for Step 2)

If the Mass Storage Test Fails...

Check that:

- You typed the correct msus
- There is an **initialized** disc or tape in the drive
- The drive is turned on and connected to the computer
- The HP-IB primary address is set correctly
- The msus is recorded correctly on the BASIC System Worksheet (see the *Peripheral Installation Guide*)
- Your display screen is turned on
- You have loaded the drivers required to access the drive and its interface (type LIST BIN to check)

If the Printer Test Fails...

Check that:

- You typed the correct device selector
- The printer is turned on and connected to the computer
- The printer's "on-line" light is on (if it has one)
- The HP-IB primary address is set correctly (if using an HP-IB printer)
- The device selector is recorded correctly on the BASIC System Worksheet (see the *Peripheral Installation Guide*)
- You have loaded the drivers required to access the printer and its interface (type LIST BIN to check)

If the Plotter Test Fails...

Check that:

- You typed the correct device selector
- The plotter is turned on and connected to the computer
- The plotter's "Chart Load" light is **not** on (if it has one)
- The HP-IB primary address is set correctly
- The device selector is recorded correctly on the BASIC System Worksheet (see the *Peripheral Installation Guide*)
- You have loaded the drivers required to access the plotter and its interface (type LIST BIN to check)

If an HP-HIL Device Test Fails...

Check that:

- You have loaded the KBD language extension file (type LIST BIN to check)
- You have correctly connected the HP-HIL device to the keyboard (see the *Peripheral Installation Guide*)

Responding to Error Messages

A few of the more common error messages are given below, along with their probable cause:

Error Number	Probable Cause
ERROR 1	Configuration error. This means one of the drivers required to access the device is missing. Check that you loaded all of the drivers required for the particular device. See the Loading Drivers section in the Help section for Step 2.
ERROR 72	Drive not found or bad address. This means the disc drive was not found. Check that the drive is turned on and check that you correctly typed the msus.
ERROR 80	Medium changed or not in drive. This means there is no disc in the disc drive that BASIC is trying access. Check that there is a disc in the drive, or check that you correctly typed the msus of the drive containing the disc.
ERROR 163	I/O interface not present. This means the driver required to access the disc drive's interface is not present. Check that you loaded HPIB if the disc is connected to a built-in HP-IB or HP 98624 HP-IB interface; check that you loaded FHPIB if the disc is connected to an HP 98625 Disc interface; check that you loaded GPIO if the disc is connected to a GPIO interface.
ERROR 949	This symbol not allowed here. This means BASIC couldn't recognize what you typed as a valid command. Check that you typed the statement correctly, and that you have loaded the language extension file required for the statement (e.g., you must have GRAPH to use the PLOTTER IS statement). Also check that you typed the open quote.
ERROR 985	Invalid quoted string. This means you probably forgot the close quote around the file name or string.

Introduction to BASIC

Chapter 3

What Is The BASIC Language System?

Just as we have a common set of words, or vocabulary, for communicating with one another, the computer has a set of words it recognizes and acts on to perform your operating and programmed tasks. Operating tasks include keyboard, display, and disc drive controls. Programmed tasks include input/output operations, decision making, and program debugging. The computer's set of words is called its language system. This guide describes operating your computer with the BASIC language system. BASIC is just that: an easy to learn, yet powerful language for both the keyboard and programming operations.

As with other programming languages, the BASIC language system assigns a unique "keyword" from its vocabulary to each computer instruction or task. For example, to start running a program in BASIC, you can either press the "run" key on the keyboard or type in the keyword `RUN` and press the "execute" key on the keyboard. To list the files on a disc, you type in the keyword `CAT` and press the "execute" key. (Don't worry about your keyboard labels now; "run" and "execute" are key functions, not labels, in this example.) The language system interprets each key pressed or keyword executed and performs a complex set of internal instructions to accomplish the task.

What Does BASIC Do?

The best way to demonstrate what the BASIC system does is to try using your computer without it.

Try It Yourself

Turn your computer on. Try typing some keys. Rotate the cursor control wheel if you have one. If you're lucky, you might stumble onto a few special keys that are operational, but for the most part, there is very little your computer can do. At the bottom of the screen you'll see the message:

```
SEARCHING FOR A SYSTEM
```

or

```
UNABLE TO FIND SYSTEM
RESET TO RETRY
```

The computer is telling you that it desperately needs a system program; without one, it is helpless.

The BASIC system is a master set of instructions that takes care of all the minute computing details we human beings would rather not worry about. As you saw in the demonstration above, one of the functions of the BASIC system is to define the operation of each key on the keyboard. The BASIC system also manages the exchange of information between the computer and its peripherals, refreshes the display on the CRT, reserves space for programs in random access memory (RAM), and performs several other functions too numerous to mention.

Besides these resource management duties, the BASIC system also supports a **human interface** that makes it easy for people to communicate with the computer and to design and run programs. The BASIC language system supports the BASIC programming language. By "supports" we mean that the BASIC language system interprets the BASIC language for the processor.

People and computer processors don't speak the same language. BASIC, although not exactly a "natural" language, is fairly easy for people to learn. It uses simple mathematical expressions and common English words in its statements. Processors, on the other hand, find BASIC far too complex for their weak intellects. Processors prefer **machine language**, with instructions expressed in terms of binary digits: ones and zeroes. But asking people to learn a hundred or so instructions that look like 0001110110011110 is asking a bit much.

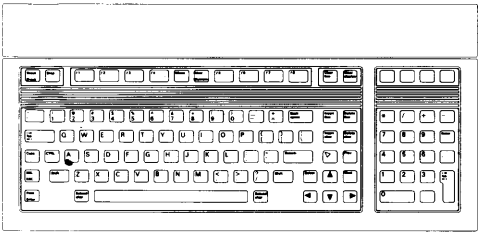
So the BASIC language system was invented to interpret between people and processors. The BASIC language system intercepts BASIC language statements, translates them into their equivalent machine language instructions, and hands these to the processor. This way, processors can speak one language, and people can speak another.

Using Your Keyboard With BASIC

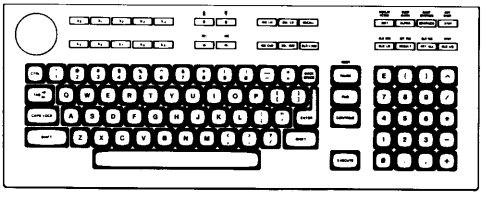
Chapter 4

Remember that the BASIC language system is a *system* program: it defines the keyboard, organizes the display, handles discussions between the processor and the peripherals–*plus* supports the BASIC language. In this chapter, we'll explain how the display is organized and help you become familiar with your keyboard.

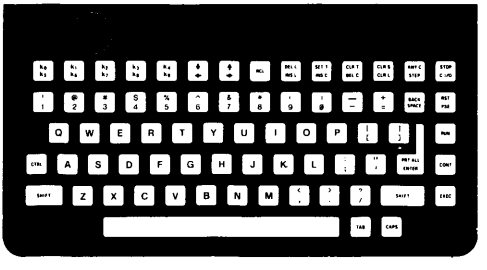
The BASIC system supports three keyboards:



HP 46020A



HP 98203B



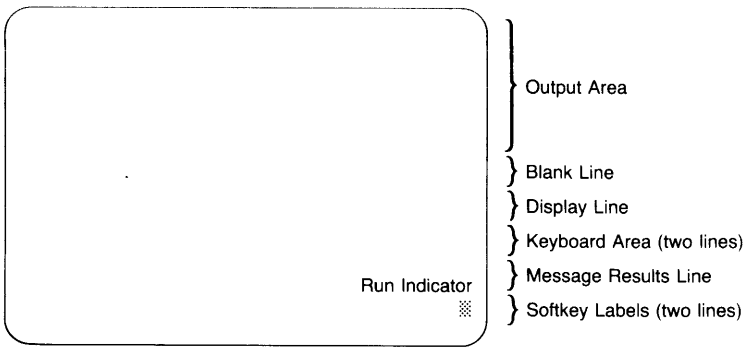
HP 98203A

After describing display organization, this chapter describes each of the keyboards and its key functions. The keyboards are described in the order in which they're shown in the preceding illustrations. Find the section that describes your keyboard and work through that section.

The exercises in this chapter assume you have already booted BASIC and loaded any required binaries as described in Chapter 1.

The BASIC Screen Organization

The BASIC language system partitions the screen into the following areas:

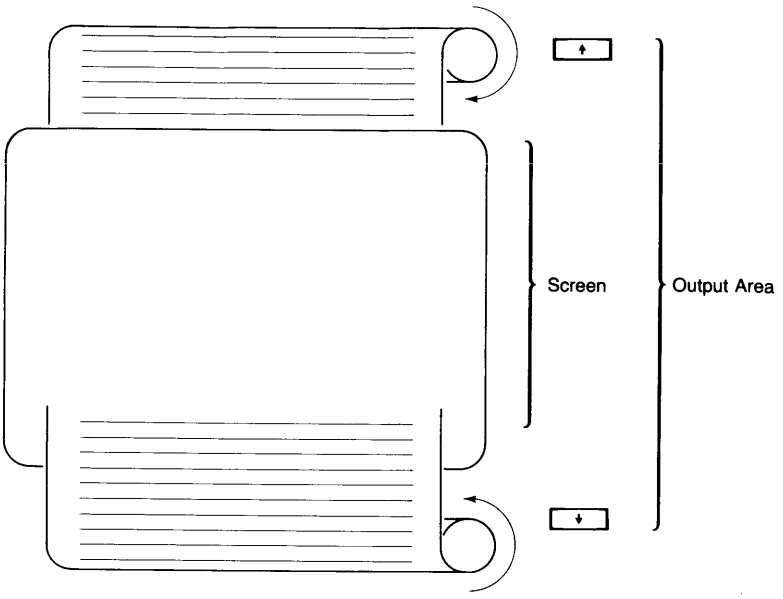


BASIC Display Organization

There are several types of screens that are supported by the BASIC language system. Screens differ in three major ways:

- Number of total lines that are displayed
- Number of characters per line
- Type of display, whether bit-mapped or non-bit-mapped

On some screens, the **output area** can hold more lines of information than can appear on the screen at any one time. Think of the output area as a large multi-line scroll that rolls up and down in front of a small multi-line window. To view lines that you can't presently see, you simply "scroll" a new set of lines in front of the window. This "scrolling" effect is accomplished with the arrow keys, the cursor control wheel, or the "mouse". When the entire output area is filled, each new line entered causes the top-most line to be permanently lost.



The Output Area

Running programs use the **display line** to communicate with the operator.

The **keyboard area** echoes whatever you type at the keyboard, up to two lines.

The **message/results line** displays answers to arithmetic and string operations typed at the keyboard, as well as error messages from the BASIC language system.

Try It Yourself

Note

Don't worry about making a mistake. You won't hurt anything. The computer beeps and gives an error message to help you understand what you did wrong. Check your typing, which must be precise.

Note

When three keys are shown together
(e.g., Clear display, CLR SCR or CLR S),
the first key is for the HP 46020A, the second for the HP 98203B, and the third for the HP 98203A. Press the key specified for your keyboard.

First press Clear display, CLR SCR, or CLR S. Now try this arithmetic problem:

```
98+26
```
Return, EXECUTE, or EXEC

Note

Whenever you see a key symbol
(e.g., Return, EXECUTE or EXEC)
following a line, press that key after typing the line.

The operation is first displayed in the keyboard area. When executed, the result `124` appears in the message/results line. To generate an error message in the message/results line, type:

```
98/0
```
Return, EXECUTE, or EXEC

Assuming you have loaded the ERR binary, the following message is displayed:

```
ERROR 31  Division by 0 or X MOD 0
```

The SYSTEM **softkey labels** are defined on booting BASIC with the HP 46020A. With the other keyboards, loading language extension file KBD enables user-defined softkeys.

The **run indicator** shows what the computer is doing at any given time. When the indicator is blank, the computer is free and awaiting your command. Run indicators are given in the following table:

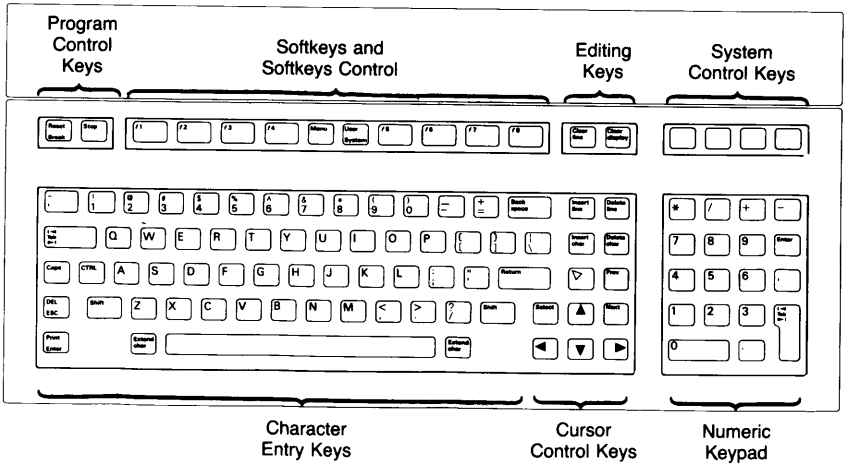
Symbol	Meaning
▒	Running a program.
–	Paused in a program.
blank	Stopped.
?	Waiting for a keyboard input.
*	Keyboard execution.
IO	Paused; waiting for an I/O operation to complete.

HP 46020A Keyboard Definitions

Note

If you do not have the HP 46020A keyboard, skip ahead to one of the following sections, which describe the HP 98203B and HP 98203A keyboards.

The keys on the HP 46020A keyboard are arranged into the following functional groups.



HP 46020A Keyboard

This section provides a handy reference guide to BASIC's key definitions for the HP 46020A keyboard. Keep in mind that other system programs may define the keys differently. Each key will be demonstrated where possible. One point to clarify: the **cursor** that we refer to in the following paragraphs is the blinking-underline that points to a location on the screen. (If you have a Model 237 computer or an HP 98700 Graphics Display Station, the cursor does not blink.)

Note

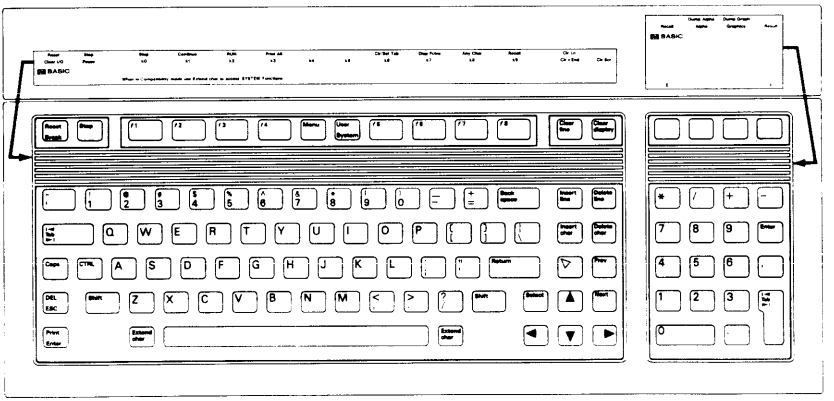
Before you proceed, type:

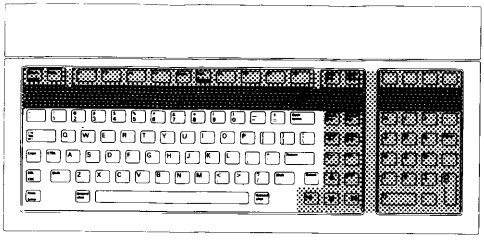
SCRATCH (Return), (EXECUTE), or (EXEC)

This clears the computer of any programs that might be left in memory from previous demonstrations.

BASIC Keyboard Overlays

Two keyboard overlays designed for the HP 46020A keyboard were included with your BASIC Language System. Place the overlays on the keyboard as shown below.





Character Entry Keys

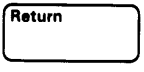
The character entry keys are arranged like a typewriter, but have some added features.

The Caps key sets the unshifted keyboard to either uppercase (which is the default after BASIC is booted) or lowercase (normal typewriter operation). The computer displays which mode the computer is in when you press the Caps key.

Type a few words, then press Caps and continue typing. Notice the case change. Press Shift-Clear line when finished.

You can enter standard uppercase and lowercase letters, using the Shift key to access the alternate case.

Type a few words, pressing Shift to change the case of the first letter of each word. Now press Caps and continue typing. Notice that the alternate case accessed by Shift depends on the setting of Caps. Press Shift-Clear line when finished.



The Return key has three functions:

1. When a running program prompts you for data, you respond by typing the requested data and then pressing Return. This signals the program that you have provided the data and that it can resume execution.
2. When typing in lines of a program, the Return key is used to store each line of program code.
3. After typing in a command, the Return key causes the command to be executed.

Type EDIT and press (Return). Notice the number 10 now displayed on the screen–this is the line number of the first line of a BASIC program. The computer is waiting for you to type in the line. Type:

```
!FIRST LINE
```

and press (Return). Notice that the computer accepts the statement as a program line and displays 20 in preparation for the next one. Press (Stop) when finished.

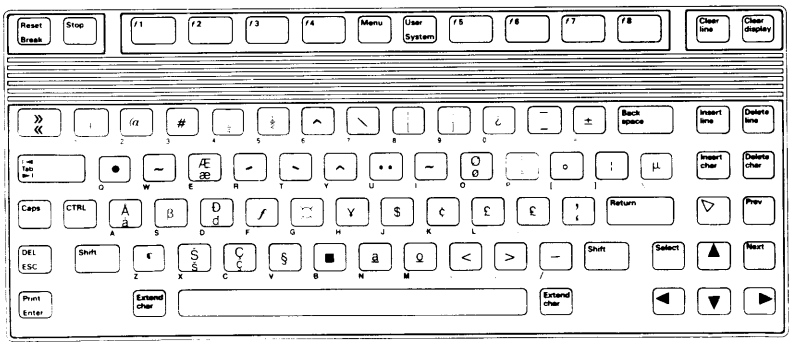
(Print / Enter)

Pressing (Enter) is the same as pressing the (Return) key.

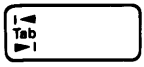
Pressing (Print) prints a complete copy of the alpha display on the default printer. The shifted version of the key directly above the (/) key in the numeric keypad (labeled Dump Alpha on the overlay) performs the same function.

(Extend char)

When pressed along with another key, this key allows you to generate the rest of the full 256-bit character set from the main typewriter section on Standard and European keyboards (see illustration). On a Katakana keyboard, the "Roman" and "Katakana" keys select the other character sets. To get Katakana characters 161 through 254 on a medium-resolution Series 300 screen, you must load the LEX language extension file.



Extended Character Set



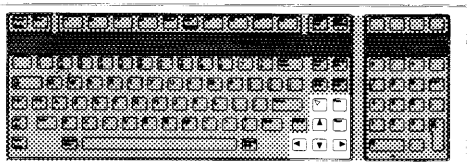
The Tab key moves the cursor forward to preset tabs. Pressing Shift-Tab moves the cursor back to preset tabs.

Before Tab can be used, a tab must be set. Tabs are set and cleared with System menu softkeys. The Tab key is demonstrated along with the **Set Tab/Clr Tab** softkey under "System Softkeys" later in this section.

The CTRL (control) key works like Shift to access a set of standard control characters, such as line-feed and form-feed. These characters are useful to the programmer for controlling some devices and for communicating with other computers. You probably won't need them when running programs. The available control characters are listed in the *BASIC Language Reference*, Useful Tables section.

The Select key beeps but performs no function unless it is program-defined.

Cursor-Control Keys



The cursor-control keys move the display cursor. The [▲] and [▼] keys allow you to scroll lines in the output area up and down. Shifted, the keys allow you to "jump" to the top and bottom of the output area. The [▶] and [◀] keys allow you move horizontally along a line. Shifted, they allow you to "jump" to the left and right limits of a line. The [Backspace] key works just like the [◀] key.

The unshifted [▽] key positions the print position at the beginning position on the page. The shifted [▽] key places the print position at the beginning of the first empty line in the display (scrolls up if necessary). In edit mode, pressing this key (shifted or unshifted) causes the computer to beep.

To verify operation of the [▽] key, press [Clear display]. Then type `PRINT "SOMETHING"` and press [Return]; repeat twice. You should now have the following display:

```
SOMETHING
SOMETHING
SOMETHING
```

Press the [▽] key (unshifted).

Type `PRINT "ANY "` and press [Return]. Your display should look like this:

```
ANY THING
SOMETHING
SOMETHING
```

Press [Clear display].

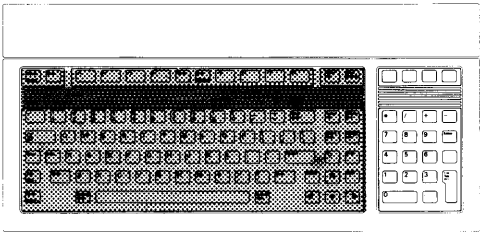
In normal mode, pressing the [Prev] key causes the display to scroll down one page and pressing the [Next] key causes the display to scroll up one page. In edit mode, these keys move the display one-half page.

To test the horizontal movement of the cursor, type a few words and press the shifted and unshifted [◀] and [▶] keys. Notice that the cursor cannot be moved beyond the characters you have typed. Press [Shift]-[Clear line] when finished.

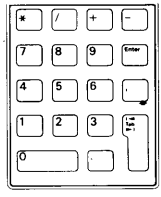
To test the vertical movement of the cursor, type `EDIT` and press [Return]. Now type the following lines, pressing [Return] after each line (the first line may be there already, so just press [Return] to accept it):

```
10 !FIRST LINE
20 !SECOND LINE
30 !THIRD LINE
40 !FOURTH LINE
```

Try out the shifted and unshifted [▲], [▼], and [▷] keys. Then try the [Prev] and [Next] keys. When you're done, press [Stop] to exit. Then, type `SCRATCH` [Return] to clear memory.



Numeric Keypad



The numeric keypad provides a convenient way to enter numbers and perform arithmetic operations. Simply type in the arithmetic expression you want to evaluate, then press [Enter]. The result is displayed in the lower-left corner of the screen.

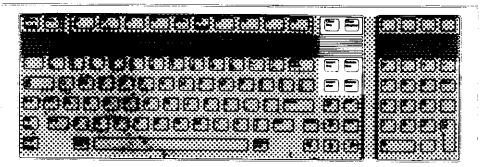
The [Enter] key performs the same function as the [Return] key. The [Tab] key on the numeric pad functions like the [Tab] key in the character entry area. The shifted versions of the [*], [/], [+], and [-] keys are E, (,), and ^, respectively (see labels on the overlay). The shifted versions are also available in the character entry area.

Type in the following problem using the numeric pad:

```
(26+14)/4
```

Now press [Enter] to perform the calculation. The answer, `10`, is displayed in the lower-left corner of the screen.

Editing Keys



The editing keys put easy character editing and line editing at your fingertips.

Pressing (Insert line) inserts a new line above the cursor's current position (edit mode only).

Type EDIT, then press (Return). Type in this line (if it isn't already there):

```
10 !FIRST LINE
```

Now, with the cursor somewhere on line 10, press (Insert line). Notice that a new line number (1) is inserted before line 10. Press (Stop) when finished.

Pressing (Delete line) deletes the line containing the cursor (edit mode only).

Type EDIT, then press (Return). Position the cursor to the line:

```
10 !FIRST LINE
```

and press (Delete line). The line is removed. To restore it, press the key directly above (*)(labeled Recall on the overlay) to recall it, then press (Return) to enter it into the program. Press (Stop) to exit edit mode.

Pressing (Insert char) sets insert mode, allowing you to insert characters to the left of the cursor. Press the key a second time to cancel insert mode.

Carefully type the following line exactly as shown:

```
THIS IS A TEST .
```

Position the cursor under the period and press [Insert char]. Now type:

```
OF INSERT MODE
```

and press [Insert char] again. The line should now look like this:

```
THIS IS A TEST OF INSERT MODE.
```

The new characters were inserted to the left of the period. Press [Shift]-[Clear line] when finished.

Pressing [Delete char] deletes the character at the cursor's position.

Type a few words and experiment with [Delete char], positioning the cursor at various places on the line. Notice that if you hold the key down, characters are deleted until you release it. Delete all of the characters you typed.

Pressing unshifted-[Clear line] (labeled Clr → End on the overlay) clears from the current cursor position to the end of the line.

Pressing [Shift]-[Clear line] (labeled Clr Ln on the overlay) clears the keyboard line and message/results line.

Type in a few words and use the [◀] key to position the cursor in the middle of the line. Press unshifted-[Clear line] to clear to the end of the line. Press [Shift]-[Clear line] to clear the rest of the line.

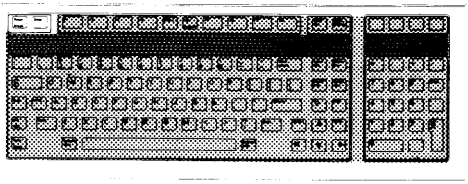
Pressing either the shifted or unshifted version of [Clear display] clears the entire alpha screen.

Type the following BASIC command:

```
PRINT "PUT THIS MESSAGE IN THE OUTPUT AREA."
```

Now press [Return] to execute it. Press the key directly above [*] (labeled Recall on the overlay) to recall the command, and press [Return] again. Repeat this step several times to fill the screen with messages. Now press [Clear display] to erase all lines at once.

Program Control Keys



The following keys allow you to control execution of the program stored in the computer's memory.

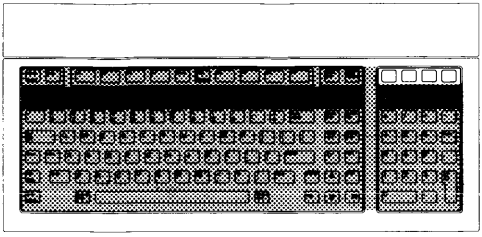
Pressing unshifted-Stop (labeled Pause on the overlay) **pauses** program execution after the current line. Pressing **Continue** (unshifted f2) with System menu active resumes program execution from the point where it was paused.

Pressing Shift-Stop (labeled Stop on the overlay) stops program execution after the current line. To restart the program, press **RUN** (unshifted f3) with System menu active.

Pressing Break (labeled Clr I/O on the overlay) pauses program execution when the computer is performing or trying to perform an I/O operation. Press Break instead of unshifted Stop when the computer is hung up on an I/O operation, since unshifted Stop works only after the computer finishes the current program line. Pressing Break cancels the I/O operation and pauses the program at the current line.

Pressing Reset pauses program execution immediately without erasing the program from memory. The `BASIC Reset` message indicates the computer is ready for your command.

System Control Keys



Four unlabeled keys directly above the numeric keypad control various system functions related to the display, printer, and editing operations. Most of these keys execute their functions immediately, as the key is pressed.

To easily identify the keys in the following description, we'll use this convention:

- Key 1 - Above the [*] key (labeled Recall on the overlay).
- Key 2 - Above the [/] key (labeled Alpha/Dump Alpha on the overlay).
- Key 3 - Above the [+] key (labeled Graphics/Dump Graph on the overlay).
- Key 4 - Above the [-] key (labeled RES on the overlay).

Key 1 - Recall

Pressing unshifted-Key 1 (Recall) recalls the last line that you entered, executed, or deleted. Several previous lines can be recalled this way. Recall is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again.

Type:

```
PRINT "1"  [Return]
```

to print the number 1 on the screen. Now press Key 1 to recall the print statement. Edit the statement to print the number 2 by positioning the cursor under the 1 and typing [2] over it. Press [Return] again. Now press Key 1 several times to see all of the statements it remembers. Then press [Clear display] when finished.

[Shift]-Key 1 moves forward through the recall stack.

Pressing [f8] with System menu active performs the same recall function as Key 1.

Key 2 - Alpha/Dump Alpha
Pressing unshifted-Key 2 (Alpha) once turns on the alphanumeric display. Pressing it the second time turns off the graphics display. This key function requires that the GRAPH BIN file be loaded. If you have a Model 237, an HP 98700 Graphics Display Station, or Series 300 computer, this key may perform no function.

Pressing Shift-Key 2 (Dump Alpha) prints a complete copy of the alpha display on the default printer. The Dump Alpha function is also executed by Print.

Key 3 - Graphics/Dump Graph
Pressing unshifted-Key 3 (Graphics) once turns on the graphics display. Pressing it the second time turns off the alphanumeric display. If you have a Model 237, an HP 98700 Graphics Display Station, or Series 300 computer, this key may perform no function.

Pressing Shift-Key 3 (Dump Graph) prints a complete copy of the graphics display on the default printer. If you have a Model 237, an HP 98700 Graphics Display Station, or Series 300 computer, the combined alpha and graphics display is printed.

Both key functions require that the GRAPH language extension file be loaded.

Key 4 - RES
Pressing Key 4 (RES) either shifted or unshifted returns the result of the last arithmetic expression that was executed.

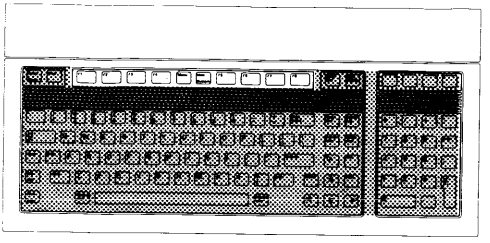
Press Shift-Clear line, then type:

```
23+45 Return
```

The result, `68`, is displayed in the lower left corner of the screen. To add 123 to this value, press Key 4 and type:

```
+123 Return
```

The new result, `191`, is now displayed. Press Shift-Clear line when finished.



Softkeys and Softkeys Control

There are eight softkeys (labeled f1 through f8) and two keys that control the definitions of the softkeys (Menu and System).

When the BASIC system is booted, the softkeys default to System mode. The System mode menu that appears at the bottom of your display is shown. System softkeys are defined following control key definitions. In addition to the System mode, there are also three User modes: User 1, User 2, and User 3. *BASIC Programming Techniques*, Chapter 2, describes how to set up User modes.

Softkeys Control Keys

User
System

Pressing unshifted-System causes softkeys to assume System mode. The System menu is displayed, **if** the Menu key is toggled to the "on" position.

Pressing Shift-User puts the softkeys in User 1 mode. The User 1 menu is displayed **if** the Menu key is toggled to the "on" position.

Menu

Pressing unshifted-Menu toggles the softkey labels—turns them on if they're off and turns them off if they're on.

Pressing Shift-Menu increments User mode and menu **if** User mode is "on".

User menus are blank unless language extension file KBD is loaded.

Try It Yourself

Let's get familiar with the two control keys.

First we want to get the System mode selected and menu displayed. If the System menu is displayed, continue with the next paragraph. If it is not displayed, press (System). If it is still not displayed, press (Menu).

With the System menu displayed, press unshifted-(Menu) several times. The System menu display should go on and off. Leave the System menu displayed, and continue.

Now press (Shift)-(User). The User 1 menu should appear on your display.

Press (Shift)-(Menu) several times. The displayed menus should rotate successively through the three User menus (User 1→ User 2→User 3→User 1→User 2, etc.).

Press unshifted-(Menu) several times and the last User menu goes on and off. Leave the User menu on.

Finish this exercise by pressing unshifted-(System) to get your computer back in System mode.

System Softkeys

The following paragraphs define the eight System softkeys.

Step (unshifted-(f1))

Step allows you to execute one program line at a time. This is particularly useful for debugging (fixing) programs.

Continue (unshifted-(f2))

Continue resumes program execution from the point where it was paused (by an unshifted-(Stop)).

RUN (unshifted-(f3))

RUN starts a program running from the beginning.

Print All (unshifted-f4)

The **Print All** key turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device. Press Print once to set printall "on" and again to set printall "off". An asterisk (*) appears next to All to indicate that printall is "on".

The display's output area is the default printall device at powerup. The *BASIC Programming Techniques* explains how to select other printall devices.

Press **Print All** to turn on printall mode. Now type in the following command:

```
PRINT "THIS IS A KEYBOARD OPERATION" Return
```

Both the PRINT command and the message itself are displayed on the screen, which is the default printall device. Now type:

```
THIS WILL CAUSE AN ERROR Return
```

Because this is not an executable BASIC statement, an error message is displayed, both at the bottom of the screen and in the printall area at the top. This way, a log is produced of all commands typed and executed at the keyboard, along with any error messages. Press Clear display to clear the display, and press **Print All** to turn off printall mode.

Set Tab/Clr Tab (f5)

Set Tab (unshifted-f5) sets a tab at the cursor's current position. Tabs remain in effect until cleared by either **Clr Tab** or the SCRATCH A statement (explained in *BASIC Programming Techniques*, Chapter 2).

Clr Tab (Shift-f5) clears a tab previously set at the cursor's position.

Press the space bar to move the cursor forward a few spaces and press **Set Tab**. Move the cursor back several spaces using ◀, then press Tab. Move the cursor forward several more spaces with the space bar, then press Shift-Tab. To clear the tab, move the cursor to the unwanted tab position and press **Clr Tab**. Press Shift-Clear line when finished.

Display Fctns (unshifted-f6)

Display Fctns sets the display-functions mode, allowing you to see special control characters (e.g., form-feed, carriage return) on the screen. Pressing this key a second time cancels the display-functions mode. An asterisk (*) appears next to Fctns to indicate that display-functions mode is "on".

Type the following line:

```
PRINT "DISPLAY-FUNCTIONS MODE OFF"  Return
```

Notice the display at the top of the screen. Now press **Recall** (unshifted-f8) to recall the line, and edit it to read:

```
PRINT "DISPLAY-FUNCTIONS MODE ON"
```

Press **Display Fctns**, and then press Return. Notice that the carriage return (CR) and line-feed (LF) control characters are now displayed. Press **Display Fctns** again to exit display-functions mode. Press Clear display when finished.

Any char (unshifted-f7)

Any char is used to find any ASCII character. First press **Any char**. The following message appears above the menu:

```
Enter 3 digits, 000 to 255
```

Enter a three-digit number from 000 through 255 representing the decimal equivalent of an ASCII character. The computer automatically displays the character on the screen. For a list of characters and their equivalent decimal values, see the US ASCII Character Codes table in the Useful Tables section of the *BASIC Language Reference*.

Press **Any char**, then type 189 which is the decimal equivalent of "§". The display line now displays "§". Press Shift-Clear line to erase it.

Recall (f8)

The **Recall** softkey (unshifted-f8) acts just like System Control Key 1 (described earlier). **Recall** recalls the last line that you entered, executed, or deleted. Several previous lines can be recalled this way. **Recall** is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again.

Type:

```
PRINT "1"  Return
```

to print the number 1 on the screen. Now press **Recall** to recall the PRINT statement. Edit the statement to print the number 2 by positioning the cursor under the 1 and typing 2 over it. Press Return again. Now press **Recall** several times to see all of the statements it remembers. Note that **Recall** goes backward through the queue.

Pressing Shift-f8 allows you to cycle forward through the queue until the last line entered, executed, or deleted is displayed. In the previous exercise you pressed unshifted-f8 several times, cycling backward through the queue. Now press Shift-f8 several times to cycle forward through the queue until the last line is displayed.

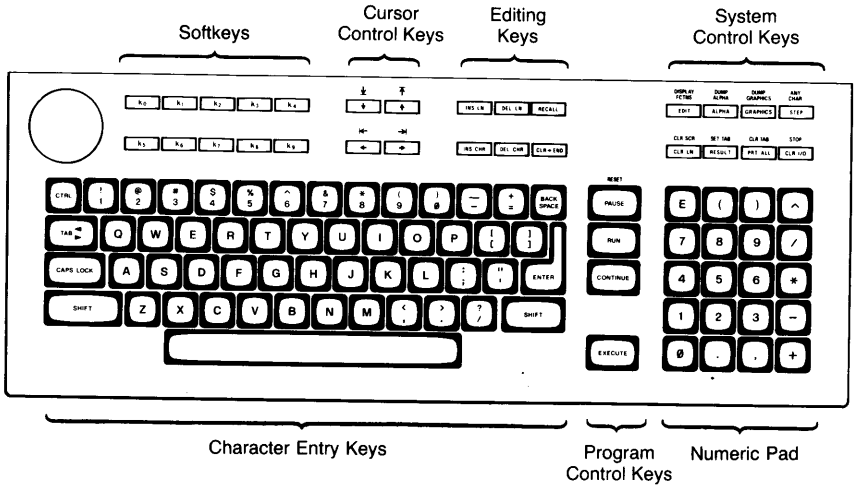
Now skip ahead to Chapter 5.

HP 98203B Keyboard Definitions

Note

If you have the HP 46020A keyboard, refer to the preceding section. If you have the HP 98203A keyboard, skip to the following section.

The keys on the HP 98203B keyboard are arranged into the following functional groups:



HP 98203B Keyboard

This section provides a handy reference guide to BASIC's key definitions for the HP 98203B keyboard. Keep in mind that other system programs may define the keys differently. Each key will be demonstrated where possible. One point to clarify: the **cursor** that we refer to in the following paragraphs is the underline that points to a location on the screen.

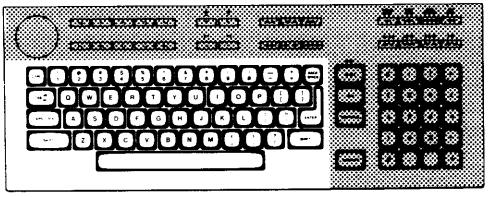
Note

Before you proceed, type:

SCRATCH **EXECUTE**

This clears the computer of any programs that might be left in memory from previous demonstrations.

Character Entry Keys



The character entry keys are arranged like a typewriter, but have some added features.

The CAPS LOCK key sets the unshifted keyboard to either upper case (which is the default after BASIC is booted) or lower case (normal typewriter operation). The computer displays which mode the computer is in when you press the CAPS LOCK key.

Type a few words, then press CAPS LOCK and continue typing. Notice the case change. Press CLR LN when finished.

You can enter standard upper-case and lower-case letters, using the SHIFT key to access the alternate case.

Type a few words, pressing SHIFT to change the case of the first letter of each word. Now press CAPS LOCK and continue typing. Notice that the alternate case accessed by SHIFT depends on the setting of CAPS LOCK. Press CLR LN when finished.

The ENTER key has several functions:

1. When a running program prompts you for data, you respond by typing the requested data and then pressing ENTER. This signals the program that you have provided the data and that it can resume execution. The EXECUTE key can also be used for this function.

2. When typing in lines of a program, the [ENTER] key is used to store each line of program code. The [EXECUTE] key can also be used for this function.
3. Like the [EXECUTE] key, the [ENTER] key can be used to execute commands and calculations.

Type `EDIT` and press [ENTER]. Notice the number `10` now displayed on the screen–this is the line number of the first line of a BASIC program. The computer is waiting for you to type in the line. Type:

```
!FIRST LINE
```

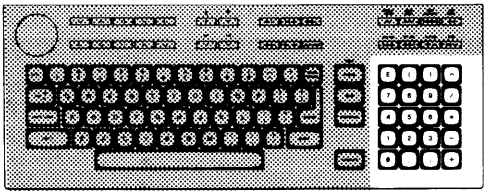
and press [ENTER]. Notice that the computer accepts the statement as a program line and displays `20` in preparation for the next one. Press [PAUSE] when finished.

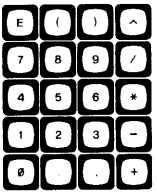
The [TAB] key moves the cursor forward to preset tabs. Pressing [SHIFT]-[TAB] moves the cursor back to preset tabs.

Before [TAB] can be used, a tab must be set. Press the space bar to move the cursor forward a few spaces and press [SET TAB] ([SHIFT]-[RESULT]). Move the cursor back several spaces using [←], then press [TAB]. Move the cursor forward several more spaces with the space bar, then press [SHIFT]-[TAB]. To clear the tab, move the cursor to the unwanted tab position and press [CLR TAB] ([SHIFT]-[PRT ALL]). Press [CLR LN] when finished.

The [CTRL] (control key works like [SHIFT] to access a set of standard control characters, such as line-feed and form-feed. These characters are useful to the programmer for controlling some devices and for communicating with other computers. You probably won't need them when running programs. The available control characters are listed in the Useful Tables section of *BASIC Language Reference.*

Numeric Pad



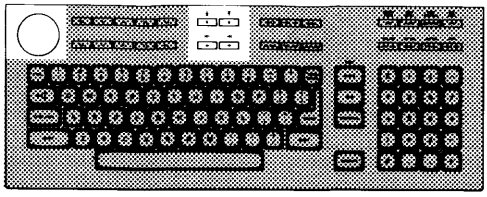


The numeric pad provides a convenient way to enter numbers and perform arithmetic operations. Simply type in the arithmetic expression you want to evaluate, then press EXECUTE. The result is displayed in the lower-left corner of the screen.

Type in the following problem using the numeric pad:

```
(26+14)/4
```

Now press EXECUTE to perform the calculation. The answer, 10, is displayed in the lower-left corner of the screen.



Cursor-Control Keys

The cursor-control keys move the display cursor. The [↑] and [↓] keys allow you to scroll lines in the output area up and down. Shifted, they allow you to "jump" to the top and bottom of the output area. The [→] and [←] keys allow you to move horizontally along a line. Shifted, they allow you to "jump" to the left and right limits of a line. The [BACK SPACE] key works just like the [←] key.

The cursor control wheel (also called the knob) allows you to rapidly scroll the print area (with [SHIFT] depressed) or move the cursor left and right (unshifted).

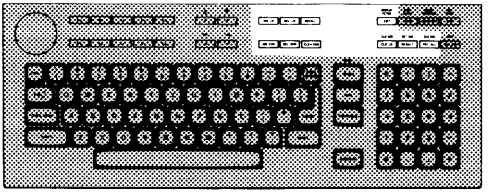
To test the horizontal movement of the cursor, type a few words and press the [←] and [→] keys. Notice that the cursor cannot be moved beyond the characters you have typed. Now rotate the wheel to move the cursor. Press [CLR LN] when finished.

To test vertical scrolling, type `EDIT` and press [EXECUTE]. Now type the following lines, pressing [ENTER] after each line (the first line may be there already, so just press [ENTER] to accept it):

```
10 !FIRST LINE
20 !SECOND LINE
30 !THIRD LINE
40 !FOURTH LINE
```

Press the [SHIFT] key and rotate the wheel to scroll the text up and down. Also try out the [↑] and [↓] keys. When you're done, press [PAUSE] to exit.

Editing Keys



The editing keys put easy character editing and line editing at your fingertips.

DISPLAY FCTNS
EDIT

The EDIT key is a typing convenience; pressing EDIT followed by EXECUTE puts the computer in program edit mode. Edit mode allows the programmer to enter and edit program lines.

Press EDIT, then EXECUTE to enter edit mode. The number 10 appears on the screen. This is a line number for a BASIC program; the computer is waiting for you to type in a line of code. If there is a program already in memory, the computer displays it on the screen. Press PAUSE to exit edit mode.

RECALL

The RECALL key recalls the last line that you entered, executed, or deleted. Several previous lines can be recalled this way. RECALL is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again.

Type:

```
PRINT "1" EXECUTE
```

to print the number 1 on the screen. Now press RECALL to recall the PRINT statement. Edit the statement to print the number 2 by positioning the cursor under the 1 and typing 2 over it. Press EXECUTE again. Now press RECALL several times to see all of the statements it remembers. Then press CLR SCR when finished.

SHIFT-RECALL moves forward through the recall stack.

(INS LN) inserts a new line above the cursor's current position (edit mode only).

Press (EDIT), then (EXECUTE). Type in this line (if it isn't already there):

```
10 !FIRST LINE
```

Now, with the cursor somewhere on line 10, press (INS LN). Notice that a new line number (1) is inserted before line 10. Press (PAUSE) when finished.

(DEL LN) deletes the line containing the cursor (edit mode only).

Press (EDIT), then (EXECUTE). Position the cursor to the line:

```
10 !FIRST LINE
```

and press (DEL LN). The line is removed. To restore it, press (RECALL) to retrieve it, then (ENTER) to enter it into the program. Press (PAUSE) to exit edit mode.

(INS CHR) sets insert mode, allowing you to insert characters to the left of the cursor. Press the key a second time to cancel insert mode.

Carefully type the following line exactly as shown:

```
THIS IS A TEST .
```

Position the cursor under the period and press (INS CHR). Now type:

```
OF INSERT MODE
```

and press (INS CHR) again. The line should now look like this:

```
THIS IS A TEST OF INSERT MODE.
```

The new characters were inserted to the left of the period. Press (CLR LN) when finished.

DEL CHR — **DEL CHR** deletes the character at the cursor's position.

Type a few words and experiment with **DEL CHR**, positioning the cursor at various places on the line. Notice that if you hold the key down, characters are deleted until you release it. Delete all of the characters you typed.

shift- SET TAB / RESULT — **SET TAB** sets a tab at the cursor's current position. Tabs remain in effect until cleared by either **CLR TAB** or the SCRATCH A statement. The SCRATCH commands are explained in *BASIC Programming Techniques*, Chapter 2. To demonstrate **SET TAB**, see **TAB**.

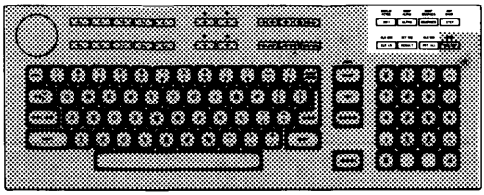
shift- CLR TAB / PRT ALL — **CLR TAB** clears a tab previously set at the cursor's position. To demonstrate **CLR TAB**, see **TAB**.

CLR LN — **CLR LN** clears the keyboard line and message/results line.

Type a few words and press **CLR LN** to clear them.

CLR→END — **CLR→END** clears from the current cursor position to the end of the line.

Type in a few words and use the cursor control wheel or ← to position the cursor in the middle of the line. Press **CLR→END** to clear to the end of the line. Press **CLR LN** to clear the rest of the line.



System Control Keys

These keys control various system functions related to the display, printer and editing operations. Most of these keys execute their functions immediately, as the key is pressed.

EDIT

EDIT types the EDIT command on the keyboard line. See the Editing Keys section for more information.

shift- DISPLAY FCTNS / EDIT

DISPLAY FCTNS sets the display- functions mode, allowing you to see special control characters (e.g., form-feed, carriage return) on the screen. Pressing this key a second time cancels the display-functions mode.

Type the following line:

```
PRINT "DISPLAY-FUNCTIONS MODE OFF"  EXECUTE
```

Notice the display at the top of the screen. Now press **RECALL** to recall the line, and edit it to read:

```
PRINT "DISPLAY-FUNCTIONS MODE ON"
```

Press the **DISPLAY FCTNS** key, and then press **EXECUTE**. Notice that the carriage return (CR) and line-feed (LF) control characters are now displayed. Press **DISPLAY FCTNS** again to exit display-functions mode. Press **CLR SCR** when finished.

ALPHA

GRAPHICS

ALPHA and **GRAPHICS** allow you to turn the alpha and graphics display modes on and off. These keys are demonstrated in Chapter 7. The GRAPH BIN file must be loaded for these keys to function.

DUMP ALPHA
shift- ALPHA

The **DUMP ALPHA** key prints a complete copy of the alpha display on the default printer. Refer to the "Dumping to a Printer" section of Chapter 6 for additional information.

DUMP GRAPHICS
shift- GRAPHICS

The **DUMP GRAPHICS** key prints a complete copy of the graphics display on the default printer. Refer to the "Dumping to a Printer"section of Chapter 6 for additional information. The GRAPH language extension file must be loaded for this key to function.

STEP

STEP allows the programmer to step through a program, one line at a time. Using the **STEP** key to debug programs is covered in *BASIC Programming Techniques*, Chapter 12.

ANY CHAR
shift- STEP

ANY CHAR is used to find any ASCII character. First press **ANY CHAR**. Then enter a three-digit number from 000 thru 255 representing the decimal equivalent of an ASCII character. The computer automatically displays the character on the screen. For a list of characters and their equivalent decimal values, see the US ASCII Character Codes table in the Useful Tables section of the *BASIC Language Reference.*

Press **ANY CHAR**, then type 189. 189 is the decimal equivalent of "§", which is now displayed in the keyboard line. Press **CLR LN** to erase it.

CLR SCR
shift- CLR LN

CLR SCR clears the entire alpha screen.

Type the following BASIC command:

```
PRINT "PUT THIS MESSAGE IN THE OUTPUT AREA."
```

Now press **EXECUTE** to execute it. Press **RECALL** to recall the command and press **EXECUTE** again. Repeat this step several times to fill the screen with messages. Now press **CLR SCR** to erase all lines at once.

RESULT returns the result of the last arithmetic expression that was executed.

Press CLR LN, then type:

```
23+45 EXECUTE
```

The result, 68, is displayed in the lower-left corner of the screen. To add 123 to this value, type:

```
RESULT +123 EXECUTE
```

The new result, 191, is now displayed. Press CLR LN when finished.

The PRT ALL key turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device. Press PRT ALL once to set printall to ON and again to set printall to OFF. The printall mode is displayed in the lower-left corner of the screen.

The screen's output area is the default printall device at powerup. Selecting an external printall device is explained in *BASIC Programming Techniques*, Chapter 8.

Press PRT ALL to turn on printall mode. Now type in the following command:

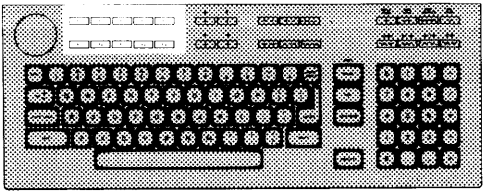
```
PRINT "THIS IS A KEYBOARD OPERATION" EXECUTE
```

Both the PRINT command and the message itself are displayed on the screen, which is the default printall device. Now type:

```
THIS WILL CAUSE AN ERROR EXECUTE
```

Because this is not an executable BASIC statement, an error message is displayed, both at the bottom of the screen and in the printall area at the top. This way, a log is produced of all commands typed and executed at the keyboard, along with any error messages. Press CLR SCR to clear the screen, and press PRT ALL to turn off printall mode.

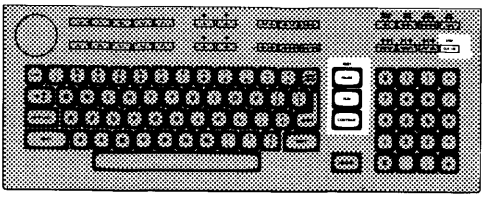
Softkeys



The ten keys labeled [k0] thru [k9] are defined under program control. The program may also display a label for each defined key. Pressing a defined key tells the computer to interrupt whatever it's doing and start running another part of the program.

We call these keys "softkeys" because the program or "software" defines and labels them. Another ten softkeys (without the displayed labels) can be defined at the same time and accessed with the [SHIFT] key. These shifted softkeys are often referred to as **k10** thru **k19**.

With language extension file KBD loaded, softkeys are defined as typing aids.



Program Control Keys

The keys shown below allow you to control execution of the program stored in the computer's memory. Most of these keys are demonstrated in the remaining chapters of this manual.

RUN

RUN starts a program running from the beginning.

PAUSE

PAUSE pauses program execution after the current line. It is also used to exit the Editor.

CONTINUE

CONTINUE resumes program execution from the point where it was paused. It is also used like **ENTER** or **EXECUTE** to respond to a program prompt.

STOP
shift- CLR I/O

STOP stops program execution after the current line. Unlike **PAUSE**, you cannot resume execution of a program stopped with **STOP** by pressing **CONTINUE**. To restart the program, use the **RUN** key.

RESET
shift- PAUSE

RESET stops program execution immediately without erasing the program from memory. The `BASIC Reset` message indicates the computer is ready for your command.

CLR I/O

CLR I/O pauses program execution when the computer is performing or trying to perform an I/O operation. Press **CLR I/O** instead of **PAUSE** when the computer is hung up on an I/O operation, since **PAUSE** works only after the computer finishes the current program line. Pressing **CLR I/O** cancels the I/O operation and pauses the program at the current line.

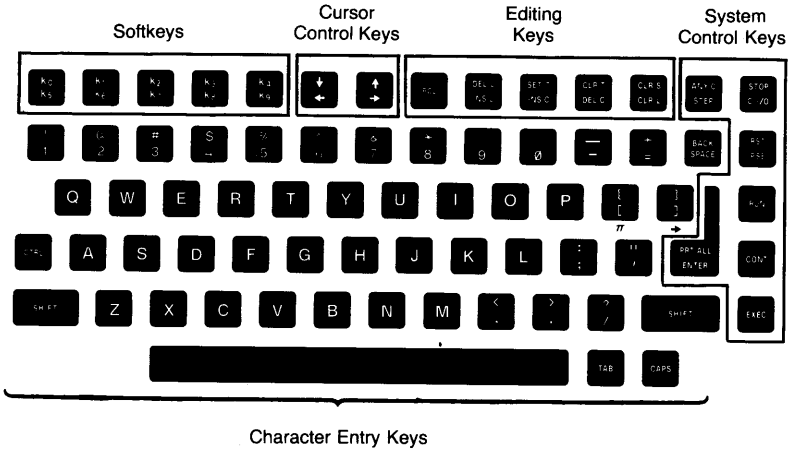
Now skip ahead to Chapter 5.

HP 98203A Keyboard Definitions

Note

If you have an HP 46020A or an HP 98203B keyboard, ignore this section and refer to one of the two preceding sections.

The keys on the HP 98203A keyboard are arranged into the following functional groups:



HP 98203A Keyboard

This section provides a handy reference guide to BASIC's key definitions for the HP 98203A keyboard. Keep in mind that other system programs may define the keys differently. Each key will be demonstrated where possible. One point to clarify: the **cursor** that we refer to in the following paragraphs is the blinking-underline that points to a location on the screen.

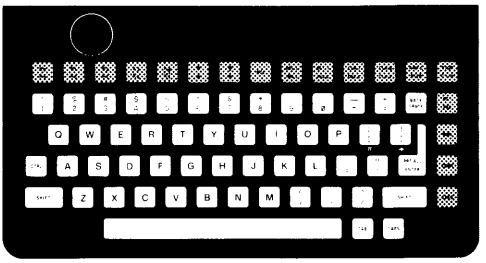
Note

Before you proceed, type:

SCRATCH (EXEC)

This clears the computer of any programs that might be left in memory from previous demonstrations.

Character Entry Keys



The character entry keys are arranged like a typewriter, but have some added features.

The CAPS key sets the unshifted keyboard to either upper case (which is the default after BASIC is booted) or lower case (normal typewriter operation). The computer displays which mode the computer is in when you press the CAPS key.

Type a few words, then press CAPS and continue typing. Notice the case change. Press CLR L when you're done.

You can enter standard upper-case and lower-case letters, using the SHIFT key to access the alternate case.

Type a few words, pressing SHIFT to change the case of the first letter of each word. Now press CAPS and continue typing. Notice that the alternate case that SHIFT accesses depends on the setting of CAPS. Press CLR L when you're done.

The ENTER key has several functions:

1. When a running program prompts you for data, you respond by typing the requested data and then pressing ENTER. This signals the program that you have provided the data and that it can resume execution. The EXEC key can also be used for this function.

2. When typing in lines of a program, the [ENTER] key is used to store each line of program code. The [EXEC] key can also be used for this function.
3. Like the [EXEC] key, the [ENTER] key can be used to execute commands and calculations.

Type EDIT and press [ENTER]. Notice the number 10 now displayed on the screen–this is the line number of the first line of a BASIC program. The computer is waiting for you to type in the line. Type:

```
!FIRST LINE
```

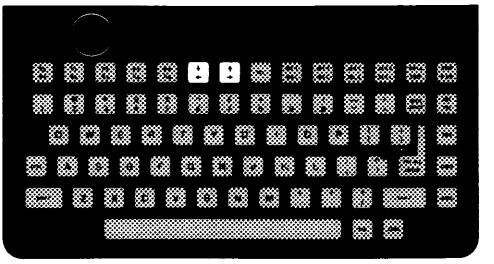
and press [ENTER]. Notice that the computer accepts the statement as a program line and displays 20 in preparation for the next one. Press [PSE] to exit.

The [TAB] key moves the cursor forward to preset tabs. Pressing [SHIFT]-[TAB] moves the cursor back to preset tabs.

Before [TAB] can be used, a tab must be set. Press the space bar to move the cursor forward a few spaces and press [SET T]. Move the cursor back several spaces using [←], then press [TAB]. Move the cursor forward several more spaces with the space bar, then press [SHIFT]-[TAB]. To clear the tab, move the cursor to the unwanted tab position and press [CLR T].

The [CTRL] (control) key works like SHIFT to access a set of standard control characters, such as line-feed and form-feed. These characters are useful to the programmer for controlling some devices and for communicating with other computers. You probably won't need them when running programs. The available control characters are listed in the Useful Tables section of the *BASIC Language Reference.*

Cursor-Control Keys



The cursor-control keys move the display cursor. The [↑] and [↓] keys allow you to scroll lines in the output area up and down. The [→] and [←] keys allow you to move horizontally along a line. The [BACK SPACE] key works just like the [←] key.

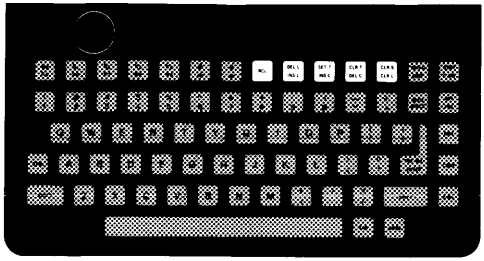
The cursor control wheel (also called the knob) allows you to rapidly scroll the output area up and down or move the cursor left and right, depending on the [SHIFT] key. With the [SHIFT] key depressed, the knob scrolls the output area up and down. Without the [SHIFT] key depressed, the knob moves the cursor left and right.

To test the horizontal movement of the cursor, type a few words and press the [←] and [→] keys. Notice that the cursor cannot be moved beyond the characters you have typed. Now rotate the wheel to move the cursor. Press [CLR L] when you're done.

To test the vertical scrolling, type `EDIT` and press [EXEC]. Now type the following lines, pressing [ENTER] after each line (the first line may be there already, so just press [ENTER] to accept it):

```
10 !FIRST LINE
20 !SECOND LINE
30 !THIRD LINE
40 !FOURTH LINE
```

Now, press [SHIFT] and rotate the knob to scroll the text up and down. Also try out the [↑] and [↓] keys. When you're done, press [PSE] to exit.



Editing Keys

The editing keys put easy character editing and line editing at your fingertips. Some of these keys only work when you are in edit mode, which is entered by typing:

EDIT (EXEC)

Edit mode is described in detail in *BASIC Programming Techniques*, Chapter 2. To exit edit mode, press (PSE).

The (RCL) key recalls the last line that you entered, executed or deleted. Several previous lines can be recalled this way. (RCL) is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again.

Type:

PRINT "1" (EXEC)

to print the number 1 on the screen. Now press (RCL) to recall the print statement. Edit the statement to print the number 2 by positioning the cursor under the 1 and typing (2) over it. Press (EXEC) again. Now press (RCL) several times to see all of the statements it remembers from the last entered to the earliest entered. Then press (SHIFT)-(RCL) several times to review the statements from the earliest to the last. Press (CLR S) to exit.

(INS L) inserts a new line above the cursor's current position (edit mode only).

Type EDIT (EXEC). Type in this line (if it isn't already there):

```
10 !FIRST LINE
```

Now, with the cursor somewhere on line 10, press (INS L). Notice that a new line number (1) is inserted before line 10. Press (PSE) to exit.

(DEL L) deletes the line containing the cursor (edit mode only).

Type EDIT (EXEC). Position the cursor to the line:

```
10 !FIRST LINE
```

and press (DEL L). The line is removed. To restore it, press (RCL) to retrieve it, then (ENTER) to enter it into the program. Press (PSE) to exit.

(INS C) sets insert mode, allowing you to insert characters to the left of the cursor. Press the key a second time to cancel insert mode.

Carefully type the following line exactly as shown:

```
THIS IS A TEST .
```

Position the cursor under the period and press (INS C). Now type:

```
OF INSERT MODE
```

and press (INS C) again. The line should now look like this:

```
THIS IS A TEST OF INSERT MODE.
```

The new characters were inserted to the left of the period. Press (CLR L) when you're done.

(SET T) sets a tab at the cursor's current position. Tabs are in effect for the keyboard line until cleared by either (CLR T) or the SCRATCH A statement. The SCRATCH commands are explained in *BASIC Programming Techniques*, Chapter 2. To demonstrate (SET T), see (TAB).

(CLR T) clears a tab previously set at the cursor's position. To demonstrate (CLR T), see (TAB).

(DEL C) deletes the character at the cursor's position. Type a few words and experiment with (DEL C), positioning the cursor at various places on the line. Notice that if you hold the key down, characters are deleted until you release it. Delete all of the characters you typed.

(CLR L) clears the keyboard line and message/results line. Type a few words and press (CLR L) to clear them.

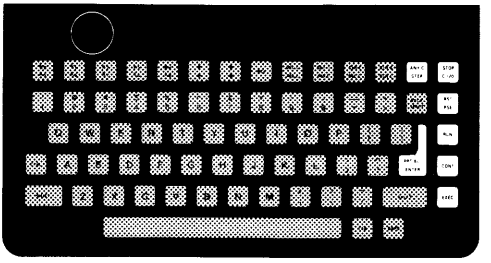
(CLR S) clears the entire alpha screen.

Type the following BASIC command:

```
PRINT "PUT THIS MESSAGE IN THE OUTPUT AREA."
```

Now press (EXEC) to execute it. Press (RCL) to recall the command and press (EXEC) again. Repeat this step several times to fill the screen with messages. Now press (CLR S) to erase all lines at once.

System Control Keys



The keys on the right-hand side of the keyboard control various system functions related to the display, printer and editing operations. Most of these keys execute their functions immediately, as the key is pressed.

(STEP) allows the programmer to step through a program, one line at a time. Using the (STEP) key to debug programs is covered in *BASIC Programming Techniques*, Chapter 12.

(ANY C) is used to find any ASCII character. First press (ANY C). Then enter a three-digit number from 000 thru 255 representing the decimal equivalent of an ASCII character. The computer automatically displays the character on the screen. For a list of characters and their equivalent decimal values, see the US ASCII Character Codes table in the Useful Tables section of the *BASIC Language Reference*.

Press (ANY C), then type 189. 189 is the decimal equivalent of "§", which is now displayed in the keyboard line. Press (CLR L) to erase it.

(RST) (reset) stops program execution immediately without erasing the program or data from memory. The `BASIC Reset` message indicates that the computer is ready for your command.

The PRT ALL key turns the printall mode on and off, allowing keyboard operations and displayed error messages to be copied to a printall device. Press PRT ALL once to set printall to ON and again to set printall to OFF. The printall mode is displayed in the lower-left corner of the screen.

The screen's output area is the default printall device at powerup. Selecting an external printall device is explained in *BASIC Programming Techniques*, Chapter 8.

Press PRT ALL to turn on printall mode. Now type in the following command:

```
PRINT "THIS IS A KEYBOARD OPERATION" EXEC
```

Both the PRINT command and the message itself are displayed on the screen, which is the default printall device. Now type:

```
THIS WILL CAUSE AN ERROR EXEC
```

Because this is not an executable BASIC statement, an error message is displayed, both at the bottom of the screen and in the printall area at the top. This way, a log is produced of all commands typed and executed at the keyboard, along with any error messages. Press CLR S to clear the screen, and press PRT ALL to turn off printall mode.

RUN starts a program running from the beginning.

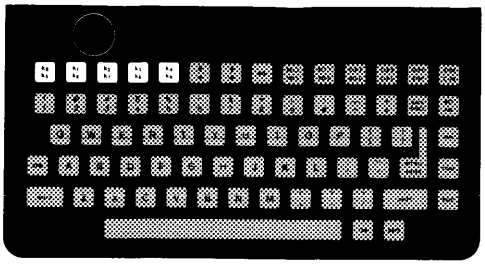
PSE pauses program execution after the current line. When in edit mode, PSE causes the computer to exit edit mode. Some BASIC keyboard commands cannot be executed while a program is running. In this situation, you can press PSE to suspend program execution, type and execute your keyboard command, then resume the program with the CONT key (described next). (Some keyboard commands make a program non-continuable.)

CONT resumes program execution from the point where it was paused. It may also be used like ENTER and EXEC to respond to a program prompt.

C I/O pauses program execution when the computer is doing an I/O operation. Press C I/O instead of PSE when the computer is hung up on an I/O operation, since PSE works only after the computer finishes the current statement. Pressing C I/O cancels the I/O operation and pauses the program at the current line.

STOP stops program execution after the current line. Unlike PSE, you cannot resume execution of a program stopped with STOP by pressing CONT. To restart the program from the beginning, use the RUN key.

Softkeys



The ten keys labeled [k0] thru [k4] (using the [SHIFT] key) and [k5] thru [k9] are defined under program control. The program may also display a label for each defined key. Pressing a defined key tells the computer to interrupt whatever it's doing and start running the designated part of the program.

We call these keys "softkeys" because the program or "software" defines and labels them.

With the KBD language extension file loaded, softkeys can be used as typing aids.

Working With BASIC

Chapter 5

You now have BASIC loaded and have become familiar with your keyboard. In this chapter you will continue to learn how BASIC works by writing, running, editing, storing, and loading a simple program. You will also learn the BASIC arithmetic operations. At the end of this chapter are some hints on dealing with errors and interpreting error messages.

Programming With BASIC

Note

When entering program lines and doing other operations, you must press certain keys. The keys differ depending on your keyboard. Keys for all three keyboards are shown in this order: HP 46020A, HP 98203B, and HP 98203A (e.g., Return, EXECUTE, or EXEC).

Press the key that is on **your** keyboard.

Writing A Program

Type:

SCRATCH Return, EXECUTE, or EXEC

This erases any program that may have been in your computer's memory. You now have a "blank slate".

Type:

EDIT Return, EXECUTE, or EXEC

The number 10 should appear in the middle left of your display, indicating the computer is ready to accept the first program line.

Type:

For A=0 to 99 Return, ENTER, or ENTER

The computer accepts line 10 and presents line 20 so you can type in the second statement.

Continue the process until you have the following program on your screen:

```
10    FOR A=0 TO 99
20    PRINT A
30    NEXT A
40    END
50
```

You have now written a program that will print all numbers from 0 to 99, then stop.

Running Your Program

Note

If you have an HP 46020A keyboard, you must be in System mode with the System softkeys displayed in order to use the **RUN** softkey (f3). Otherwise, you can type RUN Return.

Press: **RUN** (f3), RUN, or RUN

Numbers up to 99 are printed at the left of the display.

Now let's pause the program before it stops.

Press: **RUN**, RUN, or RUN
and then immediately press unshifted-Stop, PAUSE, or PSE

The program pauses after executing the current line, displays the next line to be executed, and displays the paused indicator (-) at the lower-right of the screen.

You can now step through the program one line at a time by pressing **Step** (f1) STEP, or STEP. You can also continue running the program by pressing **Continue** (f2), CONTINUE, or CONT. If you press **RUN**, RUN, or RUN while paused, the program begins running from the beginning.

Editing Your Program

Now suppose you want to change your program so it counts from 0 to 199.

Type:

EDIT (Return), (EXECUTE), or (EXEC)

Your program should appear on the screen.

Use your (▼), (↓), or (↓) key, your cursor control wheel, or the mouse to place the cursor at line 10. (You could accomplish the same result by typing EDIT 10 (Return), (EXECUTE), or (EXEC).)

Edit line 10 by changing 99 to 199. Remember to press
(Return), (ENTER), or (ENTER)
to successfully change the program statement.

Press: **RUN**, (RUN), or (RUN) to see if your new program works.

Storing And Loading Your Program

Verify that your program is now in your computer's memory.

Type:

LIST (Return), (EXECUTE), or (EXEC)

If someone turns your computer off now, your program will be lost. The computer's memory goes blank when it is turned off. Therefore, you want to store your program where it will be safe and can be used again and again.

Store your program on the same drive from which you booted your system. Insert a blank, initialized disc into the drive.

Type:

STORE "COUNT" (Return), (EXECUTE), or (EXEC)

Your program, named COUNT, is now stored on your disc.

Now, erase your program from computer memory.

Type:

SCRATCH (Return), (EXECUTE), or (EXEC)

Verify that your program is no longer in computer memory.

Type:

LIST (Return), (EXECUTE), or (EXEC)

The display should just show available memory. The program should be gone.

Let's **load** the program from the disc.

Type:

LOAD "COUNT" (Return), (EXECUTE), or (EXEC)

Press: **RUN**, (RUN), or (RUN)

The program should have been loaded into your computer and should run normally.

Arithmetic Operations

The BASIC language system lets you perform arithmetic calculations like a typical hand-held calculator. Just type in an arithmetic expression, press [Return], [EXECUTE], or [EXEC], and your computer will deliver the answer.

Some of the arithmetic operators used by your computer may differ from those you're familiar with. Look over the following table of operators, then we'll try out some examples:

Operator	Symbol	Key(s)
Exponentiation	^	[Shift]-[6]
Division	/	[/]
Multiplication	*	[Shift]-[8]
Subtraction	-	[-]
Addition	+	[Shift]-[=]

Arithmetic Examples

Before you perform arithmetic operations, clear the display, if necessary, by pressing [Clear display], [CLR SCR], or [CLR S]. Also, never insert spaces between the digits of a number.

Try It Yourself

Problem 1: If a computer's screen can display 25 lines, and each line can be up to 80 characters long, how many characters can be displayed on the computer's screen?

Type:

```
80*25
```
[Return], [EXECUTE], or [EXEC]

Your computer returns the answer: `2000` characters.

Problem 2: If you spend 3% of your time today reading this manual, how much of your eight-hour workday is left for work?

Type:

8-8*.03 (Return), (EXECUTE), or (EXEC)

Your computer returns the answer: 7.76 hours for work.

Problem 3: If the floor in your office is square, with each side measuring 6.2 metres, how many square metres of carpeting are needed to cover it? You can either multiply 6.2 * 6.2, or find the square of 6.2 by raising it to the second power (6.2^2).

Type:

6.2^2 (Return), (EXECUTE), or (EXEC)

Your computer returns the answer: 38.44 square metres.

You can always use **RECALL,** (RECALL), or (RCL) to recall an expression for additional computations. Press the recall key to recall the last problem and add 2 to the expression by typing:

+2 (Return), (EXECUTE), or (EXEC)

Your computer returns the answer: 40.44.

Your computer saves time by automatically assuming INTEGER notation whenever you enter small numbers without a decimal point, or an "E" to indicate an exponent. But INTEGER notation has its limits: from −32 768 to +32 767. So if you execute an expression with results beyond this integer range, say:

126*261 (Return), (EXECUTE), or (EXEC)

The computer reports:

```
ERROR 20  INTEGER overflow
```

To avoid this error, be sure to enter a decimal point in at least one of the numbers. This causes the computer to use REAL notation.

Try It Yourself

Type:

`126*261` (Return), (EXECUTE), or (EXEC)

Your computer reports the answer, `32886`.

Now try multiplying some really large numbers.

Type:

`6000000*90000` (Return), (EXECUTE), or (EXEC)

The computer reports the answer, `5.4E+11`, in scientific notation. When the absolute value of the result cannot be represented in six digits (result >= 1E6), the computer displays it in scientific notation. The number after the "E" is the exponent. It indicates how many places the decimal point must be moved to the right of the mantissa. In this case, the number is 540 000 000 000.

Arithmetic Hierarchy

When an arithmetic operation having more than one operator is executed, the computer evaluates the expression according to a predefined hierarchy:

^	(exponentiation)	performed first
*/	(multiplication and division)	
+-	(addition and subtraction)	performed last

As an expression is scanned, the highest-priority operations are performed first. The results of these operations are used as inputs to the lower-priority operators. Equal-priority operations are performed from left to right. When you want to be certain that one operation in the expression is evaluated before another, place that operation in parentheses.

Try It Yourself

Type in Problem 2 from the previous demonstration again:

8-8*.03 (Return), (EXECUTE), or (EXEC)

The computer responds with the answer: 7.76. The multiplication (8*.03) was evaluated first. By enclosing the subtraction in parentheses, you can change the order of evaluation and thus the result.

Type:

(8-8)*.03 (Return), (EXECUTE), or (EXEC)

The computer now returns the answer: 0.

The following example shows the order of evaluation for an expression with various operators:

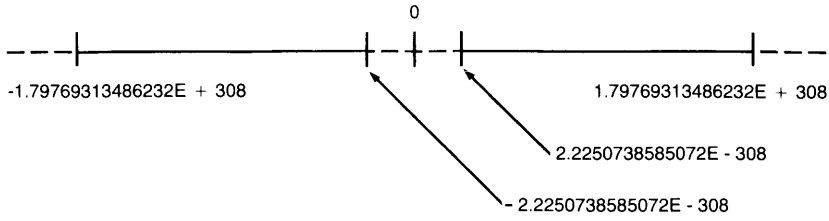
	2+3*6/(7-4)^2
1. evaluate parentheses:	2+3*6/**(7-4)**^2
2. exponentiation:	2+3*6/**3^2**
3. multiplication:	2+**3*6**/9
4. division:	2+**18/9**
5. addition:	**2+2**
result:	**4**

If you nest parentheses, the inner-most operation is evaluated first.

Whenever you're in doubt about the order of execution, use parentheses to force the order you want. However, using parentheses for "implied" multiplication is not allowed. So the expression, $4(5-2)$ must be entered as 4*(5-2).

Computing Range

The range of values that your computer can handle is from about 10^{-308} thru 10^{308}. More exact limits are:



Computing Range

Significant Digits and Rounding

Your computer can input, store, and calculate numbers having approximately 15 significant digits. Significant digits are those which determine a number's actual value: leading zeros are not counted. When you enter a number with more than 16 significant digits, the excess digits are truncated (dropped) and the 16th digit is rounded before the number is accepted. The computer automatically rounds each final result to 12 digits before it's output. This can be decreased or increased up to 15 digits by specifying an image in your output operations. For details on printing with images, refer to your *BASIC Programming Techniques* manual, Chapter 8.

The computer rounds a number by first looking at the 13th digit. If it's 5 or greater, the 12th digit is incremented by one. If the 13th digit is less than 5, it's simply truncated from the number. This is called "rounding up."

BASIC functions are available for rounding numbers before processing or storage. For details on these functions, consult your *BASIC Programming Techniques*, Chapter 4.

Dealing with Errors

As you saw in some of the earlier demonstrations, the BASIC system responds to an incorrect operation with an error message. The message contains an error number and a short description of the error. (If the ERR BIN file is not loaded, only the error number is displayed.) If the error occurred in a running program, the number of the offending line is also given. These error messages may seem short and cryptic, but it's important for you to understand what the computer is trying to say. If you're unfamiliar with error messages, take a few minutes to read this section.

Keyboard Error Messages

The computer rejects incorrect keyboard entries by beeping and displaying an error message.

Try It Yourself

Suppose you want to calculate 10/.03, but instead you typed:

```
10/,0
```
(Return), (EXECUTE), or (EXEC)

The computer knows division by zero is illegal and politely reminds you of this fact with the message:

```
ERROR 31  Division by 0 or X MOD 0
```

As another example, what would happen if you meant to type the BASIC command CAT, but typed the following line by mistake:

```
CAR
```
(Return), (EXECUTE), or (EXEC)

The computer knows this is not a legal BASIC command, and displays the following error message:

```
ERROR 310  Ident not found in context
```

Whenever you make this type of error, the computer positions the cursor under the word it doesn't recognize.

To understand the meaning of error messages, you need to look at the error as the computer sees it–not as you *expected* to see it. This is not always easy to do. Fortunately, unless you're writing programs, you don't need to understand every error message.

Most keyboard error messages simply point out typing mistakes, such as CAR instead of CAT, or a missing quote mark. When an error occurs, carefully check your spelling and punctuation, make any corrections, and execute the line again. Remember, computers are very narrow-minded owing to their limited intellects, so you must humor them with extremely precise instructions. Don't let error messages bother you; they're a normal part of the computing experience.

Program Errors

All the rules for keyboard commands–and many more–apply to running a program. When an error occurs during execution of a program, the computer pauses and displays an error message like this:

```
ERROR 31 IN 280  Division by 0 or X MOD 0
```

This error code (31) refers to an attempted division by 0 in program line 280. There are many different error messages, with expanded definitions, listed in the back of both the *BASIC Programming Techniques* and the *BASIC Language Reference* manuals. Some represent programming mistakes; others represent mistakes you have made while entering data. The latter case probably results from a simple typing mistake on the input line, and the computer is waiting for you to re-enter the correct data. No matter what the error message, you should get the programmer involved.

There is one golden rule to remember whenever you get an error message while running a program: **Any time you get a computer error message from a running program, it is the programmer's fault–not yours!** This is always true unless, of course, the programmer is you.

The programming language on your computer allows a programmer to anticipate or intercept any error resulting from an operator's mistake. Admittedly, such a comprehensive level of error-catching is difficult and time-consuming to achieve. The programmer may not have taken the time to deal with your particular error, but the interception of all errors is an admirable goal. The program should display an explanation of the error, suggest corrective action, and give you a second chance. If it doesn't, write down the error message and contact the person who is supporting your program. Tell them what you did, show them the error message, and act like you expect them to do something about it.

Talking To Peripherals With BASIC

Chapter 6

This chapter introduces some fundamental disc drive and printer operations that you should know about. We'll also demonstrate how to generate a printed program listing and how to "dump" graphics and alpha displays from the screen to the printer.

Simple Disc Drive Operations

Note

If you want additional information on the msus and other mass storage topics, refer to your *BASIC Programming Techniques*, Chapter 7.

Initializing Discs

Refer to Chapter 1 for disc initialization procedures.

Copying Discs

Refer to Chapter 1 for disc copying procedures.

Listing a Disc's Directory

You can find out a lot of information about the files you have on a disc by listing the disc's directory with the CAT command.

Try It Yourself

Insert the *Drivers* (or *Language Extensions and Driver*) *Disc* supplied with your BASIC language system into one of your disc drives. Execute the CAT command with the msus of the drive containing the disc. For example, if the disc is inserted into the left drive (unit 0) of an HP 9121 connected to the internal HP-IB (select code 7) at primary address 0, the CAT command would look like this:

```
CAT ":,700,0"
```

The resulting display should look something like this:

```
:INTERNAL
VOLUME LABEL: DRIVER
FILE NAME PRO TYPE   REC/FILE BYTE/REC   ADDRESS

DISC          BIN          31      256        16
CS80          BIN          36      256        47
BUBBLE        BIN          15      256        83
EPROM         BIN          13      256        98
HP9885        BIN          18      256       111
HPIB          BIN          46      256       129
FHPIB         BIN           9      256       175
SERIAL        BIN          17      256       184
GPIO          BIN          21      256       201
BCD           BIN           9      256       222
DCOMM         BIN          25      256       231
CRTA          BIN          17      256       256
CRTB          BIN          24      256       273
```

Example of a Disc Directory

All of the files on your *Drivers* (or *Language Extensions and Drivers*) *Disc* are file type BIN (binary). BIN files hold system programs coded in a binary (machine language) format. BIN files are loaded into memory with the LOAD BIN command.

Other file types are stored on discs:

- PROG files hold programs that are stored on disc with the STORE command and loaded into memory with the LOAD command.
- ASCII files contain either programs or other information coded in a universal data format so that they can be read by other computers. ASCII files are stored on disc with the SAVE command and are brought into memory with the GET command.

- SYSTM files contain system programs that can be booted when the power is turned on.
- BDAT files contain data used by a BASIC program.

If the file type is a number, rather than a word, it indicates that the file type is not compatible with the BASIC language system. These files can be cataloged, but the contents of the file cannot be accessed by the computer.

The directory tells you more than just each file's name and type: it also indicates each file's size and location on the disc. File sizes are of great use to a programmer who must organize and support each file.

SRM Disc Files

If your files are located on a Shared Resource Management disc, the CAT display will look slightly different as shown below:

```
:REMOTE 21, 0
LABEL:
FORMAT: SDF
AVAILABLE SPACE:    274280
                        SYS  FILE   NUMBER   RECORD     MODIFIED      PUB OPEN
FILE NAME             LEV TYPE  TYPE  RECORDS   LENGTH DATE        TIME ACC STAT
-------------------- --- ---- ----- -------- -------- ------------- --- ----
lost_found                    DIR          0       24 19-Jan-84  17:48  RW
WORKSTATIONS                  DIR         57       24 16-Feb-84  12:28  R
SRM_NEWS                      DIR          1       24 11-Feb-84   0:16  RW
SYSTEMS                       DIR        102       24 30-Mar-84  13:29  R
STATUS                        DIR          2       24 29-Mar-84   8:19  RW
USERS                         DIR         26       24  1-Mar-84  13:49  R
BOOKS                         DIR          2       24 25-Jan-84  18:16  RW
TOOLS                         DIR         22       24 28-Mar-84  14:46  RW
GAMES                         DIR         11       24  7-Mar-84  15:46  RW
INFO                          DIR          7       24 22-Mar-84  10:15  RW
```

Example of an SRM Directory

Note that the SRM CAT display shows Public Access instead of Protection status and has an additional column for open status. These features are described thoroughly in the SRM chapter of the *BASIC Programming Techniques* manual.

Copying Files

You created backup discs by copying the entire contents of one disc to another disc. You can also copy a specific file on one disc to another disc. The COPY statement is used to copy files. For example, suppose you wanted to copy a file named COUNT from the disc in the right-hand drive of a Model 236 to a file named SUMS on the disc in the left-hand drive. You would type:

```
COPY "COUNT:INTERNAL,4,0" TO "SUMS:INTERNAL,4,1"
```

The file named COUNT is written from one disc to the other and assigned the new name SUMS. The destination file **can** be assigned the same name as the source file.

Notice that when copying files, the COPY statement requires a filename before the colon (:) in both the source and destination sides of the statement. When copying entire discs, the filenames are deleted.

The MASS STORAGE IS Statement

The MASS STORAGE IS statement (or MSI, for short) allows you to designate a disc drive as the system default mass storage device. Once this is done, all mass storage operations automatically refer to the default drive, and you don't have to type the msus each time.

Note

If you booted BASIC from a disc drive, the drive it was booted from automatically becomes the default mass storage device. The MASS STORAGE IS statement is only needed when you want to *change* the current default mass storage device.

To designate a drive as the system default, type:

```
MSI "msus"
```

Substitute your drive's msus in the line above, and press Return, EXECUTE, or EXEC.

For example, to designate unit 1 (the right-hand drive) of an HP 9121 disc drive connected to the internal HP-IB at primary address 0 as the system default, type:

```
MSI ":HP8290X,700,1" or ":,700,1" Return, EXECUTE, or EXEC
```

Now you can just type and execute CAT to list this drive's directory, for example. No further reference to the msus is required for most BASIC statements (INITIALIZE is an exception–you must use the msus). If, however, you want to reference a disc drive other than the system default, you must use its msus.

Try It Yourself

Execute an MSI statement as shown previously for one of your disc drives. Now insert any BASIC disc into this drive and type:

CAT (Return), (EXECUTE), or (EXEC)

The default drive is automatically accessed and the directory is listed.

Simple Printer Operations

The PRINTER IS Statement

The PRINTER IS statement is similar to the MASS STORAGE IS statement. It designates a printer as the system default printer, and all printer operations are directed to it.

The PRINTER IS statement is very simple to use. Just type `PRINTER IS` followed by the device selector of your printer (or `PRT` if you have just one printer). For example, to assign a printer on the internal HP-IB (select code 7) at primary address 1 as the default system printer, type:

`PRINTER IS 701` Return, EXECUTE, or EXEC

or

`PRINTER IS PRT` Return, EXECUTE, or EXEC

Note

The computer's screen is also considered a "printer". When you type `PRINTER IS 1` or `PRINTER IS CRT` Return, EXECUTE, or EXEC you set the system default printer to the screen, and all output is directed to it. The screen is automatically assigned as the system printer when BASIC is booted.

The PRINT Statement

The PRINT statement is the primary means of sending characters to the system default printer. The PRINT statement syntax is simply `PRINT` followed by a list of things to be printed, separated by commas or semicolons. The options available with the PRINT statement are discussed in detail in *BASIC Programming Techniques*, Chapter 8. In this book, we limit our discussion to sending simple messages to the printer.

Try It Yourself

Designate your printer as the system default by executing the appropriate PRINTER IS statement. Now, to send a message to the printer, just enclose the message in double quotes and type it following PRINT. For example, type:

```
PRINT "THIS IS A TEST OF THE SYSTEM DEFAULT PRINTER"
```

(Return), (EXECUTE), or (EXEC)

The message is printed on the printer. (If nothing was printed, make sure you have declared the printer as the default with the PRINTER IS statement. Also check that the printer is turned on and "on line".)

Type any other messages you want to print. Then restore the computer's screen as the default printer by typing:

```
PRINTER IS CRT
```
(Return), (EXECUTE), or (EXEC)

The Printall Printer

As discussed in the last chapter, a record of all keyboard commands and error messages can be logged to a printall printer. When you press **Print All,** (PRT ALL), or (PRT ALL), you activate printall mode.

When the computer is turned on, the printall printer is automatically set to the computer's screen. To make an external printer the printall printer, use the PRINTALL IS command.

The PRINTALL IS command works just like the PRINTER IS statement. Simply type `PRINTALL IS` followed by the device selector of your printer and press (Return), (EXECUTE), or (EXEC). For example, to set a printer on the internal HP-IB (select code 7) at primary address 1 to be the printall printer, just type:

```
PRINTALL IS 701
```
(Return), (EXECUTE), or (EXEC)

Now when printall mode is activated with the **Print All,** (PRT ALL), or (PRT ALL) key, all commands are logged on the printer. Try doing a CAT on your disc drive, and a PRINT on the default printer. Notice that each command you execute is printed.

Getting a Program Listing

To get a listing of a program, you must first LOAD it into your computer's memory. Then you can use the LIST command to print it.

If you simply type:

LIST (Return), (EXECUTE), or (EXEC)

the listing will be printed on the default system printer. If you want the listing to go to a printer other than the default, type LIST followed by # and the device selector of the printer. For example, to send the listing to a printer on the internal HP-IB (select code 7) at primary address 1, you would type:

LIST #701 (Return), (EXECUTE), or (EXEC)

Try It Yourself

To get a listing of the Breakout game, follow these steps:

1. Assign one of your disc drives as the default mass storage device with a MASS STORAGE IS statement. Insert the *Manual Examples Disc* into that drive and close the door.
2. Type:

   LOAD "BREAKOUT" (Return), (EXECUTE), or (EXEC)

   to load the program into your computer. Wait for the run light to go out.
3. Set your printer to default system printer with the PRINTER IS statement.
4. Type:

   LIST (Return), (EXECUTE) or (EXEC)

   to list the program on your printer.

Dumping to a Printer

Note

You cannot dump graphics to an HP 82905 printer with the techniques described in this section. However, there is an HP 82905B Dump Graphics Subprogram (82905BDUMP) on your *Utilities Disc.* See the *BASIC Utilities Library* manual for details.

You can "dump" alpha and graphics displays from your computer's screen to a printer by executing BASIC commands. To dump the alpha display, type:

```
DUMP ALPHA
```
Return, EXECUTE, or EXEC

Note

You must have the GRAPH BIN file loaded in order to execute `DUMP GRAPHICS` and `DUMP DEVICE IS`, and to run the demonstration.

To dump the graphics display, type:

```
DUMP GRAPHICS
```
Return, EXECUTE, or EXEC

Note

If you have a bit-mapped screen, this statement dumps the combined alpha and graphics display.

Before you can dump alpha and graphics to your printer, you may need to designate your printer as the dump device with the DUMP DEVICE IS statement. If your printer is connected to the internal HP-IB at primary address 1, it is already set as the system default dump device when your computer is turned on.

If your printer's device selector is not 701, you must execute the DUMP DEVICE IS statement. Type `DUMP DEVICE IS`, followed by the device selector of your printer, then press Return, EXECUTE, or EXEC. For example, if your printer is connected to the internal HP-IB (select code 7) at primary address 2, you would type:

```
DUMP DEVICE IS 702
```
Return, EXECUTE, or EXEC

Try It Yourself

This demonstration assumes you have a graphics printer.

1. Set one of your disc drives to the system default with the MASS STORAGE IS statement. If not already done, boot BASIC using the instructions is Chapter 1. Insert the *Manual Examples Disc* into the system default drive.
2. If necessary, designate your printer as the dump device with the DUMP DEVICE IS statement.
3. Type:

   `LOAD "SRM",1` (Return), (EXECUTE) or (EXEC)

   to load a program that demonstrates the HP Shared Resource Management network.
4. The initial display of the SRM program is an alpha display.

   Type:

   `DUMP ALPHA` (Return), (EXECUTE), or (EXEC)

   to print a copy of this display on your printer. Press the **CONTINUE** softkey ((f5) or (k0)) to move on to the next display.
5. The next display is a graphics display. Type:

   `DUMP GRAPHICS` (Return), (EXECUTE), or (EXEC)

   to print a copy of the display on your printer.

Special BASIC Features

Chapter 7

This chapter explores some of the more impressive features of BASIC on your computer. We'll demonstrate several ways to use graphics, and show you how to dress up your alphanumeric displays with display-enhancement characters. You'll also learn how to make your computer do two things at once, and how you can interact with a running program. Even if you have no intention of programming in BASIC, you'll still enjoy trying the demonstrations and seeing some of the "tricks" your computer can do.

Note

As in previous chapters, key presses for all keyboards are provided. The order in which the keys are listed is: HP 46020A, HP 98203B, HP 98203A. Press the key listed for your keyboard.

The following demonstrations assume you have booted BASIC according to the instructions given in Chapter 1, and designated one of your disc drives as the default mass storage device with the MASS STORAGE IS statement (refer to Chapter 6). The display should also be the default system printer; if you're not sure that it is, type:

```
PRINTER IS CRT
```
(Return), (EXECUTE), or (EXEC)

One other point to watch out for: many of these demonstrations ask you to transfer programs from disc into memory using the LOAD statement. If you execute a LOAD statement and get an "ERROR 120: Not allowed while prog running", press unshifted-(Stop), (PAUSE), or (PSE) or (Shift)-(Stop), (STOP), (STOP) and try it again. (Use the **Recall**, (RECALL), or (RCL) key to recall the statement, rather than typing it over.)

Display-Enhancement Characters

Note

Bit-mapped screens do not have blinking mode.

Display-enhancement characters are special, non-printable characters that affect how regular printable characters are displayed. Because these characters are non-printable, they are entered into the computer via their character codes.

A **character code** is simply a numeric code that represents a character to the computer. Since computers can store only numbers in their memories, character codes are used in lieu of actual characters.

The character code and display effect for each display-enhancement character is given in the following table.

Character Code	Action Resulting from Displaying the Character
128	All enhancements off (except colors on a color CRT)
129	Inverse mode on.
130	Blinking mode on.
131	Inverse and Blinking modes on.
132	Underline mode on.
133	Underline and Inverse modes on.
134	Underline and Blinking modes on.
135	Underline, Inverse, and Blinking modes on.
136-143	Color enhancements on a color CRT.[1]

Do You Have Display Enhancements?

Your computer may or may not have display enhancements. To check if your computer has the alpha enhancements, type:

```
SYSTEM$("CRT ID")
```

(Return), (EXECUTE), or (EXEC)

Your computer will display a response similar to the following in the message/results line:

```
6:128HCGB
```

1 Additional color enhancements are available on bit-mapped color screens via CRT control registers.

The "H" in the response indicates that the computer has "Highlights" (or display enhancements). If your computer's response has an "H", your computer has display enhancements; if your response does not have an "H", you do not have enhancements.

Incidentally, the number following the colon indicates the width of your display in characters. The "C" indicates Color capability; the "G" indicates Graphics; and the "B" indicates Bit-mapped display. If any letter is missing, your computer does not have that particular capability.

Using Display Enhancements

Display-enhancement characters are entered into your computer with the CHR$ function. The CHR$ function takes a character code and converts it into the corresponding character.

Try it first with a printable character. Type:

```
PRINT CHR$(65)
```

(Return), (EXECUTE), or (EXEC)

As you can see, 65 is the character code for the letter "A". Unlike printable characters like "A", when you enter a display-enhancement character with the CHR$ function, you will not see a character displayed on the screen. What you *will* see is a change in all subsequent *printable* characters that are displayed.

Try It Yourself

This demonstration shows the effect of each display-enhancement character. First, display a normal line of text. Type:

```
PRINT "NORMAL"
```
(Return), (EXECUTE), or (EXEC)

Now, use the inverse mode (character code 129). Type:

```
PRINT CHR$(129);"INVERSE MODE"
```
(Return), (EXECUTE), or (EXEC)

A display enhancement character affects all subsequent printable characters until it is replaced by another display-enhancement character. Thus, inverse mode was "turned on" in the first part of the PRINT statement, and caused the following characters to be displayed in inverse mode. Now, replace inverse mode with blinking mode (character code 130). Press the **Recall**, (RECALL), or (RCL) key and edit the line to read:

```
PRINT CHR$(130);"BLINKING MODE"
```
(Return), (EXECUTE), or (EXEC)

Next, change to inverse and blinking mode (character code 131):

```
PRINT CHR$(131);"INVERSE AND BLINKING MODE"
```
(Return), (EXECUTE), or (EXEC)

Next, change to underline mode (character code 132):

```
PRINT CHR$(132);"UNDERLINE MODE"
```
(Return), (EXECUTE), or (EXEC)

Next, change to underline and inverse mode (character code 133):

```
PRINT CHR$(133);"UNDERLINE AND INVERSE MODE"
```
(Return), (EXECUTE), (EXEC)

The underline is difficult to see, but it's there. Now, change to underline and blinking mode (character code 134):

```
PRINT CHR$(134);"UNDERLINE AND BLINKING MODE"
```
(Return), (EXECUTE), or (EXEC)

Now, put them *all* together in underline, inverse and blinking mode (character code 135):

```
PRINT CHR$(135);"UNDERLINE, INVERSE & BLINKING MODE"
```

Return, EXECUTE, or EXEC

Finally, turn off the display enhancements with the enhancements-off character (character code 128):

```
PRINT CHR$(128);"BACK TO NORMAL"
```

Return, EXECUTE, or EXEC

Be sure you remember to turn off the enhancements this way; they will stay on until you turn them off. Press Clear display, CLR SCR, or CLR S to clear the screen.

For more information on using the display enhancements, refer to *BASIC Interfacing Techniques*, Chapter 8.

Graphics

In this section, you will run a few graphics demonstrations from the *Manual Examples Disc*. For details on how to do your own graphics programming, consult *BASIC Programming Techniques* and *BASIC Graphics Techniques*.

Demonstrations

Insert the *Manual Examples Disc* into your default disc drive and close the door.

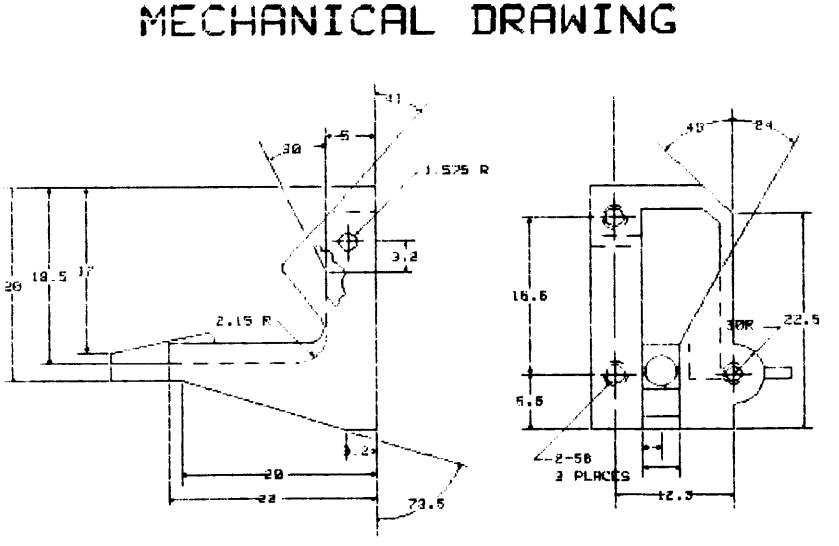
Now you're ready to load and run some demonstration programs. To get the first one, type:

```
LOAD "MECH",1
```

Return, EXECUTE, or EXEC

Placing the 1 after the program file name tells the computer to automatically begin executing the program after it has loaded it.

The following graphics display now appears on the screen. Press the softkey labeled **EXIT** (f8 or k8) to clear the screen.



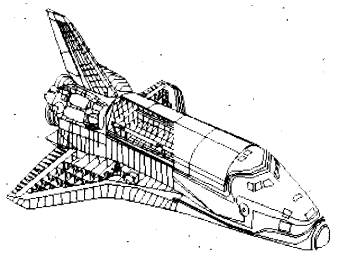
Mechanical Drawing Display

Now type the following line:

```
LOAD "SHUTTLE",1
```

Return, EXECUTE, or EXEC

Press Return, ENTER, or ENTER and **Continue**, CONTINUE, or CONT as directed by the program to get a picture of the space shuttle.



Shuttle Display

Press Return, EXECUTE, or EXEC to exit the program.

If you're interested in seeing the BASIC program that draws the shuttle, type:

```
EDIT
```

Return, EXECUTE, or EXEC

Use the cursor control wheel, "mouse", or arrow keys to scroll through the program. Press unshifted-Stop, PAUSE, or PSE when you're finished.

Calculating While a Program Is Running

You can use your computer as a calculator even when the computer is busy running a program. Try this demonstration:

Try It Yourself

First, type the following line to clear any programs out of memory:

SCRATCH (Return), (EXECUTE), or (EXEC)

Now type:

EDIT (Return), (EXECUTE), or (EXEC)

to enter edit mode. Now enter the following six-line program that counts to ten forever. Press (Return), (ENTER), or (ENTER) after typing each line:

```
10  J=10
20  FOR I=1 TO J
30  PRINT I
40  NEXT I
50  GOTO 20
60  END
```

Now press **RUN** (unshifted-(f3)), (RUN), or (RUN) to run the program. The program counts too fast for you to read the display, so press unshifted-(Stop), (PAUSE), or (PSE) to pause the program for a moment and verify that it is counting to 10 over and over again. Press **Step** (unshifted-(f1)), (STEP), or (STEP) several times. Each time you press it, the program line displayed at the bottom of the screen is executed.

Now, press **Continue** (unshifted-(f2)), (CONTINUE), or (CONT)) to resume program execution. Type in the following arithmetic problem:

25+15 (Return), (EXECUTE), or (EXEC)

The computer delivers the answer, 40. Meanwhile, the program continues to run. Leave it running, and go on to the next section.

Changing Program Variables

You can interactively change program variables from the keyboard. The program you're running now counts to ten because the variable J is set to 10 in line 10. If you want the program to count from 1 to 20, just change the value of J by typing:

J=20 (Return), (EXECUTE), or (EXEC)

Now press unshifted-(Stop), (PAUSE), or (PSE) again to pause the program. Notice that the program is now indeed counting from 1 to 20. Finally, press (Shift)-(Stop), (STOP) or (STOP) to terminate the program.

Task Reference

Appendix A

This appendix provides quick-reference, task-oriented procedures. If you have worked through this guide or are familiar with BASIC programming, you should be comfortable with this section.

All the tasks described in detail in this guide are given here in condensed form. This appendix should serve to jog your memory when you want to perform a certain task.

Note

When a keypress is required, all three keyboard keys are given in this order: HP 46020A, HP 98203B, and HP 98203A.

The word "execute" in this appendix simply means to press the appropriate key on your keyboard: Return, EXECUTE, or EXEC, depending on your keyboard.

The index for this appendix follows:

ASCII Characters

To find an ASCII character from its decimal equivalent:

Press: **Any char**, (ANY CHAR), or (ANY C)

Type the three-digit decimal equivalent of the character you want displayed.

The character is displayed.

Example:

Press: **Any char**, (ANY CHAR), or (ANY C)

Type: `065`

The letter `A` is displayed.

Arithmetic Hierarchy

Your computer evaluates mathematical expressions in the following order:

1. Parentheses (if nested, inner-most parentheses evaluated first)
2. Exponentiation
3. Multiplication/division (left to right in order)
4. Addition/subtraction (left to right in order)

Example:

```
2+3*32/((7-6)+3)^2
2+3*32/(1+3)^2
2+3*32/4^2
2+3*32/16
2+96/16
2+6
8
```

Autostart File

Write an autostart program and store it by executing:

```
STORE "AUTOST"
```

This stores the autostart program on the system disc as a PROG file. It is automatically executed when the system is booted.

Binaries

Deleting

To delete all binaries from computer memory except the driver required for your display, execute:

```
SCRATCH BIN
```

Note

You cannot delete individual binaries.

Listing

To see what binaries are loaded in computer memory, execute:

```
LIST BIN
```

All binaries loaded in memory are displayed.

Loading

To load a binary from a disc, execute:

```
LOAD BIN "name:msus"
```

where **name** is the binary's file name, and **msus** is the mass storage unit specifier of the disc drive

The msus is unnecessary if the disc drive is specified as the default mass storage device (refer to msus in this appendix).

Example:

```
LOAD BIN "GRAPH:,701"
LOAD BIN "GRAPH"
```

Boot ROM

To determine the revision of your boot ROM, turn on your computer without a system connected.

If you have a boot ROM 3.0 or later, the revision is displayed.

If `UNABLE TO FIND SYSTEM` appears at the bottom of the display, you have a boot ROM of revision 1.0 or 2.0.

Booting BASIC

With Boot ROMs 1.0 or 2.0

If BASIC is the only system "on line", it is automatically booted at powerup.

If more than one system is "on line", the boot ROM lists the first letter of each system and waits 10 seconds for a key press to select a system. If no key is pressed, the first system found is booted.

Example:

```
WHICH SYSTEM?
B P H
```

In this example, the computer found three systems: BASIC, Pascal, and HPL. If none of the three keys is pressed within 10 seconds, the computer loads BASIC.

With Boot ROMs 3.0 and Later

The computer goes through a pre-determined search sequence at power-up and boots the first "on line" system it finds.

If more than one system is "on line", you **can** choose the system you want booted by pressing the space bar a few times while the computer is running its self-test. The computer responds by listing all "on line" systems preceded by two-character codes. Type the two-character code for the system you want booted.

Example:

```
:HP9895, 700, 0
   1P SYSPASCAL
:HP8290X, 702, 0
   1B SYSTEM_BA3
```

Type: `1B` to select BASIC

CRT

Configuration

To determine your CRT's configuration, execute:

```
SYSTEM$("CRT ID")
```

Your computer displays a response similar to this:

```
G:128HCGB
```

The number following the colon (`:`) specifies your display's character width (128 in the example).

The `H` specifies that your CRT has display enhancements; the absence of `H` means no display enhancements.

The `C` means your display has color capability; its absence means you have a monochrome display.

The `G` indicates graphics capability; you do not have graphics if no `G` appears.

The `B` means you have a bit-mapped display; otherwise, you have a non-bit-mapped display.

Hz Setting

To change the Hz setting of your display from 50 Hz to 60 Hz or vice versa, press the [CTRL] and [C] keys at the same time during self-testing or booting.

In response to the display, press either [5] to select 50 Hz or [6] to select 60 Hz.

Discs

Copying Flexible Discs with Two Drives

1. Insert source disc into one drive and initialized destination disc in the other drive.
2. Execute:

   `COPY ":source msus" TO ":destination msus"`

   where the **source msus** is the mass storage unit specifier of the drive containing the disc you want to copy, and **destination msus** is the mass storage unit specifier of the drive containing the initialized destination disc

Copying Flexible Discs with One Drive

1. Execute: `INITIALIZE ":MEMORY,0"`
2. Insert source disc into disc drive.
3. Execute: `COPY ":msus" TO ":MEMORY,0"`

   where **msus** is the mass storage unit specifier of the disc drive
4. When copying is done, remove the source disc from the disc drive and insert an initialized destination disc into the drive.
5. Execute: `COPY ":MEMORY,0" TO ":msus"`

   where **msus** is the mass storage unit specifier of the disc drive

Initializing

CAUTION

INITIALIZING A DISC DESTROYS DATA OR PROGRAMS RECORDED ON THE MEDIA. BE SURE YOU KNOW WHAT DISC YOU ARE USING. YOU CAN INADVERTENTLY WIPE OUT YOUR BASIC 3.0 LANGUAGE SYSTEM, OR OTHER FILES, IF YOU INITIALIZE THE WRONG DISC.

IF YOU INADVERTENTLY INITIALIZE A HARD DISC DRIVE, DO NOT INTERRUPT THE INITIALIZATION PROCESS. WAIT UNTIL THE DISC DRIVE LIGHT GOES OFF BEFORE CONTINUING. THIS CAN TAKE OVER AN HOUR. INTERRUPTING A HARD DISC DRIVE INITIALIZATION CAN SEVERELY DAMAGE THE DRIVE.

Some disc drives provide disc formatting options that are selected with the INITIALIZE statement. For example, you can select single-sided or double-sided when initializing a disc on the HP 9122 disc drive. Refer to your disc drive manual for details.

1. If you're initializing a flexible disc, insert the disc to be initialized into a flexible disc drive.
2. Execute: `INITIALIZE ":msus"`

   where **msus** is the mass storage unit specifier of the disc drive

---

Note

INITIALIZE, unlike other statements, **requires** the msus. This helps prevent inadvertent destruction of data by initializing the wrong disc.

Listing a Directory

To list the disc's directory, or catalog, execute:

```
CAT ":msus"
```

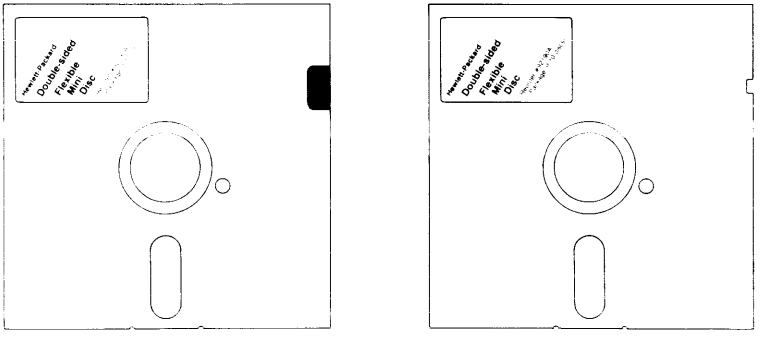
where **msus** is the mass storage unit specifier of the disc drive containing the disc you want to catalog

The msus is unnecessary if the disc drive is specified as the default mass storage device.

Example:

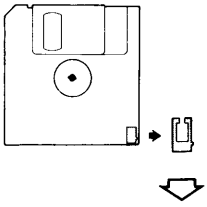
```
CAT ":,700,2"
CAT
```

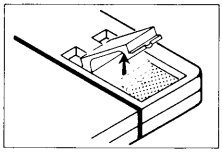
Write-protecting Flexible Discs



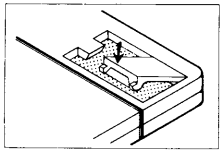
Write-protected **Write-enabled**

5¼-inch Mini-discs

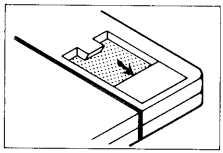




1. Weaken or score the attach point so as not to break the tab.
   Break off the write-protect tab.

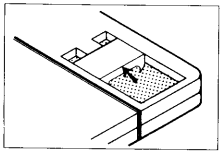


2. Align the protrusion on the tab with the groove in the disc.



Write-protect

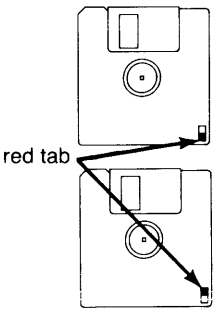
3. Depress the tab into the groove–tab should fit snugly. In the position shown, the tab write-protects the disc.



Write-enable

4. To write-enable the disc, slide the tab up.

3½-inch Single-Sided Micro-discs



To write-enable the disc, make sure you can see through the write-protect hole in the corner of the disc–the red tab should **not** block the hole.

To write-protect the disc, make sure the red tab shows through the write-protect hole in the corner of the disc–you should not be able to see through the hole.

3½-inch Double-sided Micro-discs

Display Enhancements

To determine if your computer has display enhancements, execute:

```
SYSTEM$("CRT ID")
```

If an `H` is included in the displayed response, you have display enhancements; if not, you don't.

Select display enhancements with the CHR$ function according to the following table. Note that the bit-mapped screens do not have blinking mode.

Character Code	Action Resulting from Displaying the Character
128	All enhancements off (except colors on a color CRT)
129	Inverse mode on.
130	Blinking mode on.
131	Inverse and Blinking modes on.
132	Underline mode on.
133	Underline and Inverse modes on.
134	Underline and Blinking modes on.
135	Underline, Inverse, and Blinking modes on.
136-143	Color enhancements on a color CRT.[1]

1 Additional color enhancements are available on bit-mapped color screens via CRT control registers.

Files, Copying

To copy one file from a disc to another disc, execute:

```
COPY "oldname:source msus" TO "newname:dest msus"
```

where **oldname** is the name of the file you are copying; **source msus** is the msus of the disc drive you're copying from; **newname** is the name of the duplicate file you're creating (can be same as **oldname**); and **dest msus** is the msus of the disc drive you're copying to.

Example:

```
COPY "WORK:,700" TO "WORK:,702"
```

The file named WORK is copied from a disc at address 700 to a disc at 702; the new file is also named WORK.

msus

Determining an msus

You can determine an msus in one of two ways:

- If you're determining the msus of a flexible disc drive, take all systems "off line". Insert your BASIC system disc into the drive and power up. The msus is listed in the upper right corner of the display.
- Refer to the BASIC System Worksheet in the *Peripheral Installation Guide.*

Setting a Default msus

To automatically refer all mass storage operations to a specific drive, execute:

```
MSI "msus"
```

where **msus** is the msus of the default drive. MSI is an abbreviation for MASS STORAGE IS.

Example:

```
MSI ",702"
```

could set the flexible disc drive of the HP 9135 as the default disc drive. All mass storage operations would default to that drive unless another msus was specified in the statement.

Printer

Setting Hard-copy Printer as Default Device

To specify a hard-copy printer as the default printer in your system, execute:

```
PRINTER IS PRT
```

if your printer is connected to interface select code 7, primary address 1; or

```
PRINTER IS device selector
```

if your printer is somewhere else.

device selector is the device selector of the printer.

Example:

```
PRINTER IS 701
```

selects the printer connected to the internal HP-IB (select code 7) that is set to primary address 1.

Setting CRT as Default Printer

To set the CRT as the default system printer, execute:

```
PRINTER IS CRT
```

Note that the CRT is automatically selected as the default printer at powerup.

Dumping to a Printer

To dump displays to a printer, use the DUMP ALPHA and DUMP GRAPHICS statements.

If you want to dump to a different printer than the printer at select code 7, primary address 1, execute:

```
DUMP DEVICE IS device selector
```

or

```
DUMP ALPHA #device selector
DUMP GRAPHICS #device selector
```

where **device selector** is the device selector of the printer you want to dump the display to.

Programs

Clearing Memory

To clear computer memory in preparation for writing a program, execute:

```
SCRATCH
```

Editing

To get into edit mode, either to begin writing a program or to edit an existing program, execute:

```
EDIT
```

Listing

To list a program, execute:

```
LIST
```

to list the program on the default printer; or

```
LIST #device selector
```

to list the program on the **device selector** printer if it is not the default printer

Example:

```
LIST #701
```

Loading

To load a program into computer memory from a mass storage device, execute:

```
LOAD "name:msus"
```

to load a PROG file;

where **name** is the program name and **msus** is the device containing the program (msus is unnecessary if it is the default device).

```
GET "name:msus"
```

to load an ASCII file.

Running

To run a program, execute:

```
RUN
```

or press: **Run**, RUN, or RUN

Your computer can perform calculations executed from the keyboard while a program is running.

You can execute changes in program variables at the keyboard while the program is running. The program continues to run with the new variable.

Storing

To store a program from computer memory to a mass storage device, execute:

```
STORE "name:msus"
```

to store a PROG file;

where **name** is the program name and **msus** is the device containing the program (msus is unnecessary if it is the default device).

```
SAVE "name:msus"
```

to store an ASCII file.

Softkeys, HP 46020A

The [Menu] and [System] keys control softkeys on the HP 46020A keyboard.

[System] selects System softkey mode.

[User] selects User 1 mode.

When the keyboard is in a user mode, [Shift]-[Menu] cycles softkey menus through User modes (User 1 → User 2 → User 3 → User 1, etc.).

Unshifted-[Menu] toggles the softkey display on and off, irrespective of the softkey mode.

System, Configuring a

You can configure a BASIC system by:

- Hand-loading the system and all required BIN files on powerup.
- Loading the system and required BIN files into your computer memory and executing:

  ```
  STORE SYSTEM "filename:msus"
  ```

  where **filename** is the name of your system file; and where **msus** is the mass storage unit specifier of your storage device

  filename must have a prefix of SYSTEM_ if you have a 1.0 or 2.0 boot ROM; otherwise, it must have a prefix of SYSTEM_ or SYS
- Writing an autostart program that automatically loads predetermined BIN files when the discs are inserted. Store the autostart file on the *System Disc* by executing:

  ```
  STORE "AUTOST"
  ```

- Using the CONFIGURE program (see the *Software and Manual Catalog*).

BASIC Documentation

Appendix B

As with most products, learning how to use the manuals properly will help you to get the most from the product. In order to use the manuals most effectively, you should know both the objective and content of each manual. This appendix lists the manual set and gives a brief description of each of the major manuals.

BASIC Manuals

Part Number	Description
98613-90011	BASIC Programming Techniques
98613-90021	BASIC Interfacing Techniques
98613-90031	BASIC Graphics Techniques
98613-90041	BASIC User's Guide
98613-90051	BASIC Language Reference
98613-90061	BASIC Condensed Reference
98613-90071	BASIC Master Index
98613-90081	BASIC Software and Manual Catalog
98613-90090	CSUB Preparation Manual
98613-90091	BASIC Utilities Library Manual
98613-90092	Loader Utility Manual
98613-90093	Demo Instructions Manual

BASIC Programming Techniques described writing, editing, storing, running and debugging BASIC programs. The manual also describes such programming topics as string and math operations, using the real-time clock, and communicating with the operator.

BASIC Interfacing Techniques describes how to communicate with external devices. Both general and interface-specific techniques are described in the manual.

Read Chapter 1, "Manual Overview", to see this manual's objectives and contents. This chapter also describes the organization of information in the manual and briefly describes each chapter. You may want to scan chapters of interest in the main part of the manual. The "Useful Tables" contains information relevant to interfacing, and the Index provides an index to the topics in this manual.

BASIC Graphics Techniques describes using the graphics capabilities of Series 200/300 computers. Plotting on the CRT and on external graphics devices are fully described in this manual, as well as using external graphics input devices.

Chapter 1, "Introduction to Graphics", describes the objectives of the manual and assumptions made about your knowledge of BASIC programming. You may want to scan the individual chapters of the manual as your interest dictates. An index is also provided in this manual.

BASIC User's Guide introduces you to BASIC on your Series 200/300 computer. It describes many of the operating features such as keyboards, displays, and discs. It show you how to do simple operations on the computer.

BASIC Language Reference provides a complete "dictionary" of precise descriptions of every keyword in the Series 200/300 BASIC language. Drawings graphically show the proper syntax of each keyword, and any parameters are described in an accompanying table. The semantics section describes the resultant action of different keyword syntaxes.

The "Keyword Dictionary" section is the main part of this manual, providing the following three sections: 1) "Language History", which provides valuable information about how and when the language has been revised and updated; 2) "Using the Keyword Dictionary", which describes what information is provided by the dictionary and explains how to use it; and finally 3) the actual dictionary entries. You should read the first of these sections before attempting to use the rest of the manual.

The "Glossary" provides concise definitions of technical terms used throughout the manual set, which you can refer to as you encounter unfamiliar terms. The "Interface Registers" section contains listings of all status and control registers of I/O paths, CRT, keyboard, and optional interfaces. The contents of the "Useful Tables" and "Error Messages" sections are self-evident. The "Keyword Summary" section provides a complete list of keywords in the Series 200/300 BASIC language, grouped according to the function that it performs.

Glossary Of Terms

Alpha Display: The part of the screen defined by BASIC for displaying normal text characters (letters and numbers).

Binary Programs: Programs written in machine language which extend the command set and capabilities of a language system.

Bit: A binary digit (1 or 0).

Bit-mapped Display: A display with a graphics scheme which allows the user to select (turn on or off) a single pixel, or point, on the display. The Model 237, HP 98700 Graphics Display Station, and Series 300 computers have a bit-mapped display.

Boot ROM: The Boot ROM stores instructions that tell the computer how to search for a system program. They basically keep the computer running until a system program can take over.

Bus Address: A number that identifies the location of a device on the HP-IB; also called primary address.

Byte: The unit of memory used on your computer. One byte equals eight bits.

Character Code: A numeric code which is used to represent a character inside the computer.

CRT: The computer's screen (cathode ray tube).

Cursor: The underline character that marks the position on the screen where the next character will be typed.

Default Device: If you execute an I/O operation (e.g., LOAD, PRINT) without specifying which device the operation refers to, the computer *assumes* you want to use the default device and carries out the command. When loading a program from the default disc drive, for example, you don't have to include the msus–the computer *assumes* the msus of the default drive.

Disc: Similar to a phonograph record, except that it stores programs and data instead of music.

Disc Drive: An input/output device that transfers programs and data between a disc and the computer's memory.

Graphics Display: The part of the screen defined by BASIC for drawing graphs, charts, and other pictorial displays.

Hardware: All of the electrical and mechanical components of the computer.

Human Interface: The part of a program that handles all interaction between the computer and the person who operates it. A good human interface makes using the computer easy.

Input/Output: Anything relating to the exchange of information between the computer and its peripherals.

Input Device: A peripheral device which transfers programs and data into the computer. Common input devices include keyboards, disc drives, and graphics tablets.

K bytes: 1 024 bytes.

Language System: A large program which performs all of the functions of a system program, plus supports a programming language like BASIC or Pascal.

Language-dependent Program: A program which requires a language system in order to run. Language-dependent programs are always loaded into memory *after* a language system has been booted.

Machine Language: The language the computer's processor understands, expressed in terms of bits (1s and 0s).

Mass Storage Device: For most applications, a mass storage device is simply a disc drive.

Mass Storage Unit Specifier: A string of characters used to designate a particular disc drive (or other mass storage device). Abbreviated as msus.

Memory: The area of the computer where programs and data are stored. The processor cannot run a program unless it is in memory.

Memory Address: A number which uniquely identifies one byte of memory.

Menu: Usually, a list of things that a program can do. You control the progress of the program by selecting one of the menu items.

Mouse: A hand-operated device that moves the cursor in all directions and scrolls the display. It also has programmer-definable buttons.

msus: See Mass Storage Unit Specifier.

Output Device: A peripheral device which accepts information from the computer for storage or display purposes. Common output devices include computer screens, disc drives, printers and plotters.

Peripheral Devices: Devices that allow the computer to communicate with the outside world. See "Input Device" and "Output Device."

Primary Address: See Bus Address.

Program: A set of instructions that tell the processor how to perform a particular task. Most programs are written in a high-level programming language like BASIC.

Program Listing: A list of all statements in a program.

Prompt: A message that a running program displays on the screen when it needs data or other information from you.

RAM: Random Access Memory. This is erasable program memory. Programs and data are usually copied into RAM from a disc drive or other mass storage device, executed by the processor, and then erased from RAM. When the power is turned off, RAM is erased.

ROM: Read-Only Memory. This is permanent program memory, used primarily for storing essential programs. Programs in ROM are never erased, so ROM is not reusable.

Select Code: A number which uniquely identifies an interface. The processor uses the select code to select which interface will be used in a data transfer operation.

Software: A synonym for program.

System Program: A program which handles all of the overhead functions of computing, such as defining the keyboard, managing the peripherals, refreshing the display, etc. When the computer is turned on and passes its self-test, it immediately begins searching for a system program to boot.

Stand-alone Program: A program that has a "built-in" system program and can run without any underlying language support.

Subject Index

a

b

C

d

e

f

g

h

i

k

l

m

n

o

p

r

S

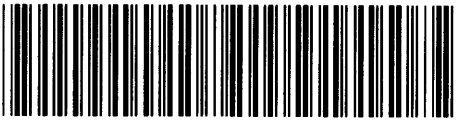
t

u

V

W

X



Reorder Number
98613-90041
Printed in U.S.A. 9/85

98613-90663
Mfg. No. Only