Chapter 8

Number bases

Entering and displaying binary integers

You can enter and display integers in one of four forms:

- decimal (base 10)
- hexadecimal (base 16)
- octal (base 8) or
- binary (base 2).

You use the pound symbol (#) together with a suffix to indicate the base of a number. The suffixes are:

d (decimal)

h (hexadecimal)

o (octal) and

b (binary).

For example, # 182d, # B6h, # 266o, and # 10110110b are all ways of representing 182. In this guide, numbers represented in this way are called *binary integers* regardless of their base.

The current base setting determines which base is used to display integers.

On the HP 49G, binary integers can take from 1 to 64 bits, depending on the current *wordsize*.

To set the base

To set the base, select the appropriate command from the command catalog (CAT):

- DEC (to set the base to decimal)
- HEX (to set the base to hexadecimal)
- OCT (to set the base to octal)
- BIN (to set the base to binary).

The settings for flags –11 and –12 determine the current base. The DEC, HEX, OCT, and BIN commands control the settings of these flags:

		–11	
		Set	Clear
–12	Set	Hex	Bin
	Clear	Oct	Dec

An annunciator on your default screen indicates the current base setting.

To set the wordsize

1. Press ↱ BASE STWS.
2. Key in the new wordsize (from 1 to 64).
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

If a binary integer argument exceeds the current wordsize, the excess leading bits are dropped before the command is executed. If necessary, results are also truncated.

To recall the current wordsize

1. Press ↱ BASE RCWS.
2. Press ENTER.

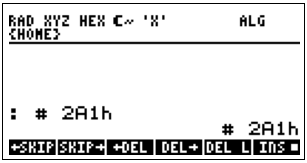
In RPN mode: follow step 1 only.

To enter a binary integer

1. Press ↰ #.
2. Enter the value of the binary integer.
3. Enter the suffix that indicates the base you want to use: d, h, o, or b. Your entry must be a lower-case character.

> Step 3 is optional if the base you want to use is the same as the current base setting.

4. Press ENTER.



```
RAD XYZ HEX C~ 'X'        ALG
{HOME}

: # 2A1h
                       # 2A1h
```

Binary integers are displayed on the HP 49G with a space after the # sign. You do not need to enter a space when creating a binary integer.

If you specify a base other than the current base setting, the HP 49G converts the binary integer you entered to an integer to the base of the current setting. If you want to see the binary integer you entered, press ↰ CMD.

For example, if your current base setting is hexadecimal and you enter # 1101b, the calculator displays your entry as # Dh. To see your entry, press ↰ CMD. Your entry is at the top of the list of last four executed commands.

Binary integer arithmetic

To add, subtract, multiply, or divide two binary integers

1. Enter the first binary integer.
2. Press the key for the operation you want to perform: + − X or ÷.
3. Enter the second binary integer.
4. Press ENTER.

In RPN mode: follow steps 1, 3, and 2.

The two binary integers do not have to be to the same base.

The answer is expressed in the current base setting, and the two arguments are converted to that setting.

Note that any remainder from a division is lost and the answer is truncated to be an integer.

To find the negative of a binary integer

1. Press CAT NEG.
2. Enter the binary integer.
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

The negative of a binary number is its *two's complement* (all bits inverted and 1 added).

To convert a binary integer to a different number base

1. Press (CAT) *n*, where *n* is the command that represents the base you want to convert to: DEC, BIN, HEX, or OCT.
2. Enter the binary integer.
3. Press (ENTER).

For example, to convert # 1101b to hexadecimal, enter HEX(# 1101b) and press (ENTER).

In RPN mode: follow steps 2 and 1.

Note that converting a binary integer to a different number base also changes the base setting to the base that you converted the integer to.

To convert a binary integer to a real number

1. Press ⮣ (BASE) B→R.
2. Enter the binary integer.
   For example, # 3Ah.
3. Press (ENTER).
   The answer is 58.

In RPN mode: follow steps 2 and 1.

To convert a real number to a binary integer

1. Set the base to the type that you want to convert to. See "To set the base" on page 8-1.
2. Press ⮣ (BASE) R→B.
3. Enter the real number.
4. Press (ENTER).

In RPN mode: follow steps 2 and 1.

If necessary, the real number is rounded to an integer before converting. Negative real numbers are converted to # 0 and real numbers $\geq 1.84467440738 \times 10^{19}$ are converted to the largest binary integer (# FFFFFFFFFFFFFFFFH, for example).

Using Boolean operators

The HP 49G provides a number of commands that enable you to perform Boolean operations and comparisons on binary integers. These commands—available by pressing ↱ BASE LOGIC—are illustrated in the following table. The input syntax shown assumes that you are in algebraic mode.

Commands	Examples	
	Input	Output
AND Logical bit-by-bit AND of two arguments. Compares corresponding bits and returns true (1) if both bits are 1.	# 1100b AND # 1010b	# 1000b
NOT Returns the one's complement of the argument. Each bit in the result is the complement of the corresponding bit in the argument.	NOT (#111b)	#111111110000b[a]
OR Logical bit-by-bit OR of two arguments. Compares corresponding bits and returns true (1) if either bit is 1.	# 1100b OR # 1010b	# 1110b
XOR Logical bit-by-bit exclusive OR of two arguments. Compares corresponding bits and returns true (1) if only one of the bits is 1.	# 1101b XOR # 1011b	# 110b

[a] This answer assumes that the wordsize has been set to 12.

Manipulating bits and bytes

The following commands enable you to manipulate binary integers one bit or one byte at a time. The commands are available by pressing ↱ BASE BIT OR ↱ BASE BYTE. Unless otherwise stated, each example assumes the wordsize is set to 24.

Commands	Example	
	Input	Output
ASR Arithmetic Shift Right. Performs 1 bit arithmetic right shift. The most significant bit is regenerated.	# 1100010b	# 110001b
RL Rotate Left. Binary integer rotates left one bit. (The example assumes that the wordsize is 4.)	# 1100b	# 1001b
RLB Rotate Left Byte. Binary integer rotates left one byte.	# FFFFh	# FFFF00h
RR Rotate Right. Binary integer rotates right one bit. (The example assumes that the wordsize is 4.)	# 1101b	# 1110b
RRB Rotate Right Byte. Binary integer rotates right one byte.	# A0B0C0h	C0A0B0h
SL Shift Left. Binary integer shifts left one bit.	# 1101b	# 11010b
SLB Shift Left Byte. Binary integer shifts left one byte.	# A0B0h	# A0B000h
SR Shift Right. Binary integer shifts right one bit.	# 11011b	# 1101b
SRB Shift Right Byte. Binary integer shifts right one byte.	# A0B0C0h	# A0B0h