2.13 Binary

Normally, a programmer can think in terms of base ten numbers. However, a compiler must allocate some finite quantity of bits (e.g., 32 bits) for a variable, and that quantity of bits limits the range of numbers that the variable can represent. Thus, some background on how the quantity of bits influences a variable's number range is helpful.

Because each memory location is composed of bits (0s and 1s), a processor stores a number using base 2, known as a ***binary number***.

For a number in the more familiar base 10, known as a ***decimal number***, each digit must be 0-9 and each digit's place is weighed by increasing powers of 10.

Table 2.13.1: Decimal numbers use weighed powers of 10.

Decimal number with 3 digits	Representation
212	$= 2 \cdot 10^2 \quad + 1 \cdot 10^1 \quad + 2 \cdot 10^0$ $= 2 \cdot 100 \quad + 1 \cdot 10 \quad + 2 \cdot 1$ $= 200 \quad + 10 \quad + 2$ $= 212$

Feedback?

In ***base 2***, each digit must be 0-1 and each digit's place is weighed by increasing powers of 2.

Table 2.13.2: Binary numbers use weighed powers of 2.

Binary number with 4 bits	Representation
1101	$= 1 \cdot 2^3 \quad + 1 \cdot 2^2 \quad + 0 \cdot 2^1 \quad + 1 \cdot 2^0$ $= 1 \cdot 8 \quad + 1 \cdot 4 \quad + 0 \cdot 2 \quad + 1 \cdot 1$ $= 8 \quad + 4 \quad + 0 \quad + 1$ $= 13$

Feedback?

The compiler translates decimal numbers into binary numbers before storing the number into a memory location. The compiler would convert the decimal number 212 to the binary number 11010100, meaning 1*128 + 1*64 + 0*32 + 1*16 + 0*8 + 1*4 + 0*2 + 0*1 = 212, and then store that binary number in memory.

PARTICIPATION ACTIVITY | 2.13.1: Understanding binary numbers. ✓

Set each binary digit for the unsigned binary number below to 1 or 0 to obtain the decimal equivalents of 9, then 50, then 212, then 255. Note also that 255 is the largest integer that the 8 bits can represent.

0	0	1	1	0	0	1	1
128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

51

(decimal value)

Feedback?

PARTICIPATION ACTIVITY | 2.13.2: Binary numbers. ✓

1) Convert the binary number 00001111 to a decimal number.

15

Check Show answer

Correct ✓

15

8 + 4 + 2 + 1 is 15

2) Convert the binary number 10001000 to a decimal number.

136

Check Show answer

Correct ✓

136

128 + 8 is 136

3) Convert the decimal number 17 to an 8-bit binary number.

Correct ✓

00010001

00010001

Check Show answer

16 + 1 is 17

4) Convert the decimal number 51 to an 8-bit binary number.

00110011

Check Show answer

✓

Correct

00110011

32 + 16 + 2 + 1 is 51

Feedback?