# 1.6 Floating-point numbers

#### Floating-point numbers and normalized scientific notation

An *integer* is a whole number, like 42, 0, or -95. A *floating-point number* is a real number, like 98.6, 0.0001, or -666.667. The point" refers to the decimal point being able to appear anywhere ("float") in the number.

- Integers are typically used for values that can be counted, like 42 cars, 0 pizzas, or -95 days.
- Floating-point numbers are typically used for values that are measured, like 98.6 degrees, 0.00001 meters, or -666.66

To improve readability and consistency, floating-point numbers are commonly written using **normalized scientific notation**.  $10^{1}$ ,  $1.0 \times 10^{-4}$ , or  $-6.66667 \times 10^{2}$ , where the number is written as a digit (+/- 1 to 9), decimal point, fractional part, times  $10^{1}$  term "normalized" is in contrast to non-normalized where more than one digit, or a 0, may precede the decimal point, such a  $0.1 \times 10^{-3}$ .

The parts of scientific notation are named **significand** for the part before  $\times$  and **exponent** for the power of 10: significand  $\times$  exponent is 0, the power of ten part is sometimes omitted, as in 5.7.

In binary, normalized scientific notation consists of  $1.f \times 2^{\text{exponent}}$ , like  $1.010 \times 2^5$ . f is the fractional part.

PARTICIPATION ACTIVITY

1.6.1: Normalized scientific notation: Decimal.

Indicate which numbers are in decimal normalized scientific notation.

1) 2.05 × 10<sup>3</sup>

O Yes
O No

2) 0.50 × 10<sup>3</sup>
O

	Yes	
	O No	
3) 2	$8 \times 10^3$	_
	O Yes	
	O No	
4) 3		_
	O Yes	
	O No	
5) -	$77 \times 10^3$	_
	O Yes	
	O No	
6) 2	15 × 10 <sup>-3</sup>	_
	O Yes	
	O No	
7) (		_
	O Yes	
	O No	

PARTICIPATION ACTIVITY

1.6.2: Normalized scientific notation: Binary.

Indicate which numbers are in binary normalized scientific notation.

1) 0.0

O Yes

2)	$1.01 \times 2^3$	
	O Yes	
	O No	

O No

3) 
$$0.10 \times 2^3$$

O	Yes

4) 
$$11.10 \times 2^3$$

5) 
$$-1.01 \times 2^3$$

6) 
$$1.01 \times 2^{-3}$$

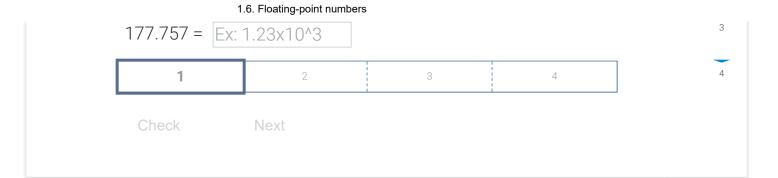
CHALLENGE ACTIVITY

1.6.1: Normalized scientific notation.

Start

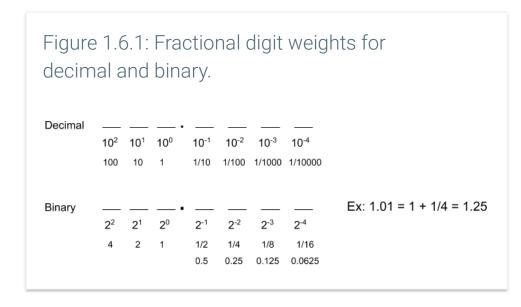
Write the number in normalized scientific notation.

Use  $^{\circ}$  for exponents. Ex: 10 $^{\circ}$ 4 for  $\mathbf{10^4}$ .



#### **Fractions in binary**

Fractions in binary are similar to fractions in decimal. In decimal, each digit after the decimal point has weight  $1/10^1$ ,  $1/10^2$  binary, each digit after the binary point has weight  $1/2^1$ ,  $1/2^2$ ,  $1/2^3$ ,  $1/2^4$ , etc. (so 1/2, 1/4, 1/8, 1/16, etc.). Ex: 1.1101 is 1 + 1 Note that for binary numbers, the "dot" is called a **binary point** (versus decimal point for decimal numbers) The general term



Converting decimal to binary or vice-versa when fractions are involved uses the same process as without fractions. Howev manually converting a decimal with a fraction into binary, converting the whole and fraction parts separately, then concater easier. Ex: For 12.25, 12 is 1100<sub>2</sub>, and 0.25 is 0.01<sub>2</sub>, yielding 1100.01.

If a decimal fraction cannot be exactly represented as a binary fraction using limited bits, one gets as close as possible, over To represent 0.8 with only four binary fraction bits:

- 0.1100 is 1/2 + 1/4 = 0.75, which is under by 0.05.
- 0.1110 is 0.875, which is over by 0.075.
- 0.1101 is 0.8125, also over but only by 0.0125, which is closest and so is the best representation.

Obviously, more bits means binary fraction values can be closer.

PARTICIPATION ACTIVITY 1.6.3: Fractions in binary.	_
1) 1.1 binary = decimal. Type as: #.#	•
Check Show answer	
2) 0.001 binary = decimal. Type as: 0.###	•
Check Show answer	
3) 10.101 binary = decimal. Type as: #.###	-
Check Show answer	
4) 0.75 decimal = binary. Type as: 0.##	_
Check Show answer	

5) 0.625 decimal = \_\_\_\_\_ binary. Type as: 0.### Check Show answer 6) 16.75 decimal = \_\_\_\_ binary. Type as: #####.## Check Show answer 7) 0.6 decimal = \_\_\_\_ binary, using two fraction bits. Type as: 0.## Check **Show answer** 

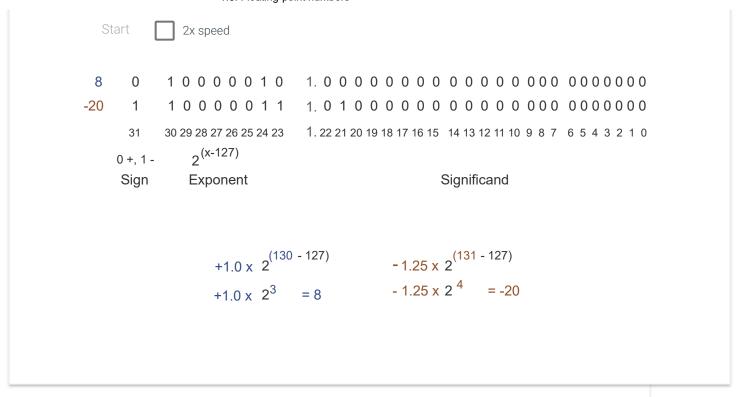
### **Binary floating-point representation**

In a computer, a binary floating-point number must be represented using a fixed number of bits. Normalized scientific notar commonly using 32 or 64 bits. A common 32-bit floating-point binary representation has these items:

- Sign: 1 bit. 0 means positive, 1 negative.
- Exponent: 8 bits. Instead of two's complement, the exponent is biased, meaning a fixed value is subtracted, in this casexponent of 00000000 means  $2^{0-127} = 2^{-127}$ , and of 1111111 means  $2^{255-127} = 2^{128}$ .
- Fraction: 23 bits. Because the significand's first digit is always 1, the 1 implicitly precedes the fraction and thus isn't expresented. The significand in scientific notation is commonly called the *mantissa*.

PARTICIPATION ACTIVITY

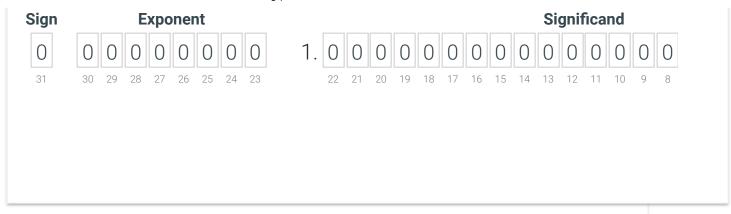
1.6.4: A 32-bit floating-point representation.



The above is known as the *IEEE single-precision binary floating-point* format, which uses 32 bits: 1 bit for sign, 8 bits for each by 127), and 23 bits for significand (leading 1 before binary point is implicit). *IEEE double-precision binary floating-point* for bits: 1 bit for sign, 11 bits for exponent, and 52 bits for significant (leading 1 before binary point is implicit).

Note: In C, C++, and Java, a variable like "float x" uses 32-bits (single precision), while "double x" uses 64 bits (double precision). Programmers usually use double to obtain more significand precision, unless memory is tightly constrained.

PARTICIPATION ACTIVITY	I 6 5. Representation of single-precision floating-point values		
Enter a decim	al value:		
Convert			



**PARTICIPATION** 1.6.6: Binary floating-point to decimal. **ACTIVITY** Given the following binary floating-point representation, determine the following. 0 10000001 001000000000000000000000 1) Sign bit (type 0 or 1) Check **Show answer** 2) Exponent bits Check **Show answer** 3) Exponent in decimal Check **Show answer** 

4) Significand's fraction bits (consider copy-pasting)

**Show answer** 

5) Decimal scientific notation

× 2<sup>2</sup>

**Check** Show answer

6) Decimal number

Check

Check Show answer

PARTICIPATION ACTIVITY

1.6.7: Negative exponents.

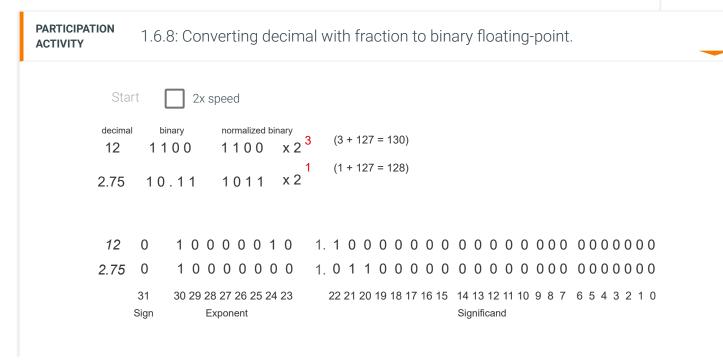
A 32-bit binary floating-point number has an exponent of 01111101 (125). Assume the sign bit is 0 and the significand is 1.000...

- 1) The number is  $+1 \times 2^{?}$ 
  - **O** 125
  - **O** 127
  - **O** 2
  - **O** -2
- 2) The number in base ten is \_\_\_\_\_.

0.25
1.000...
-2
3) In the 32-bit representation, the sign bit applies to \_\_\_\_\_\_.
the significand
the exponent
both the significand and the exponent

## **Decimal to binary floating-point**

Decimal is converted to 32-bit floating-point by first converting the decimal number to binary, then normalizing and adjustir and finally filling in the appropriate fields.



**O** 131

PARTICIPATION ACTIVITY

1.6.10: Decimal to binary floating-point.

Consider converting the decimal number -16.25 to the above 32-bit binary floating-point format. Note that -16.25 is -10000.01 in binary.

10000011

4

1.000001000...

1

000001000...

131

Sign bit

Unbiased exponent in base ten

Biased exponent in base ten

**Exponent bits** 

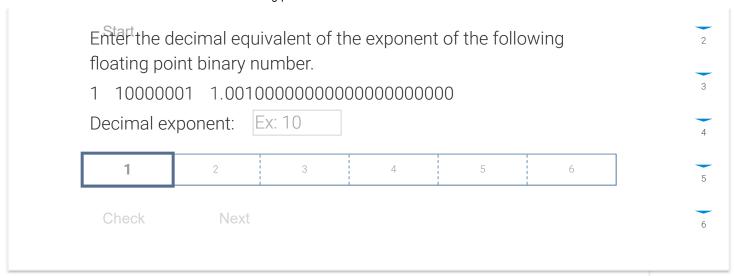
Fraction bits

Significand

Reset

CHALLENGE ACTIVITY

1.6.2: Binary floating point.



#### Exploring further:

- IEEE single-precision binary floating-point format
- IEEE double-precision binary floating-point format
- Provide feedback on this section