FLOATING POINT VARIABLE

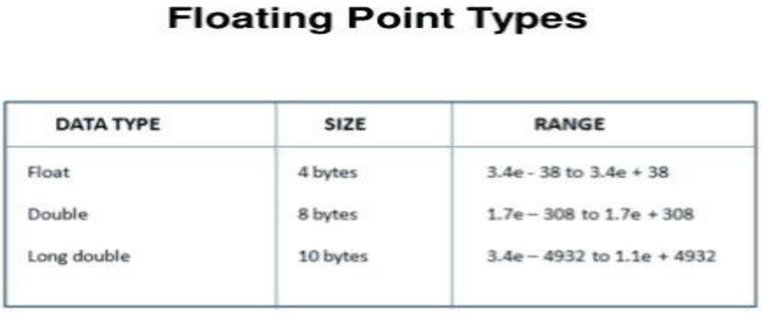
In computing, **floating point** is the formulaic representation that approximates a real number so as to support a trade-off between range and precision. A number is, in general, represented approximately to a fixed number of significant digits (the significand) and scaled using an exponent in some fixed base; the base for the scaling is normally two, ten, or sixteen. A number that can be represented exactly is of the following form:

$$\text{Significand x base}^{\text{exponent}} \ ,$$

where significand is an integer, base is an integer greater than or equal to two, and exponent is also an integer. For example:

$$0.0001 = 1 \text{ x } 10^{-4}$$

Real numbers are represented in C language by the floating point type float, double and long double.



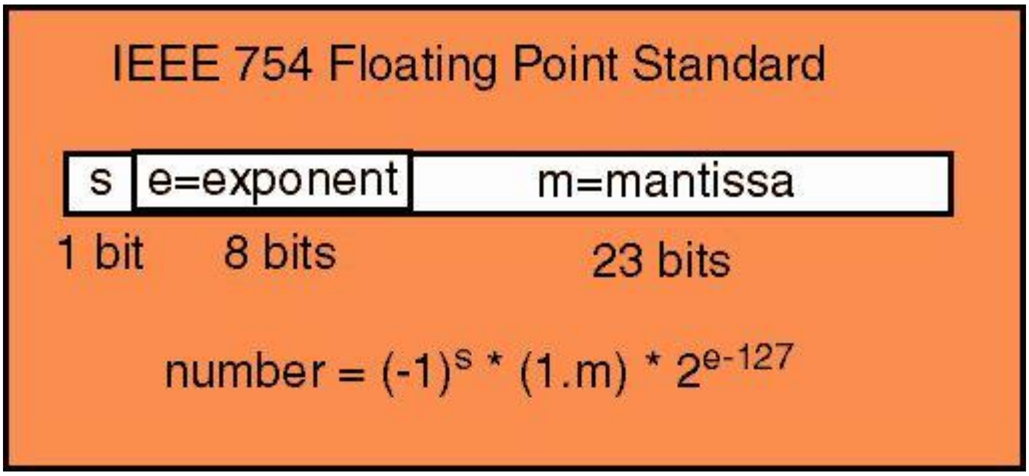
Floating Point Types

DATA TYPE	SIZE	RANGE
Float	4 bytes	3.4e - 38 to 3.4e + 38
Double	8 bytes	1.7e – 308 to 1.7e + 308
Long double	10 bytes	3.4e – 4932 to 1.1e + 4932

In main storage and in disk storage, a float is represented with a 32-bit pattern, double with a 64-bit pattern and long double with a 80-bit pattern.

According to IEEE-754 floating point standard,

- The sign is a single bit.
- The exponent is stored as an unsigned integer. For 32-bit floating point values, this field is 8 bits. 1 represents the smallest exponent and all ones, the largest.
- Considering significand, all the possible values start with a 1. This means that there is no need to store it. The rest of the binary digits are stored in an integer field. For 32-bit value, this field is 23 bits.



IEEE 754 Floating Point Standard

s	e=exponent	m=mantissa
1 bit	8 bits	23 bits

number = $(-1)^{s} * (1.m) * 2^{e-127}$