Numbers in Floating Point:

The standard to represent numbers in Floating Point is accepted by the machine. IEEE-754. In this standard, there exists 2 ways to represent numbers:

* SINGLE PRECISION FLOAT represents with 4 bytes (32 bits)
* DOUBLE PRECISION FLOAT reps with 8 bytes (64 bits)

1. Representation of a number in SINGLE FLOAT

31|30 23|22 0

|  |  |  |
| --- | --- | --- |
| S | Exponent | Fraction |

1 bit 8 bits 23 bits

Aka Float Form

Meaning of Fields:

1. The MSB bit, indicated by S, is the bit sign.

S = 0 🡪 positivie

S = 1 🡪 negative

1. Fraction AKA Mantissa or Significant

When calculating, we take the 23 bits of Mantissa from the right of the point and add a 1 from the left to the point as in it receives a number: 1.mantissa

The 1 added during the time of calculation is also known as a hidden bit because it is not saved in the memory.

In order to represent in Float form, none of the 23 bits in the mantissa field has been converted into decimal point.

Mistaking so that the fraction indicated by X is converted to Float point (to the Mantissa field) may be 1 <= x <= 2.

1. Representing the number 0:

According to its definition in 2), we don’t need to represent 0 because the mantissa’s and the exponent’s value will be 0.

1. Representing the Exponent:

The Exponent in Float is a binary number of size 8 in unsigned representation as in the domain of the values are between 0 – 255.

Important Remark: the Exponent is equal to the mathematical exponent through which is considered the number’s value.

Let us define: math exponant = Exponent – 127

E = math exponant

Remarks:

1. If the Exponent = 0, we keep it 0.
2. If the Exponent = 255, we keep it to the unique values of +/- ∞

Formula to Calculate the Decimal Value of the Binary Number that is Represented in Float Form:

Text

Description automatically generated

In the expression n it is the decimal number to represent.

The letter s is the bit for the sign (if s is 0, the expression (-1) raised to 0 results in 1 positive).

The letter e is the exponent and m is the mantissa.

Floating-point type specifiers fall into the following categories:

* Real floating-point types
* Complex floating-point types

Real floating-point types:

Generic, or binary, floating-point types consist of the following:

* float
* double
* long double

Decimal floating-point types consist of the following:

* \_Decimal32
* \_Decimal64
* \_Decimal128

If a floating-point constant is too large or too small, the result is undefined by the language.

The declarator for a simple floating-point declaration is an identifier. Initialize a simple floating-point variable with a float constant or with a variable or expression that evaluates to an integer or floating-point number.

 You can use decimal floating-point types with any of the operators that are supported for binary floating-point types. You can also perform implicit or explicit conversions between decimal floating-point types and all other integral types, generic floating-point types, or packed decimals. However, there are restrictions on the use of decimal floating-point types with other arithmetic types as follows:

* You cannot mix decimal floating-point types with generic floating-point types or complex floating-point types in arithmetic expressions, unless you use explicit conversions.
* Implicit conversion between decimal floating-point types and real binary floating-point types is only allowed via assignment, with the simple assignment operator =. Implicit conversion is performed in simple assignments, which also include function argument assignments and function return values. See [Floating-point conversions](https://www.ibm.com/docs/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.cbclx01/cplr070.htm) for details.

Complex floating-point types:

Complex floating-point types are introduced in the C99 standard.

The complex floating-point type specifiers are as follows:

* float \_Complex
* double \_Complex
* long double \_Complex

The representation and alignment requirements of a complex type are the same as an array type containing two elements of the corresponding real type. The real part is equal to the first element; the imaginary part is equal to the second element.

The equality and inequality operators have the same behavior as for real types. None of the relational operators may have a complex type as an operand.

The numeric representations described in previous sections of this chapter are for integer values. C has three numeric types that have fractional parts:

❯ float single-precision floating point

❯ double double-precision floating point

❯ long double extended-precision floating point.

Floating-point values are stored using a binary version of scientific notation.

## What is ASCII code?

The full form of ASCII is the **American Standard Code for information interchange**. It is a character encoding scheme used for electronics communication. Each character or a special character is represented by some ASCII code, and each ascii code occupies 7 bits in memory.

In [C programming language](https://www.javatpoint.com/c-programming-language-tutorial), a character variable does not contain a character value itself rather the ascii value of the character variable. The ascii value represents the character variable in numbers, and each character variable is assigned with some number range from 0 to 127. For example, the ascii value of 'A' is 65.

What is ASCII extended ASCII?

Extended ASCII (EASCII or high ASCII) character encodings are eight-bit or larger encodings that include the standard seven-bit ASCII characters, plus additional characters.

Extended ASCII is a version that supports representation of 256 different characters. This is because extended ASCII uses eight bits to represent a character as opposed to seven in standard ASCII (where the 8th bit is used for error checking).

ASCII has 256 this would be the case in extended. ... ASCII has its equivalent in Unicode. The difference between ASCII and Unicode is that ASCII represents lowercase letters (a-z), uppercase letters (A-Z), digits (0–9) and symbols such as punctuation marks while Unicode represents letters of English, Arabic, Greek etc.

What is Unicode?

Unicode, formally the Unicode Standard, is an [information technology](https://en.wikipedia.org/wiki/Information_technology) [standard](https://en.wikipedia.org/wiki/Technical_standard) for the consistent [encoding](https://en.wikipedia.org/wiki/Character_encoding), representation, and handling of [text](https://en.wikipedia.org/wiki/Character_(computing)) expressed in most of the world's [writing systems](https://en.wikipedia.org/wiki/Writing_system). The standard, which is maintained by the [Unicode Consortium](https://en.wikipedia.org/wiki/Unicode_Consortium), defines 144,697 characters covering 159 modern and historic [scripts](https://en.wikipedia.org/wiki/Script_(Unicode)), as well as symbols, [emoji](https://en.wikipedia.org/wiki/Emoji), and non-visual control and formatting codes.