**Question 1.** In Floating point representation, we have three components

1.The Sign Bit

2.Exponent

3.Fractional Part

Precision is one the prime attribute of any Floating-point Representation,

Does any of the above three components play a role in the defining the precision of the number? If so, which are the component or Components which play the role in defining precision and how? Explain this with example in your own words

**Answer:**

Precision defines accuracy of the floating-point number. That is 5.9532 is more accurate than 5.95, which is more precise than 5.9. Thereby, it is the fractional part of the floating-point number that governs the precision.

The single-precision floating-point format of 32-bits has fractional part of 23 bits and the double precision floating-point format of 64 bits has fractional part of 52 bits.

**Question 2.** What is Normal and Subnormal Values as per IEEE 754 standards explain this with the help of number line.

**Answer:**

There are many ways of representing a number, for example, 0.1 could be represented as 1\*10-1 or 0.01 \* 101.

**Normal numbers** are those which are always stored with the leading bit as a one. Here it would be 1\*10-1.

But, if the exponent bits are all ‘0’ and the leading hidden bit of the fractional part is 0, then the floating-point number is called a **Subnormal number.**

That is in subnormal numbers, the constraint of leading bit to be a 1 is relaxed. They are represented with less precision than normal numbers by trading precision for smaller size.

They are represented as:

(–1)s × 0.f × 2–126 (all 0s for the exponent)

where f has at least one 1.

It is –126 and not -127 because a bias +1 so that the gap between the largest subnormal number and the smallest normalized number is smaller

The largest subnormal number 0.999999988×2–126 is close to the smallest normalized number 2–126 .

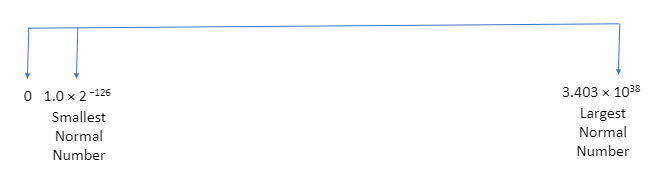
For a single precision floating point numbers:

**Normal number**:

Smallest number: ± 0000 0001 000 0000 0000 0000 0000 0000,

that is 1.0 × 2 −126 .

Largest number : ± 1111 1110 111 1111 1111 1111 1111 1111,

that is 1.99999988 × 2127 ≈ 3.403 × 1038 .

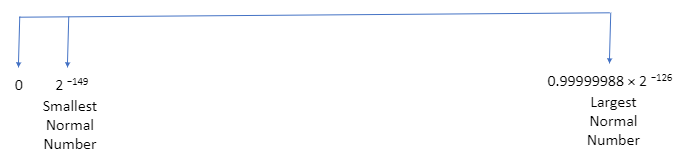
**Subnormal number:**

Smallest number: ± 0000 0000 000 0000 0000 0000 0000 0001,

that is 2 −126+( −23) = 2 −149 .

Largest number : ± 0000 0000 111 1111 1111 1111 1111 1111,

that is 0.99999988 × 2 −126 .



**Question 3.** IEEE 754vv defines standards for rounding floating points numbers to a represent able value. There are five methods defines by IEEE for this – Take time and understand what these five methods and explain it in your words using diagrams, illustrations of your own.

**Answer:**

The standard defines five rounding rules.

**Rounding to nearest**

1. [**Round to nearest, ties to even**](https://en.wikipedia.org/wiki/Rounding#Round_half_to_even)

This method of rounding involves rounding to the nearest value; if it is midway, it is rounded to the nearest value with an even least significant digit; this is the default for binary floating point and the recommended default for decimal.

1. [**Round to nearest, ties away from zero**](https://en.wikipedia.org/wiki/Rounding#Round_half_away_from_zero)

This method of rounding involves rounding to the nearest value; if it is midway, it is rounded to the nearest value above (for positive numbers) or below (for negative numbers); this is intended as an option for decimal floating point.

**Directed rounding**

1. **Round toward 0** (truncation)

This method of rounding involves directed rounding towards zero.

1. **Round toward +∞** (rounding up or ceiling)

This method of rounding involves directed rounding towards positive infinity.

1. **Round toward −∞** (rounding down or floor)**–** This method of rounding involves directed rounding towards negative infinity.

The different types of rounding can be illustrated by taking below examples:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Rounding** | **+ 13.5** | **-13.5** | **+52.5** | **-52.5** |
| [Round to nearest, ties to even](https://en.wikipedia.org/wiki/Rounding#Round_half_to_even) | + 14.0 | -14.0 | +52.0 | -52.0 |
| [Round to nearest, ties away from zero](https://en.wikipedia.org/wiki/Rounding#Round_half_away_from_zero) | + 14.0 | -14.0 | +53.0 | -53.0 |
| Round toward 0 | + 13.0 | -13.0 | +52.0 | -52.0 |
| Round toward +∞ | + 14.0 | -13.0 | +53.0 | -52.0 |
| Round toward −∞ | + 13.0 | -14.0 | +52.0 | -53.0 |