1. In Floating point representation we have three components:
   The Sign Bit
   Exponent
   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 that play the role in defining precision and how? Explain this with example in your own words.

Ans: The length of the fraction field also called as *significand*, *mantissa*, or *coefficient* determines the *precision* to which numbers can be represented. The radix point position is assumed always to be somewhere within the significand—often just after or just before the most significant digit, or to the right of the rightmost (least significant) digit. But general convention is that the radix point is set just after the most significant (leftmost) digit.

Using base-10 (decimal notation) as an example, the number 672853.1042, which has ten decimal digits of precision, is represented as the significand 6728531042 together with 5 as the exponent. To determine the actual value, a decimal point is placed after the first digit of the significand and the result is multiplied by 10^5 to give 6.728531042×10^5, or 672853.1042. In storing such a number, the base (10) need not be stored, since it will be the same for the entire range of supported numbers, and can thus be inferred.

Symbolically, this final value is:

$$\frac{s}{b^{p-1}} x\ b^e$$

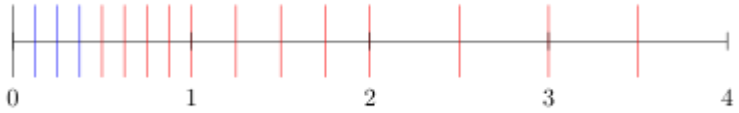
Where,'s' is the significand, 'p' is the precision(the number of digits in the significand), 'b' is the base (here 10) and 'e' is the exponent.

2. What is Normal and Subnormal Values as per IEEE 754 standard? Explain this with the help of number line.

Ans: In a normal floating-point value, there are no leading zeros in the significand; instead leading zeros are moved to the exponent. So 0.0123 would be written as 1.23×10^{-2}. Denormal (denormalized) numbers are numbers where this representation would result in an exponent that is below the minimum exponent (the exponent usually having a limited range). Such numbers are represented using leading zeros in the significand. In IEEE 754-2008, denormal numbers are renamed *subnormal numbers*, and are supported in both binary and decimal formats.

In a denormal number, since the exponent is the least that it can be, zero is the leading significand digit ($0.m_1m_2m_3...m_{p-2}m_{p-1}$), allowing the representation of numbers closer to zero than the smallest normal number.

An unaugmented floating point system would contain only normalized numbers (indicated in red). Allowing denormalized numbers (blue) extends the system's range.



3. IEEE 754 defines standards for rounding floating points numbers to a represent able value. What are the 5 methods?

Ans: The standard defines five rounding rules. The first two rules round to a nearest value; the others are called *directed roundings*.

Roundings to nearest

- **Round to nearest, ties to even** – rounds to the nearest value; if the number falls midway it is rounded to the nearest value with an even (zero) least significant bit; this is the default for binary floating-point and the recommended default for decimal.
- **Round to nearest, ties away from zero** – rounds to the nearest value; if the number falls 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 roundings

- **Round toward 0** – directed rounding towards zero (also known as *truncation*).
- **Round toward +∞** – directed rounding towards positive infinity (also known as *rounding up* or *ceiling*).
- **Round toward −∞** – directed rounding towards negative infinity (also known as *rounding down* or *floor*).

Example of rounding to integers using the IEEE 754 rules

Mode / Example Value	+11.5	+12.5	−11.5	−12.5
to nearest, ties to even	+12.0	+12.0	−12.0	−12.0
to nearest, ties away from zero	+12.0	+13.0	−12.0	−13.0
toward 0	+11.0	+12.0	−11.0	−12.0
toward +∞	+12.0	+13.0	−11.0	−12.0
toward −∞	+11.0	+12.0	−12.0	−13.0