In Floating point representation we have three components

1.The Sign Bit

2.Exponent

3.Fractional Part

Precession is one the prime attribute of any Floating point Representation,

1. Does any of the above three components play a role in the defining the Precession of the number ?

If so which are the component or Components which play the role in defining precession and how ? Explain this with example in your own words.

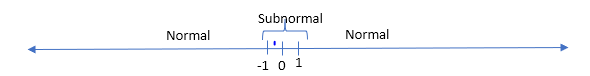
Precision is the refinement in a measurement. In floating point representation all the three components are important and contribute for the precision of a number. But if the decimal part of two numbers are same then the fractional part will decide the precision i.e more number of digits in fractional part more is the precision. So we can say that Fractional Part will play an important role.

Example: 526.6663 is more precise compared to 526.66

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

Normal values are those which are represented as (-1)^s\*(1.x)\*(2^(Exponent-bias)). For normal numbers the value of exponent should not be zero.

Subnormal values are those which are represented as (-1)^s\*(0.x)\*(2^(-bias+1)). For subnormal numbers the exponent should be zero, the fractional part should be non-zero.



3.IEEE 754 defines standards for rounding floating points numbers to a representable 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.

IEEE 754 defines 5 methods for rounding floating point numbers. The first two are nearest value rounding whereas the other three are directed rounding.

In nearest type of rounding the value output is chosen nearer of the possible values.

1.***Round to nearest, ties to even***: Round up to the nearest value , if the number is midway it is rounded to nearest value with lsb 0

2. ***Round to nearest, ties away from zero:*** Round up to the nearest value , if the number is midway it is rounded to nearest value with lsb 1

3. ***Round towards 0:*** directed towards 0

4. ***Round towards +infinity:*** directed towards +infinity

5. ***Round towards -infinity:*** directed towards -infinity