IEEE 754

Floating point representation we have three components

1.The Sign Bit
2. Exponent
3. Fractional Part

Sign	Exponent	Significand

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.

Ans:

Precision means how two or more experimental values are close to each other to the truth value in general.

IEEE 754 defines two representations for floating point numbers: Double Precision requires 64 bits and Single Precision requires 32 bits. Each precision has a range of itself. In short, more exponent bits implies greater range and more significand bits implies greater accuracy. So, specifically for better accuracy we need to have more number of bits in for significand/mantissa. With the increase in the number of bits, the range also automatically increases. Clearly, we are using less number of bits for Single Precision hence the accuracy is less.

For example,

We could represent a number in both the formats and convert back to the original. Here we can see the accuracy difference. But, here I have another observation from Matlab/Octave.

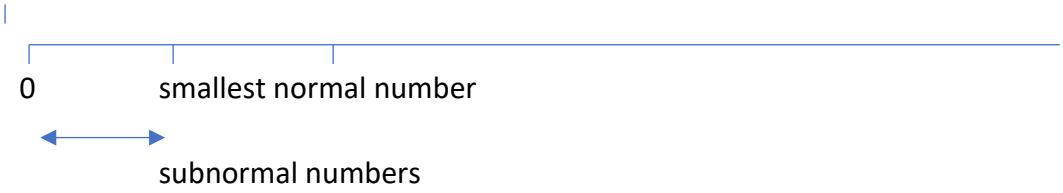
```
octave:7> Y=double ((0:499). ^2);
octave:8> sum(Y)
ans = 41541750
octave:9> clear
octave:10> Y=single ((0:499). ^2);
octave:11> sum(Y)
ans = 41541684
```

Here the range of single was limited. Hence the wrong answer. We can observe the difference in the results.

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

Ans:

For a normal number, exponent ranges from 1-254/2046 and significand can be anything. For a subnormal number, exponent is zero and significand is non-zero. The purpose of having subnormal numbers is to smooth the gap between the smallest normal number and zero.



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.

Method 1: Round toward zero or truncate

According to the availability of bits, make the results as close to zero as possible.

Method 2: Round toward negative infinity.

Method 3: Round toward positive infinity.

Method 4: Round to nearest

Case 1: rounds to the nearest value; if the number falls midway it is rounded to the nearest value with an even (zero) LSB

Case 2: 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);

Method	Before rounding	After rounding
Round to nearest: Case 1	22.5	23
Round to nearest: Case 2	22.5 -22.5	22 -23
Round up, or round toward plus infinity	1.25	1.3
Round down, or round toward minus infinity	1.25	1.2
Round toward zero, or chop, or truncate	0.6663	0.663/0.66/0.6