FLOATING POINT NUMBERS

- In some numbers, which have a fractional part, the position of the decimal point is not fixed as the number of bits before (or after) the decimal point may vary.
- **Eg: 0010.01001, 0.0001101, -1001001.01** etc.
- As shown above, the position of the decimal point is not fixed, instead it **"floats"** in the number.

  For doubts contact Bharat Sir on 98204 08217
- Such numbers are called Floating Point Numbers.
- Floating Point Numbers are stored in a "Normalized" form.

NORMALIZATION OF A FLOATING POINT NUMBER

- Normalization is the process of shifting the point, left or right, so that there is only one non-zero digit to the left of the point. #Please refer Bharat Sir's Lecture Notes for this ...

Floating Point Number		Normalized Number
01010.01	→	$(-1)^0 \times 1.\mathbf{01001} \times 2^3$
11111.01	→	$(-1)^0 \times 1.\mathbf{111101} \times 2^4$
0.00101	→	$(-1)^0 \times 1.\mathbf{01} \times 2^{-3}$
-10.01	→	$(-1)^1 \times 1.\mathbf{001} \times 2^1$

- As seen above a Normalized Form of a number is:

$$\boxed{(-1)^S \times 1.M \times 2^E}$$

Where: S = Sign, M = Mantissa and E = Exponent.

- As Normalized numbers are of the 1.M format, the "**1**" is not actually stored, it is instead **assumed**. This saves the storage space by 1 bit for each number.

- Also the Exponent is stored in the biased form by adding an appropriate bias value to it so that -ve exponents can be easily represented.

Advantages of Normalization.

1. Storing all numbers in a standard for makes **calculations easier** and **faster**.
2. By **not storing** the **1** (of 1.M format) for a number, considerable **storage space** is **saved**.
3. The **exponent** is **biased** so there is **no need** for **storing** its **sign bit** (as the biased exponent cannot be -ve).

SHORT REAL FORMAT / SINGLE PRECISION FORMAT / IEEE 754: 32 BIT FORMAT

S	Biased Exponent	Mantissa
(1)	**(8)** **Bias value = 127**	**(23 bits)**

- **32 bits** are used to store the **number**.
- **23 bits** are used for the **Mantissa**.
- **8 bits** are used for the Biased **Exponent**.
- **1 bit** used for the **Sign** of the number.
- The **Bias** value is (**127**)$_{10}$.
- The range is **$\pm 1 \times 10^{-38}$ to $\pm 3 \times 10^{38}$** approximately.
- It is called as the **Single Precision Format** for Floating-Point Numbers.

LONG REAL FORMAT / DOUBLE PRECISION FORMAT / IEEE 754: 64 BIT FORMAT

S	Biased Exponent	Mantissa
(1)	**(11)** **Bias value = 1023**	**(52 bits)**

- **64 bits** are used to store the **number**.
- **52 bits** are used for the **Mantissa**.
- **11 bits** are used for the Biased **Exponent**.
- **1 bit** used for the **Sign** of the number.
- The **Bias** value is (**1023**)$_{10}$.
- The range is **$\pm 10^{-308}$ to $\pm 10^{308}$** approximately.
- It is called as the **Double Precision Format** for Floating-Point Numbers.

Numericals on Floating Point Number Representation

1) Convert 2A3BH into Short Real and Temp Real formats {Exam question}

Short real:

Converting the number into binary we get:
0010 1010 0011 1011

Normalizing the number we get:
$\boxed{(-1)^{0} \times 1.\mathbf{0101000111011} \times 2^{13}}$
Here S = 0; M = 0101000111011; True Exponent = 13.

Bias value for Short Real format is 127:
Biased Exponent (BE) = True Exponent + Bias
= 13 + 127
= 140.

Converting the Biased exponent into binary we get:
Biased Exponent (BE) = (1000 1100)

Representing in the required format we get:

0	**10001100**	**01010001110110 0…**
S	Biased Exp	Mantissa
(1)	(8)	(23)

Converting the number into hexadecimal form we get:
4628EC00H … 32 bits.

Temp real:
Bias value for Temp Real format is 16383:
Biased Exponent (BE) = True Exponent + Bias
= 13 + 16383
= 16396.

Converting the Biased exponent into binary we get:
Biased Exponent (BE) = (100 0000 0000 1100)

Representing in the required format we get:

0	100000000001100	**1**0101000111011 00…
S	Biased Exp	Mantissa
(1)	(15)	(64)

Converting the number into hexadecimal form we get:
400C A8EC 0000 0000 0000H … 80 bits.

2) Convert $(12.125)_d$ into Temp Real format {Exam question}

<u>Temp real:</u>

Converting the number into binary we get:
1100.001 For doubts contact Bharat Sir on 98204 08217

Normalizing the number we get:
$\boxed{(-1)^{0} \times 1.\mathbf{100001} \times 2^{3}}$
Here S = 0; M = 100001; True Exponent = 3.

Bias value for Temp Real format is 16383:
Biased Exponent (BE) = True Exponent + Bias
= 3 + 16383
= 16386.

Converting the Biased exponent into binary we get:
Biased Exponent (BE) = (100 0000 0000 0010)

Representing in the required format we get:

0	100000000000010	**1**10000100000…
S	Biased Exp	Mantissa
(1)	(8)	(23)

Converting the number into hexadecimal form we get:
4002 C200 0000 0000 0000H … 80 bits.