1) The three components are
   1) sign bit S
   2) Exponent (E)
   3) fractional part (M)

Number is $f = (-1)^S M \times 2^E$

Two kinds of encoding done
   i) single precission (32 bits, E 8 bits, M 23 bits)
   ii) double precission (64 bits, E 11 bits, M 52 bits)
   1 sign bit for each type

⇒ The precission depends on fractional part.
⇒ The number of significant digits depends on the number of bits in M
when M = 24 bits
we have x significant digits
where $2^{24} = 10^x \Rightarrow x = 7.2\ldots$
$\Rightarrow x = 7$ ⇒ upto seven significant decimal places we can represent in binary floating point representation.

⇒ The range of number depends on the number of bits in exponent that is $2^8 - 1 = 1$ to 254

⇒ The exponent is encoded as a biased value
$E = exp + bias$ where bias = $127 = (2^{8-1} - 1)$ for sigle precission
bias = $1023 (2^{11-1} - 1)$ for double precission

⇒ Example

As the precission depends on the fraction part(mantissa)

for single precission

23 bits are available so we can precisely represent upto 23 bits after decimal.

for example when the decimal number is

2.3

it's representation is.

10.0100110011001100110011

for double precission

52 bits are available so we can precisely represent upto 52 bits after decimal

for example when the decimal number is

2.3

it's representation is

10.0100110011001100110011001100110011001100110011

⇒ So in normal word saying when the numbers decimal part lies in between 0 to $\frac{1}{2^{23}}$ it can be accurately representable in single precission type of representation.

decimal part

⇒ when the number's, lies between 0 to $\frac{1}{2^{52}}$ it can be represented accurately using double precission representaion.

⇒ when number is

$1.x$

if $0 < x < \frac{1}{2^{23}}$

and it's represented in sigle precission, it has 100% accuracy

⇒ when number is

$1.x$

if $0 < x < \frac{1}{2^{52}}$

and it's represented in double precission, it has 100% accuracy.

2) When the content of the exponent(e) = 0 & significant(m) is ≠0 then the sub normal number is

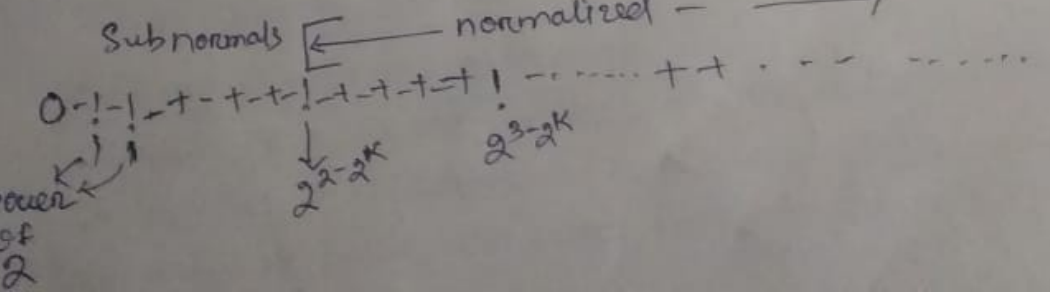
$(-1)^s \times 0.m \times 2^{-126}$

	normalized number	subnormal
smallest	2^{-126}	$2^{-149} \simeq 0$
Largest	3.4×10^{38}	$0.99999988 \times 2^{-126}$ $\simeq$ smallest normalized number

⇒ So there are total $[0.0, 2^{-126}]$ that means 2^{23} numbers with in the range.

⇒ The smallest difference between 2 normalized number is 2^{-149} which is equal to difference between any two consecutive subnormal numbers.

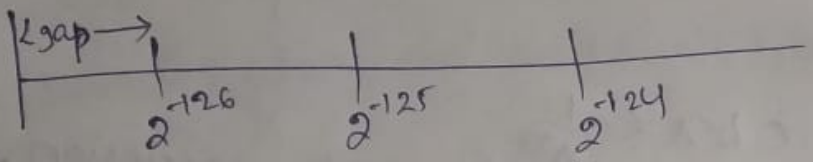
⇒ Meanwhile the largest difference b/w 2 consecutive numbers is 2^{104}



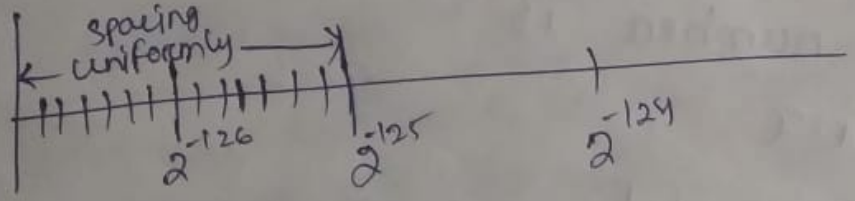
⇒ Subnormals extend range of magnitudes representable but have less precision than normalised numbers.

⇒ for a 32 bit precision type the number line distinguishes between normal & subnormal values in the figure below

i) without subnormal



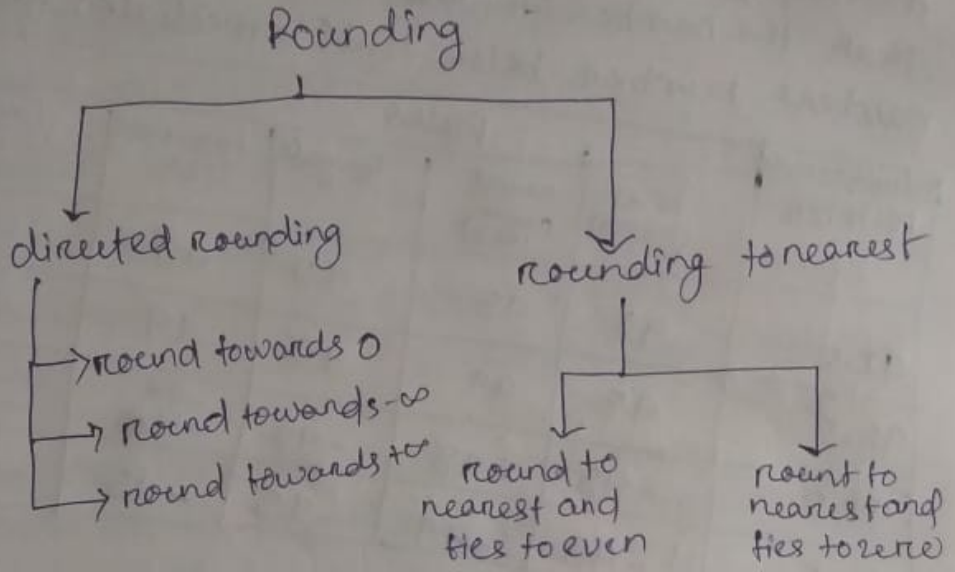
ii) with subnormal



Q3)

page-6

=> There are basically five rounding rules. And they are catagorized into two subsets.



a) Directed rounding (truncation)

i) Round towards 0

Directed rounding towards zero

ii) Round towards ∞ (ceiling)

Directed rounding towards +∞

iii) Round towards -∞ (floor)

directed rounding towards -∞

b) Rounding to nearest

i) Round to nearest, ties to even

Rounds to nearest value if the value falls in middle, it is rounded to the nearest number i.e it must have an even least significant digit.

ii) Round to nearest, ties away from zero

Rounds to the nearest value, if the number falls midway then for positive number number greater than the number (~~integer~~ nearest) taken and for negative numbers number below that (immediate) taken.

Values to be rounded	Rules				
number	nearest (to even)	nearest (away from 0)	towards 0	towards ∞	towards $-\infty$
97.5	98	98	97	98	97
98.5	98	99	98	99	98
-97.5	-98	-98	-97	-97	-98
-98.5	-98	-99	-98	-98	-99