Today's Content :-

a) How -ve no's are stored

b) Significance of MSB

c) datatype ranges.

8 bit number

→ sign bit → Concep of storing -ve no is wrong,

	7	6	5	4	3	2	1	0
10 :	0	0	0	0	1	0	1	0
-10 :	1	0	0	0	1	0	1	0

4 :	0	0	0	0	0	1	0	0
-4 :	1	0	0	0	0	1	0	0
10 :	0	0	0	0	1	0	1	0
-14	1	0	0	0	1	1	1	0

1 more problem :-

0 0 0 0 0 0 0 0 → 0

1 0 0 0 0 0 0 0 → -0

2's complement :-

$a = -a$ (in binary) $=$ 2's complement of a

$\Rightarrow$ 1's comp a + 1

$\Rightarrow$

$0 \rightleftarrows 1$

$-10 = $ 1's of $10 + 1$

$= \sim 10 + 1$

8 bit no's.

10 :	0	0	0	0	1	0	1	0
~10 :	1	1	1	1	0	1	0	1
+1	0	0	0	0	0	0	0	1
−10 ⇒	1	1	1	1	0	1	1	0

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	0	1	1	0

$2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1$

$\Rightarrow$ **246**.

$-2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1$

$\Rightarrow$ $\underline{-10}$.

// 4 bit $\underline{\text{No}}$.

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ \underline{\quad} & \underline{\quad} & \underline{\quad} & \underline{\quad} \\ -2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

8 bit $\underline{\text{No}}$.

$$\begin{array}{cccccccc} -2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \underline{\quad} & \underline{\quad} & \underline{\quad} & \underline{\quad} & \underline{\quad} & \underline{\quad} & \underline{\quad} & \underline{\quad} \end{array}$$

$\downarrow$

128

$\downarrow$

LMB $\Rightarrow$ Most Significant bit.

$\downarrow$

MSB base value is negative.

8 bit no

	-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
10 :	0	0	0	0	1	0	1	0
−4 :	1	1	1	1	1	1	0	0
6 1	0	0	0	0	0	1	1	0

−4

0 0 0 0 0 1 0 0 ← +4
1 1 1 1 1 0 1 1 ← ~4
+ 1
1 1 1 1 1 1 0 0 ← −4

$$\begin{array}{r} 11 \\ -\ 7 \\ \hline 4 \end{array}$$

} → 8 bit no's

	-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
11 →	0	0	0	0	1	0	1	1
−7 →	1	1	1	1	1	0	0	1
1	0	0	0	0	0	1	0	0

Convert Binary to decimal :-

4 bit No

-2^3	2^2	2^1	2^0		
1	0	1	1	→	-5
1	0	1	0	→	-6
0	0	1	1	→	3
1	0	0	0	→	-8
1	1	1	1	→	-1

8 bit Numbers :-

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		Decimal
0	0	0	1	0	1	0	1	→	21
1	0	0	1	0	1	0	1	→	-107
1	0	0	1	0	0	0	1	→	-111
0	0	0	1	0	0	0	1	→	17

N bit Numbers :-

N-1	...	...	3	2	1	0
-2^{n-1}	...	...	2^3	2^2	2^1	2^0

↓

Max Neg → 1 0 0 0 0 0 — - - → -2^{n-1}

Max Pos → 0 1 1 1 1 1 — . . → 1 →

will MSB Base value be always -ve ?

unsigned

Declare unsigned 4 bit no'

$\underline{2^3}$ $\underline{2^2}$ $\underline{2^1}$ $\underline{2^0}$

Declare unsigned 8 bit no'

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

Signed

Declare signed 4 bit no'

$\underline{-2^3}$ $\underline{2^2}$ $\underline{2^1}$ $\underline{2^0}$

Declare signed 8 bit.

-2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

How will system know signed / unsigned.

C / C++ / C#

unsigned

unsigned int x
unsigned long x
unsigned short x

Base value of MSB is +ve

→ we can't store -ve numbers.

Default
int x
long x
short x

By default they are signed : Base value of MSB bit is -ve.

In Java : There is no such that as unsigned, There are only signed datatypes.

Break : 10:05 → 10:15 pm .

Ranges in Signed Datatypes.

2 bit Signed no.

-2^1	2^0	Decimal
0	0	$\rightarrow$ 0
0	1	$\rightarrow$ 1
1	0	$\rightarrow$ -2
1	1	$\rightarrow$ -1

$\rightarrow$ [-2 1]

min max

Can we store -5 or 10 in a 2 bit no? [No]

3 bit Signed no.

-2^2	2^1	2^0	Decimal
0	0	0	= 0
0	0	1	= 1
0	1	0	= 2
0	1	1	= 3
1	0	0	= -4
1	0	1	= -3
1	1	0	= -2
1	1	1	= -1

Min Max

[-4 , 3]

N bit signed no.

	Min	Max
2 bits	-2	1
3 bits	-4	3

n-1 ____ . . . ____ 3 2 1 0

Min ⇒ 1 0 0 0 0 0 . . . . → -2^{n-1}

Max ⇒ 0 1 1 1 1 1 . . . 1 1 1 →

2^{n-2} $\quad 2^2 \quad 2^1 \quad 2^0$

⇒ $2^0 + 2^1 + 2^2 + \dots 2^{n-2}$

⇒ $a = 1$, $r = 2$, Terms $= n-1$.

$$S = \frac{a(r^n - 1)}{r-1} \Rightarrow \frac{1(2^{n-1} - 1)}{2-1}$$

$$\Rightarrow 2^{n-1} - 1$$

Range of n bit no :- $[-2^{n-1}$ to $2^{n-1} - 1]$

	Bytes	bits	min	max
byte / char	1	8	[-128, 127]	
short int	2	16	$\Rightarrow \{-2^{15}$ to $2^{15}-1\}$	
int	4	32	$\Rightarrow \{-2^{31}, 2^{31}-1\}$	
long int	8	64	$\Rightarrow \{-2^{63}, 2^{63}-1\}$	

byte / char → can we store 130, 240, 10^5 → overflow.

short int: [-32768, 32767]

int: close appr → $\{-2*10^9, 2*10^9\}$

long int: close appr → $\{-8*10^{18}, 8*10^{18}\}$

$2^{10} = 1024 \simeq 1000$

$(2^{10})^3 \approx (1000)^3$

$2^{30} \approx 10^9 \xrightarrow{*2} 2^{31} \approx 2*10^9$

$2^{60} \approx 10^{18} \xrightarrow{*2^3} 2^{63} \approx 8*10^{18}$

Importance of Constraints :-

Q1) Given an array, calculate sum of it.

Constraints :-

$1 <= N <= 10^5$

$1 <= arr[i] <= 10^6$

$1 <= sum <= 10^{11}$

↓

Can we store in int * int

we need to use long.

~~int~~ long sum(int arr[]){

    int n = arr.length

    ~~int~~ long sum = 0

    for (i=0; i<n; i++)

    { sum = sum + arr[i]

    }

    return sum;

}

Constraints → TLE → TLE

Constraints → Datatypes → Error

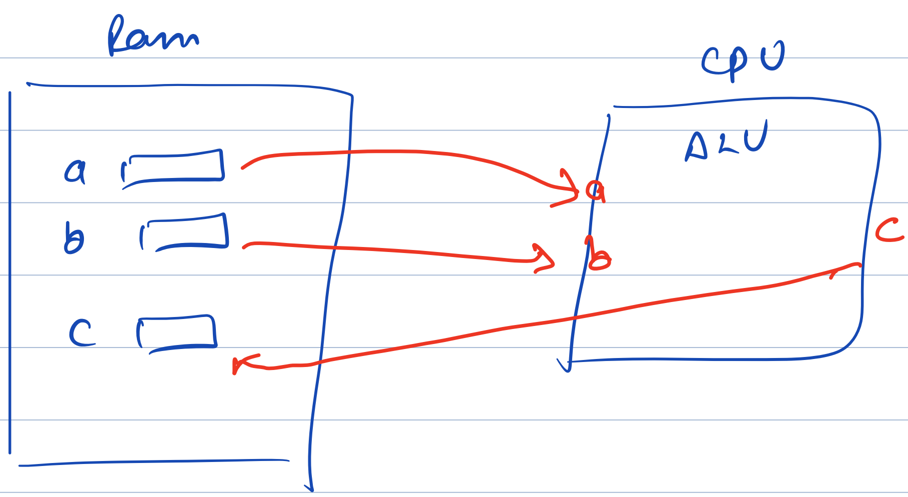
↓

V.V. Important.

Ques Given 2 number a & b, return there prod.

$1 <= a, b <= 10^6$

```
pred (int a, int b) {
    int c = a*b
    long c = (a*b);
    long c = long(a*b)
    long c = (long) a * b
}
    long a1 = a, b1 = b
    return a1*b1
}
```



Take care :-

→ when we multiply int * int.

→ when we multiply long * long.

A & (Long.MAX_VALUE << B);

(0 1 1 1 1 1) << B

Ques, given N & i, unset ith bit in N.

N = 23
i = 2,

$$\begin{array}{ccccc} 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ & & 0 & & \\ \hline 1 & 0 & 0 & 1 & 1 \end{array}$$

→ 1 0 1 1 1
& 1 1 0 1 1 ←

1 0 0 1 1

→ 1 1 0 1 1

~ (0 0 1 0 0) → 1 1 0 1 1

no :- 1 1 1 1 0 0 1 1
mask → 1 1 1 0 1 1 1 1

1 1 1 0 0 0 1 1

→ 1 1 1 0 1 1 1 1
→ (0 0 0 1 0 0 0 0)

Ques) Given x, y set x continuous bits & y unset bits.

$x = 3,\ y = 2$:- $\begin{array}{ccccc} 4 & 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array} \rightarrow 28$

↳ 111 << 2

$x = 5,\ y = 3$, 11111 000

↳ 11111 << 3

11111 << y

→ 111 → 7

→ 1000 8

$(1 << 3) - 1$

$\overbrace{1111}^{x} \rightarrow 15$

→ 10000 → 16

$(1 << 4)$

$((1 << x) - 1) << y$

	7	6	5	4	3	2	1	0
→	1	1	0	0	0	1	1	1 → **8 bit**
	l							r