Today's content

a) How to store -ve numbers
b) Significance of MSB
c) Datatypes Ranges

8 bit number.

sign bit

	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

This concept of storing -ve number is wrong.

Issue with above method

4	0 0 0 0 0 1 0 0	
-4	1 0 0 0 0 1 0 0	
	1 0 0 0 1 0 0 0	→ -8

Negative of a (-a) = 2's complement of a.

= 1's complement of a + 1

1's complement: inverse
0 → 1
1 → 0

$$
\begin{array}{rcccccccc}
 & -2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
10 = & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

$$
\begin{array}{rcccccccc}
1's\ com = s\,10 = & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
1 = & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
-10 = & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0
\end{array}
$$

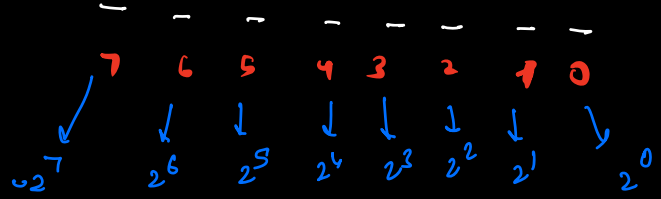
$$-2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1$$

$$= -128 + 118$$

$$= -10$$

int x = ~10

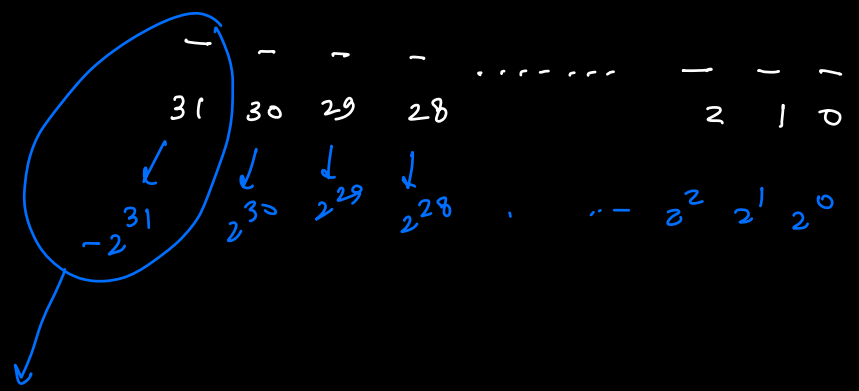
8 bit number.



1 0 0 0 0 0 0 0 → $-2^7 = -128$

int

32 bit number



LMB → most significant Bit (MSB)

Basic value will be negative.

Convert Binary to decimal Below value

4 bit number

-2^3 2^2 2^1 2^0

1 0 1 1

$= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 2^3 + 2^1 = 8 + 2 + 1 = 11$

$= 1\times-2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= -2^3 + 2^1 + 2^0$

$= -8 + 2 + 1$

$= -5.$

$-2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$1\ 0\ 1\ 0 = -2^3 + 2^1 = -8+2 = -6$

$0\ 0\ 1\ 1 = 2^1 + 2^0 = 2+1 = 3$

$1\ 0\ 0\ 0 = -2^3 = -8$

$1\ 1\ 1\ 1 = -2^3 + 2^2 + 2^1 + 2^0 = -8+4+2+1$
$= -1$

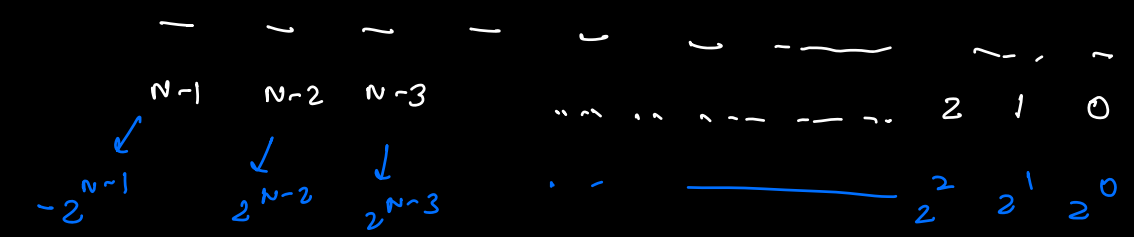
8 bit numbers

$-2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

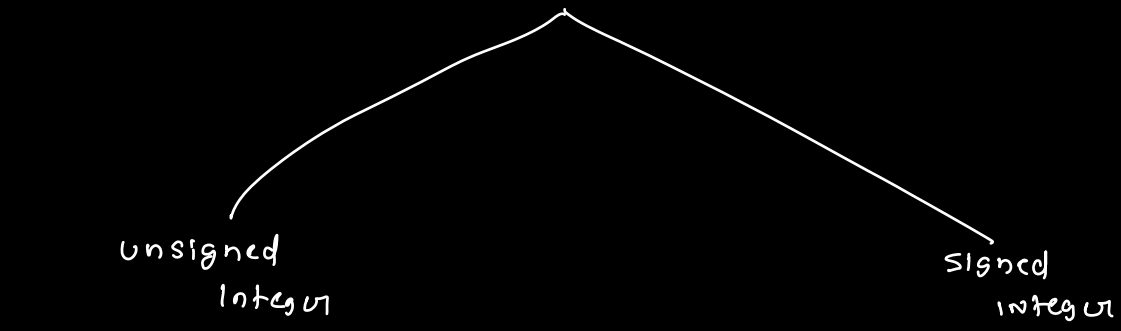
$0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 = 2^2 + 2^0 = 5$

$1\ 0\ 0\ 1\ 0\ 1\ 0\ 1 = -2^7 + 2^4 + 2^2 + 2^0$
$= -128+16+4+1$
$= -107$

N bit numbers



Will MSB Base value be always -ve ?



unsigned int x = 10

int x = 10

unsigned 4 bit number

2^3 2^2 2^1 2^0

signed 4 bit number

-2^3 2^2 2^1 2^0

C++ / C C#

unsigned int ✗ Java

unsigned int x

unsigned long x

int x

long x

Ranges of signed Datatypes.

2 bit signed number

$-2^1 \quad 2^0$

0 0 = 0

0 1 = 1

1 0 = -2

1 1 = -1

[-2 , 1]

min number — max number

3 bit number

$-2^2 \quad 2^1 \quad 2^0$

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 = -4+2 = -2

1 1 1 = -4+2+1 = -1

[-4, 3]

000

[0, 7]

8 bit

$\overline{-2^7}$ $\overline{2^6}$ $\overline{2^5}$ $\overline{2^4}$ $\overline{2^3}$ $\overline{2^2}$ $\overline{2^1}$ $\overline{2^0}$

$[\ -128 \quad , \quad 127\ ]$

-128: min value, 127: max value

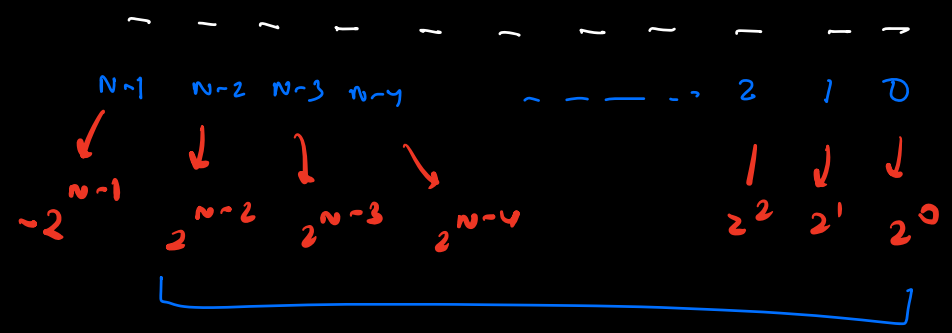
0 1 1 1 1 1 1 1

$2^6+2^5+2^4+2^3+2^2+2^1+2^0 = 127$

1 0 0 0 0 0 0 0

$-2^7 = -128$

2 bits	$[-2, 1]$
3 bits	$[-4, 3]$
4 bits	$[-8, 7]$
8 bits	$[-128, 127]$
⋮	
N bit number	$[-2^{N-1}, \ 2^{N-1}-1\ ]$



$$2^{N-2} + 2^{N-3} + 2^{N-4} + \cdots 2^2 + 2^1 + 2^0$$

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

$$= 2^{N-1} - 1$$

N bit number $\quad [-2^{N-1}, \; 2^{N-1} - 1]$

int

32 bits

$$[-2^{31}, \; 2^{31} - 1]$$

INT_MIN

INT_MAX

$$[-2147483648, \; 2147483647]$$

Approx $\quad [-2 \times 10^9, \; 2 \times 10^9]$

$2^{10} = 1024 \approx 10^3$

$2^{30} = 10^3 \times 10^3 \times 10^3 = 10^9$

$2^{31} = 2 \times 10^9$

long type

64 bits

$[-2^{63}, 2^{63}-1]$

Approx: $[-8 \times 10^{18}, 8 \times 10^{18}]$

$2^{10} = 10^3$

$2^{60} = 10^{18}$

$2^{63} = 2^3 \cdot 2^{60}$

$= 8 \times 10^{18}$

Importance of Constraints.

Q. Given array, return array sum

$1 \le N \le 10^5$

$1 \le arr[i] \le 10^9$

10^5 10^5 10^5 10^5 ....

← 10^5 times →

$10^9 * 10^5 = 10^{14}$

long

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

{

long

~~int~~ sum = 0

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

{ sum = sum + arr[i] ;

return sum

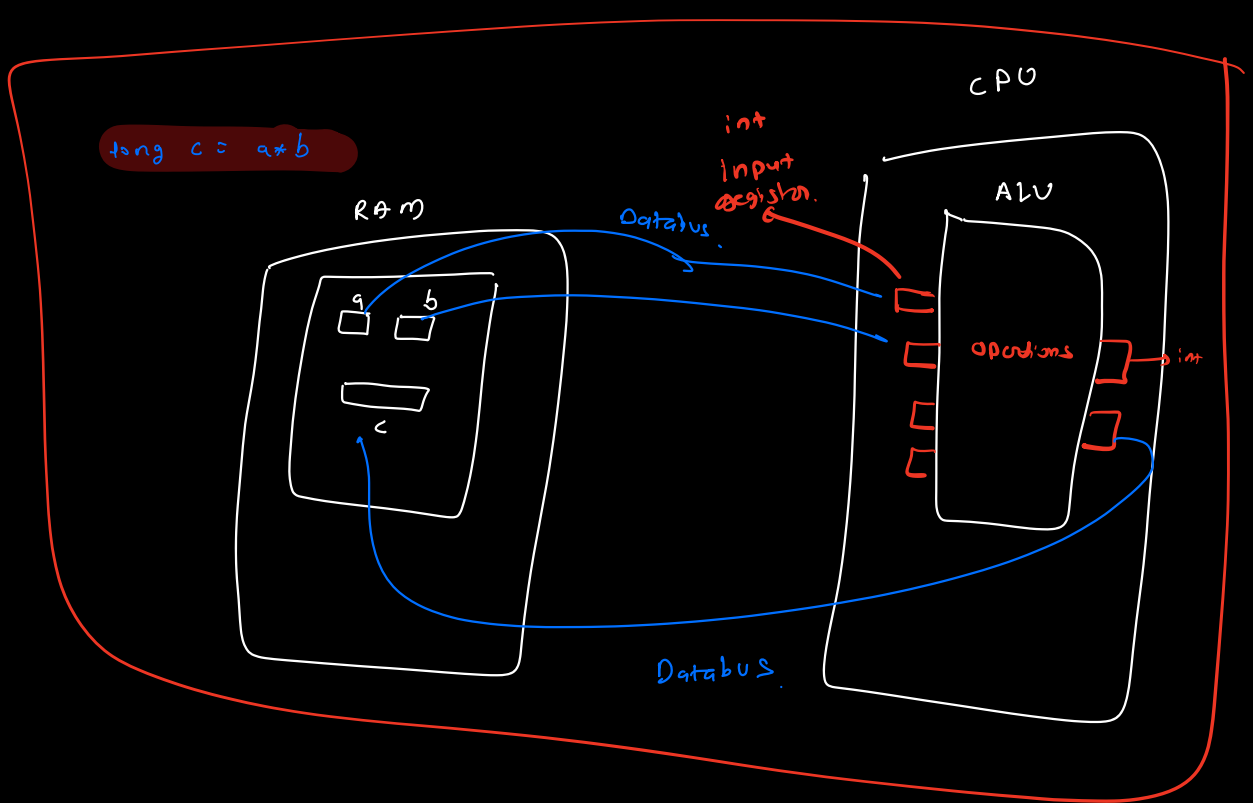
}

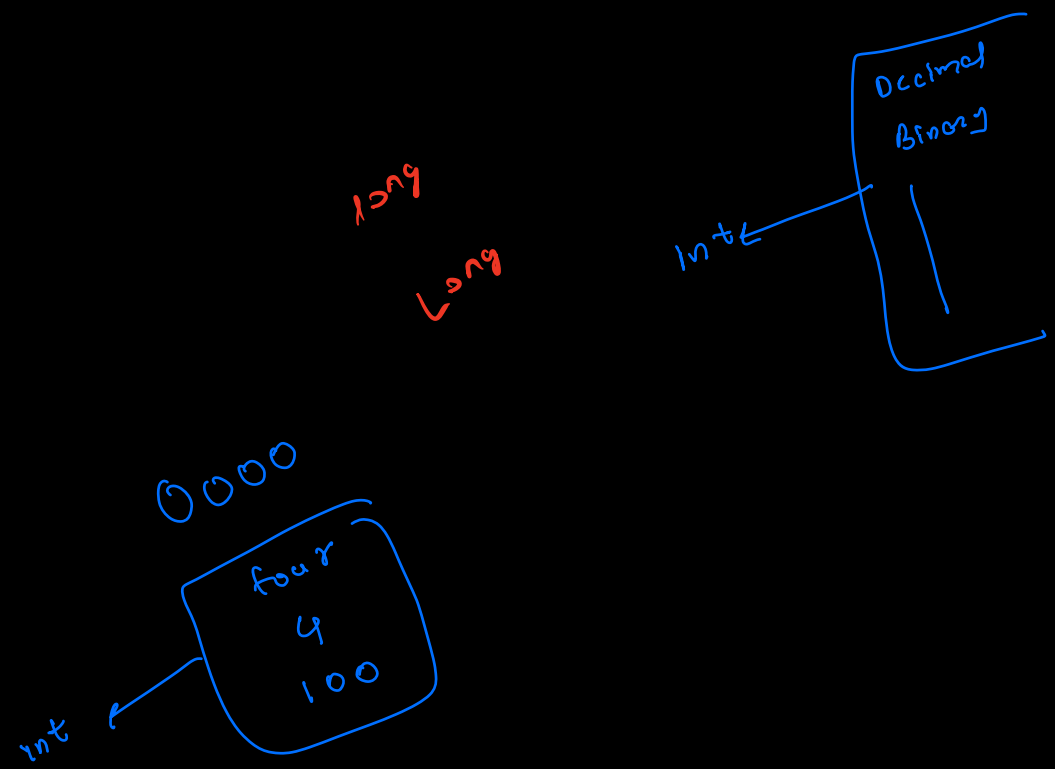
Q. Given 2 numbers, return a * b.

$1 \le a, b \le 10^6$

```
long prod(int a, int b)
{
    int c = a*b
    long c = a*b
    long c = long(a*b)

    long c = (long) a * (long) b
    long c = (long) a * b.
    long c = (long) (a*b)
    return c;
}
```




long
Long
Decimal
Binary
int
0000
four
4
100
int