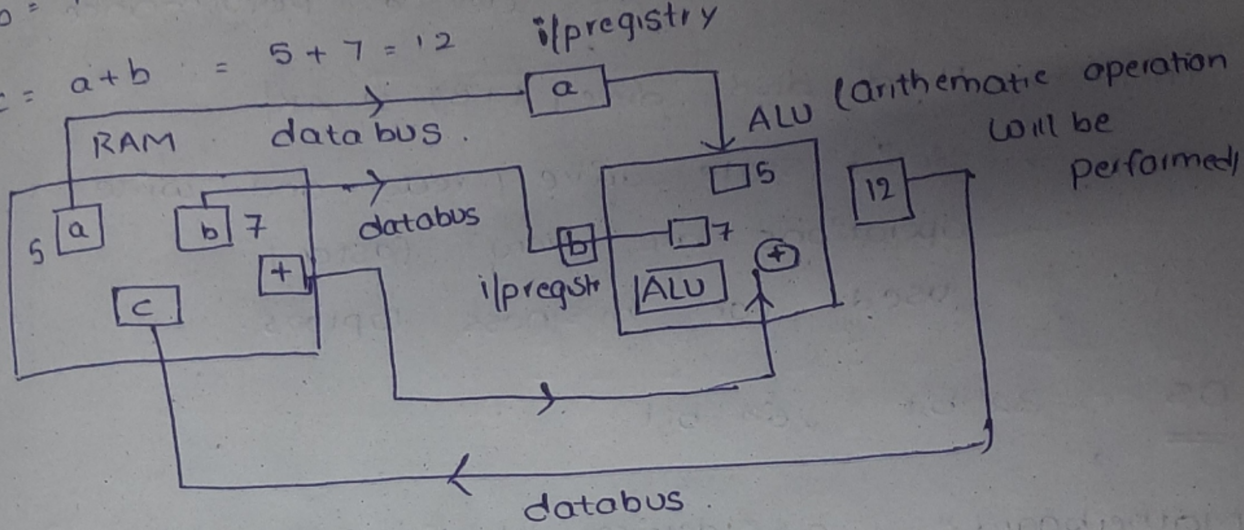
23/09/2023

Highlevel → machine level
(0's & 1's)

a = 5

b = 7

c = a+b = 5 + 7 = 12



Desktop :- specifications

	A	B
$1K = 10^3$	1×10^9 Hz	2×10^9 Hz
$1M = 10^6$	1×10^9 cc/sec	2×10^9 cc/sec
$1G = 10^9$		

1 clock cycle = 1 instruction.

A can send 10^9 Instructions/sec and b can send 2×10^9 instructions/sec.

RAM → Size/speed (we send data via buses we check how fast it is responding)

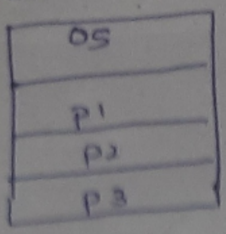
CPU → Speed
→ Cores (multicores helps run tasks parallelly)
→ Cache (L1, L2, L3 (storage and the quick retrieval of frequently used data)
→

OS

Motherboard

GPU : (VRAM)

Context switching :-



→ If we want P4 (a new program) to run we have to use the context switching but this might reduce the device performance.

HDD → Hard disk drive (1TB) ≅ 4K/-

SSD → Solid state drive (read and write operations are very faster). used for business laptops.

OS :- 32 bit , 64 bit.

Motherboard :- a device which conects the hard ware components with wires and make the hard ware software interaction.

Bit to Byte Conversion :-

1 Byte = 8 Bit

1 KB = 1024 Bytes = 2^{10} Bytes

1 MB = 1024 KB = 2^{20} Bytes = 2^{10} KB

1 GB = 1024 MB = 2^{10} MB = 2^{20} KB = 2^{30} Byte

~~4 GB RAM uses~~

8GB | 4GB which has more performance for 32 bit sys

Sol: 4GB = 4×2^{30} = $2^2 \times 2^{30}$ = 2^{32}

for 2 Bits : wehave 4 Combinations.

2 Bits : 0 0
0 1
1 0
1 1

For 3 bits we have 8 combinations.

3 Bits: 000
001
010
011
100
101
110
111

If we derive the formula:-

For n bit system it has 2^n combinations.

N Bitsystem can generate 2^n Bytes.

By Reverse Engineering:.

4GB = 2^{32} Bytes.

32 bit system generates 2^{32} Bytes.

∴ so 4 GB is Compatible with 32 bit System.

8 GB = $2^3 \times 2^{30}$

= 2^{33} (33 bit system is not available).

The space will be there to operate on more tha 32 bits, but in the question it was asked for only 32 bit system.

So both are feasible. and the performance is same.

4GB	8 GB
32 bit	32 bit

→ both functions in a same way.

Convert 24 into Binary :-

$$\begin{array}{r|l} 2 & 24 \\ \hline 2 & 12 - 0 \\ \hline 2 & 6 - 0 \\ \hline 2 & 3 - 0 \\ \hline & 1 - 1 \end{array}$$

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

Method : 2

24 → 0 0 0 1 1 0 0 0

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

$2^5 > 24$, $2^4 < 24$ (fill 1 at 2^4 th place).

$24 - 2^4 = 24 - 16$

$= 8$.

$2^3 = 8$ so fill 1 at 2^3 position

$= 8 - 8$

$= 0$.

19 → 0 0 0 1 0 0 1 1

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

$2^5 < 19$

$2^4 > 19$

$19 - 16 = 3$. — 1

$2^3 > 3$

$2^2 > 3$

$2 < 3$

$3 - 2 = 1$ — 2

$2^0 = 1$. $1 - 1 = 0$ — 3

Signed Magnitude representation :-

The MSB bit will represent the sign.

- (negative) → 1 in MSB bit

+ (positive) → 0 in MSB bit.

24 → 0 0 0 1 1 0 0 0

-19 → 1 0 0 1 0 0 1 1

→ 1 0 1 0 1 0 1 1 → 128

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

1 + 2 + 8 + 32 = 43

represents ⊖ sign

So result will be -43.

but 24 - 19 ≠ -43. → ambiguity ①

0 → 0 0 0 0 0 0 0 0

-0 → 1 0 0 0 0 0 0 0

" -0" does not exists. so will create an ambiguity. → ambiguity 2

To overcome this issue we are going for the 2's compliment.

2's compliment = 1's compliment + 1.

19 → 0 0 0 1 0 0 1 1

1's compliment of 19 → 1 1 1 0 1 1 0 0

1 → 0 0 0 0 0 0 0 1

-19. → 1 1 1 0 1 1 0 1

Mathematical operation on 24

$$
\begin{array}{rl}
 & \ 1 \ 1 \ 1 \ 1 \\
24 \rightarrow & 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0 \\
-19 \rightarrow & 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1 \\
\hline
\rightarrow 1 & 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 \rightarrow
\end{array}
$$

discarded

$00000101 \rightarrow 5$

$24 - 19 = 5$

Understanding the -19 Binary representation and find the value of MSB Bit.

$-19 = 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1$

$-19 = x + 2^6 + 2^5 + 2^3 + 2^2 + 1$

$-19 = x + 109$

$x = -(109 + 19)$

$x = -128 = -2^7$

~~x = -127~~.

Formula derivation:-

for n bits the MSB represents -2^{N-1}

mathematical operation on $-24 + 19$

$$
\begin{array}{rl}
24 \rightarrow & 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0 \\
(-24)\text{1's compliment of } 24 \rightarrow & 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1 \\
19 \rightarrow & 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1 \\
\hline
 & 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0 \\
 & 2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0
\end{array}
$$

$-128 + 64 + 32 + 16 + 8 + 2 =$

1st Compliment of 24

	1 1 1 0 0 1 1 1
Comp 24 =	
+1 =	0 0 0 0 0 0 0 1
-24 =	1 1 1 0 1 0 0 0
+19 =	0 0 0 1 0 0 1 1
	1 1 1 1 1 0 1 1

−128 + 64 + 32 + 8 + 4 + 2 + 1

= −8 + 3

= −5

−24 + 19 = −5

Datatypes :-

int
- → short (2 Bytes)
- → int (4 Bytes)
- → long. (4 or 8 bytes) → depends on the architecture. (4 − 32 bit, 8 − 64 bit)
- → long long [8 bytes].
  - ↓ not there in Java.

float → 4 bytes.

char → 1 byte / 2 bytes (Java).

Bool → 1 byte.

double → 8 bytes

16 bytes.

Deriving the ranges for the datatypes :-

3 bit unsigned ranges from [0 7]

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

3 bit Signed ranges from [−4 3]

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

Bits	usigned		Signed	
	Min	Max	Min	Max
2 Bits	0	3	-2	1
3 Bits	0	7	-4	3
4 Bits	0	15	-8	7
8 Bits	0	255	-128	127
N Bits	0	$2^n - 1$	-2^{n-1}	$+2^{n-1} - 1$

By default int is signed.
If we want only +ve integers we have to mention it explicity as unsigned.

Integer Size is 4 Bytes = 4x8 = 32 bits.

so range becomes:

$[0 \quad 2^{32} - 1]$ → for unsigned

$[-2^{31} \quad 2^{31} - 1]$ → for signed.

long long Range:-

8 bytes = 8 x 8 = 64 bits.

ull → $[0 \quad 2^{64} - 1]$

ll → $[-2^{63} \quad 2^{63} - 1]$

Converting the powers of 2 to powers of 10 for easy Understanding of range:-

$2^{10} \rightarrow 10^3$

$2^{20} \rightarrow 10^6$

$2^{30} \rightarrow 10^9$

$2^{40} \rightarrow 10^{12}$

$2^{50} \rightarrow 10^{15}$

$2^{60} \rightarrow 10^{18}$.

int ~~avg~~ range $= [0 \quad 2^{32}-1]$

for unsigned.

$= [0 \quad 2 \times 10^9 - 1]$

$\cong [0 - 10^9]$.

signed int range: $[-2^{31} \quad 2^{31}-1]$

$: [-2 \times 2^{30} \quad 2 \cdot 2^{30}-1]$

$= [-2 \times 10^9 \quad 2 \times 10^9 - 1]$

$\cong [-10^9 \quad 10^9]$.

long long range :-

ULL : $[0 \quad 2^{64}-1]$

$: [0 \quad 2^4 \times 2^{60} - 1]$

$: [0 \quad 2^4 \times 10^{18} - 1]$

$\cong [0 \quad 10^{18}]$

LL : $[-2^{63} \quad 2^{63}-1]$

$: [-2^3 \times 2^{60} \quad 2^3 \times 2^{60} - 1]$

$\cong [-2^3 \times 10^{18} \quad 2^3 \times 10^{18} - 1]$

$\cong [-10^{18} \quad 10^{18}]$

float :- 4 bytes :

Since it has decimal part & fraction part the range will be definitely less than int.

$\frac{\overline{\quad}}{decimal} \quad \frac{\overline{\quad}}{fraction}$

The range depends on the precision

double : $\frac{\overline{\quad}}{decimal} \quad \frac{\overline{\quad}}{fraction}$.

The range will be definetly less than long

but it depends on the precision.

Points to be considered when solving a problem :-

Problem Statement

I/p format

O/p format

Constraints

Sample I/o

Explanation.

Importance of constraints with example :-

sum of elements in an array with the below constraints : -

$$1 \leq N \leq 10^3$$

$$-10^7 \leq arr[i] \leq 10^7$$

If all the 10^7 values has the value of 10^3 then the sum of all elements will be $10^7 \times 10^3 = 10^{10}$

so the range of result varies from -10^{10} to 10^{10}

which is not in integer range and we have to select the datatype appropriately.