ADSA Session – 5

Bit Manipulation:

Normally we use 6 bit-wise operators

& - Bitwise AND

| - Bitwise OR
|

^ - Bitwise XOR

<< - Bitwise Left Shift

>> - Bitwise Right Shift

~ - Negation

- **These are used to perform operations on Bit level**

Take an Example:

Int a = 97;

Take the binary of 97 by taking LCM of that number.

These binary bits(1 , 0) are stored in memory location (array like)

&- bitwise AND

B1	**B2**	**B1&B2**
1	1	1
1	0	0
0	1	0
0	0	0

a = 46 = 0 0 1 0 1 1 1 0

b = 123 = 0 1 1 1 1 0 1 1

a & b = 0 0 1 0 1 0 1 0 = 42

Result : 42.

| - Bitwise OR
|

B1	B2	B1 | B2
1	1	1
1	0	1
0	1	1
0	0	0

a = 46 = 0 0 1 0 1 1 1 0

b =123= 0 1 1 1 1 0 1 1

a | b = 0 1 1 1 1 1 1 1 = 127

result : 127

^ - Bitwise XOR

B1	B2	B1^B2
1	1	0
1	0	1
0	1	1
0	0	0

a = 46 = 0 0 1 0 1 1 1 0

b =123= 0 1 1 1 1 0 1 1

a ^ b = 0 1 0 1 0 1 0 1 = 55

result : 55

Task:

a = 87   b = 59   Find a&b , a|b , a^b

a = 0 1 0 1 0 1 1 1

b = 0 0 1 1 1 0 1 1

a & b = 0 0 0 1 0 0 1 1   ~ 19

a | b = 0 1 1 1 1 1 1 1   ~ 127

a ^ b = 0 1 1 0 1 1 0 0   ~ 108

Find -97?

97 = 0 1 1 0 0 0 0 1

-97 = 1 0 0 1 1 1 1 1   ~ 2's compliment of 97

Important Conditions:

If (n % 2) -

            If n & 1 : if 1 – n is odd

                    If 0 – n is even

            If n & 0 : 0

            If n & n : n

If (n | 1) -

            If n : n is odd

            If n+1 : n is even

 n | 0 : n

 n | n : n

if (n ^ 1) -

                If n-1: n is odd

                If n+1: n is even

n ^ 0 : n

n ^ n : 0

Important Logic: