

Lecture: - 5 Operators & Loop

Bitwise Operators :- work on Bit level.

And $\rightarrow \&$ $\rightarrow a=2, b=3 \Rightarrow 2 \& 3 \Rightarrow 11$
 OR $\rightarrow |$
 NOT $\rightarrow \sim$
 XOR $\rightarrow ^$

(In case of And)

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

OR operation :-

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

$a=2 \Rightarrow 10$
 $b=4 \Rightarrow 100$
 $\underline{110}$

NOT operation :-

x	y
0	1
1	0

$a=2 \Rightarrow 10$
 int $\Rightarrow 8$ byte $\Rightarrow 32$ bits
 0000 0000 ... 0010

It means $a \& b$
 $(a \& b = 3)$

To find $(\sim a)$
 (-3)

XOR $\rightarrow ^$

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

$a=2 \Rightarrow 10$
 $b=4 \Rightarrow 100$
 $\underline{010}$
 $a=5 \Rightarrow 101$
 $b=7 \Rightarrow 111$
 $\underline{010}$

find \sim
 $(\sim a) \Rightarrow 11111111$
 this means \sim is -ve
 then find 2^{31} instead
 the binary no.
 $\Rightarrow 0000 \dots 10$
 $\& 1$
 $2^{31} \Rightarrow 0000 \dots 11 \Rightarrow 3$

Left Shift operators & Right Shift operators :-

Left Shift $\rightarrow s \ll 1$ means shift start left shift start.

$0000 \dots 0000 \dots 0101$
 $0000 \dots 1010$

$s \ll 2$ means 3 shift start left shift

$0000 \dots 0011$
 $0000 \dots 1100$

Right Shift operators :-

ex. $-15 \gg 2, 5 \gg 2$

$(\ll, \gg) \rightarrow$ if no is +ve
 padding with 0. (means shift no
 last mei ya starting mei
 ayega vo zero wala).

$(\ll, \gg) \rightarrow$ if no is -ve.
 padding is depending on compiler

Increment/Decrement operators

$i++$; or $i+=1$; → post increment
 $++i$; → pre increment
 $i--$; → post decrement
 $--i$; → pre decrement

post increment :- $i++$ (पहले use करते और तब statement execute हो जाते तब value increase करते)

$\text{int } i=4;$
 $\text{int } a = i++;$

$\text{int } i=3, a=2;$ → increase करेंगे
 $\text{sum } = a + i;$
 $2+3$
 $= 5$

$i=4$ → After executing sum statement

pre increment :- $++i$;

$\text{int } i=4;$
 $\text{int } a = ++i;$

$i=5$ → Before printing value of a.
 $a=5$

for loop:-

for (initialization; condition; inc/dec)

$\{$
 $\}$
 $\text{body} \{$
 $\}$
 $\text{else} \{$
 $\}$

$\text{for } (\text{int } a=0, b=1; a \leq b; a++, b--)$
 $\{$
 $\text{cout} << a << " " << b << endl;$
 $\}$

first → second → third → fourth

It will take out break; out of the loop.

Fibonacci Series:- 0, 1, 1, 2, 3, 5, 8, ...

$$n = (n-1) + (n-2)$$

break:- keyword use to come out of the loop.
continue:- To skip iteration.

Levenshtein distance:-

1 2 3 4 5
6 0/1 = 0
let no = 5
means 5 is Levenshtein no.

variables & scope:- Agar koi kisi block mei define ho raha hai to uski lekhani uske block mei hi hai.

int main()

{
 int a = 2;
 cout << a << endl;
}

{
 if (true)

{
 int b = 20;

 int a = 10;

 cout << a << endl;

 cout << b << endl;
}

operator precedence:- { [()] } . . .

LeetCode Question

1) Subtract the product & sum of digits of integer.

n = 234

n % 10 = 4, 3, 2

n / 10 = 23, 2, 0

if n == 0
stop the loop.

2) No of 1 Bits.

n = 0000 1011

o/p = 3 do right shift till n != 0
& check (n & 1)