

॥श्री गणेशाय नमः॥

## LECTURE 5 →

# BITWISE OPERATORS, FOR Loops, OPERATOR PRECEDENCE & VARIABLE SCORING.

## \* Bitwise Operators →

AND → &

OR → |

NOT → ~

XOR → ^

### Bitlevel

① AND → & →  $a = 2, b = 3$        $2 \rightarrow 10$   
 $3 \rightarrow 11$   
 $10 \rightarrow 2$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} a \rightarrow 5 \rightarrow 101 \\ b \rightarrow 7 \rightarrow 111 \\ \hline 101000 \end{array}$$

② OR → |

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	11

$$a = 3 \quad 11$$

$$b = 6 \quad \underline{100} \quad \text{OR} \quad \underline{111}$$

③ NOT → ~

A	Z
0	1
1	0

$$a = 2^0 \dots 10$$

$$\begin{array}{r} 11111101 \rightarrow \\ \downarrow 11111101 \\ + 1 \\ \hline 11 \end{array}$$

-Ney.

↳ 2's complement

④  $\text{XOR} \rightarrow \wedge$

A	B	V
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{array}{l} a = 2 \quad 010 \\ b = 4 \quad 100 \\ \hline 110 \end{array}$$

$$\begin{array}{l} a = 5 \quad 101 \\ b = 7 \quad 111 \\ \hline 010 \end{array}$$

\* Implement  $\rightarrow \&, |, \sim, \wedge$

int a = 4;

int b = 6;

$$4 \rightarrow 100$$

$$\text{cout} \ll (a \& b); \rightarrow 6 \rightarrow 110$$

$$\hline 100 \rightarrow 4$$

$$4 \rightarrow 100$$

$$\text{cout} \ll (a | b); \rightarrow 6 \rightarrow 110$$

$$\hline 110 \rightarrow 6$$

Cout  $\ll \sim a; \rightarrow 100$

$$\begin{array}{r} 1111011 \\ 0010100 \\ \hline 101 \end{array}$$

$\rightarrow -5$

Cout  $\ll (a \wedge b);$

$$a \rightarrow 4 \rightarrow 100$$

$$b \rightarrow 8 \rightarrow 100$$

010

$\rightarrow 2$

\* Left Shift  $\rightarrow \ll$   $\rightarrow$  Multiply by 2.

5  $\ll 1$

$\hookrightarrow$  5 ko एक बार Left Shift करदे

$$\begin{array}{r} 000 \\ 000 \\ \hline 101 \\ 1010 \end{array}$$

$$3 \ll 2 \leftarrow 3 \times 2 \rightarrow 6$$

$$6 \times 2 \rightarrow 12$$

$$0000 \rightarrow 011$$

$$0000 \rightarrow 0110 \leftarrow 1^{\text{st}}$$

$$00 \rightarrow 00100 \leftarrow 2^{\text{nd}}$$

(2)  $\ll$

\* Right Shift  $\rightarrow \gg$  Divide by 2.

15  $\gg 1 \rightarrow 8$

$5 \gg 2$

$$00 \rightarrow 101 \leftarrow 5/2$$

$$00 \rightarrow 10 \leftarrow 2/2$$

$\ll, \gg \rightarrow$  for the padding with '0'.

padding with '0'

-ve  $\rightarrow$  padding  $\hookrightarrow$  compiler dependent

$$17 \gg 1 \rightarrow 8 \rightarrow 17/2 \rightarrow 8$$

$$17 \gg 2 \rightarrow 4 \rightarrow 17/2 + 8 \rightarrow 8/2 \rightarrow 4$$

$$19 \ll 1 \rightarrow 19 \times 2 \rightarrow 38$$

$$21 \ll 2 \rightarrow 21 \times 2 \rightarrow 42 \times 2 \rightarrow 84$$

## ④ Increment Operator:

$i = i + 1 \rightarrow i++ ; \rightarrow \text{Post inc.}$   
 $\downarrow \text{@@}$        $i++ ; \rightarrow \text{Pre inc.}$

$i = i - 1 \rightarrow i-- ; \rightarrow \text{Post dec.}$   
 $\downarrow \text{@@}$        $--i ; \rightarrow \text{pre dec.}$

## ⑤ Post increment $\rightarrow i++$ :

(use karne ke baad chalta hoga)  $a = 2$ ;  
 $\text{int } i = 4 ; \quad \text{int } i = 3 ;$   
 $\text{int } a = i++ ; \quad \text{sum} = a + (i++) ;$   
 $a \rightarrow ? \rightarrow 4 \quad \text{sum} \rightarrow ? \rightarrow 5$   
 $i \rightarrow ? \rightarrow 5$

## ⑥ Pre-increment $\rightarrow ++i$ :

(pहले increment करना fir use karana)  
 $i = 3$   
 $\text{int } i = 11 ; \quad a = 2$   
 $\text{int } a = ++i ; \quad \text{sum} = a + (++i)$   
 $\checkmark \quad \quad \quad = 6 \quad \quad \quad 4$   
 $i \rightarrow 12 \quad \quad \quad a \rightarrow 12$   
 $i \rightarrow 12 \quad \quad \quad a \rightarrow 12$

## ⑦ Post Decrement $\rightarrow$

(use karne ke baad kam kar dunga (dec))

$\text{int } i = 2 ;$   
 $\text{int } a = i-- ;$   
 $\downarrow$   
 $a \rightarrow ? \rightarrow 2. \quad i \rightarrow 1 ,$

## ⑧ Pre Decrement $\rightarrow$

(phel le kam kar dena baad me use karo)

$\text{int } i = 4 ;$

$\text{int } a = --i$

$a \rightarrow ? \rightarrow 3$

$i = 3$

Codes  $\text{int } i = 7 ;$

$\text{cout} \ll (++i) ; \rightarrow 8$

$\text{cout} \ll ((i++) ; \rightarrow 8 \rightarrow i = 9$

$\text{cout} \ll (i--) ; \rightarrow 9 \rightarrow i \rightarrow 8$

$\text{cout} \ll (--i) ; \rightarrow 7 \rightarrow i \rightarrow 7$

(24.36)

प्र० गणेशाम नमः ॥

### ④ For Loop →

i to n print →

int n;

cin >> n; initial variable

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

cout << i << "

}

for ( initialization ; condition ; inc/dec )  
not mandatory      not mand.      not manda.

int i = 1;

for ( ; i <= n; ) {

cout << i << endl;

i++;

}

for ( ; ; ) {  
if ( i <= n ) {  
cout << i;  
} else {

break;

} // for loop

i++;

}

⑤ Multiple variable, conditions & operations are allowed.

for (int a = 0, b = 1; a >= 0 && b >= 1; a--, b--) {

cout << a << " " << b << endl;

}

→ 0 1.

one-time  
for ( ↓ , ← , ← )  
↓      ←      ←

3

Fibonacci Series →

0, 1, 1, 2, 3, 5, 8, 13, 21...  
+    +    +    +    +    +    +

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

Code →

int n = 10;

int a = 0;

int b = 1;

for (int i = 1; i <= n; i++)

cout

int nextNum = a + b;

cout << nextNum.

sum =  $\cancel{a}$   $\rightarrow$  a = b;  
 $\cancel{b}$   $\rightarrow$  b = nextNum;  
 $\cancel{a} = \cancel{b}$

a    b  
0, 1, 1, 2, 3, 5, 8, 13, 21  
↓    ↓    ↓  
a    b    a+b    a+b+a+b    a+b+a+b+a+b  
↓    ↓    ↓    ↓    ↓  
Fib. Num.

a = b;

b = nextNum;

3 ..

## ④ Prime Number →.

$n \rightarrow 7 \rightarrow$  prime or Not.

$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \diagdown & & & & & & \diagup \\ & i & & & & & \end{array}$

$n \% i == 0 \rightarrow$  not prime.

$j \rightarrow 2 \text{ to } n-1$ .

$n \% i == 0 \rightarrow$  prime no.

break;

```
int n;
cin >> n;
bool isPrime = true;
for (i=2; i<n; i++) {
    if (rem == 0, Not a prime).
    if (n % i == 0) {
        cout << "Not a Prime Number" << endl;
        break;
    }
    isPrime = 0;
}
if (isPrime) {
    cout << "Prime Number" << endl;
} else {
    cout << "Not prime number" << endl;
}.
```

## ④ Continue → to skip particular iteration:-

```
for (int i=0; i<5; i++) {
```

↙

cout << "Hi";

cout << "Hey";

continue;

Unreachable ↓ cout << "Reply to border";

3 4

## ④ Variables & Scopes →

```
int main() {
```

↳ lifetime / impact.

```
    int a = 3;
```

```
    cout << a << endl; → 3
```

```
    if (true){
```

↳ no error a is within if scope.

```
        int a = 5;
```

```
        cout << a; → 5
```

```
}
```

```
    cout << a; → 3
```

```
    if (true){
```

```
        int b = 5;
```

```
        cout << b << endl;
```

```
}
```

```
    cout << b; → #error
```

(we can't access it outside that scope).

Case-1 →

```

int i = 8;
for(int i <= 8; i++) {
    cout << "H" << endl;
}
    
```

Case-2 →

```

int i = 8;
for(; i <= 8; i++) {
    cout << "H" << endl;
}
    
```

**IMP** अगर कोहू कोई Variable block में Define हो रहा है। तो उसकी Presence 3ss Block के अंदर ही है। उसके बाहर को मर जाएगा। अंदर जितना मर्जी Use करलो।

**IMP** Valid No Error →

```

if(1) {
    int b;
    if(1) {
        int b;
        if(1) {
            int b;
        }
    }
}
    
```

No error. ये चलेगा।

b. १ टाक के  
b Block में  
Multiple 'b'  
not allowed,,

b. { cout << b  
} This b can be used  
is whole { } block.

④ Operator Precedence → BODMAS.

(किसकी Authority प्यारा है)  
(पहले कौनसा execute करने वाला है)

OP →  $2 \times 3 / 4 + 5$

① LEETCODE QUESTION 1 → 1281.

Given an integer n, return the difference between the product of its digits and the sum of its digits.

n → 234

O/P → 15

Explanation

Product of digits →  $2 \times 3 \times 4 \rightarrow 24$

Sum of digits →  $2 + 3 + 4 \rightarrow 9$

Result →  $24 - 9$

→ 15

$$\begin{array}{r} 23 \\ \hline 10) 234 \\ 20 \quad \quad \\ \hline 34 \\ 30 \quad \quad \\ \hline 4 \end{array}$$

$$n = 234$$

$$n \% 10 \rightarrow 4$$

$$n = 234 \rightarrow 4 \text{ को हटाना है.}$$

$$n / 10 \rightarrow 23$$

$$\begin{array}{r} 23 \\ \hline 10) 234 \\ 20 \quad \quad \\ \hline 34 \\ 30 \quad \quad \\ \hline 4 \end{array}$$

**(IMP)** किसी भी Number का Last Digit निकालना है. तो.

number % 10

$$\begin{array}{l} \textcircled{i} \ n \rightarrow 234 \% 10 \rightarrow 4, 3, 1, 2 \\ \textcircled{ii} \ n \rightarrow 23 / 10 \rightarrow 23, 2, 0, 1, 0 \end{array}$$

जब  $n \rightarrow 0$  तब लक जा भाइ!!!

$$\text{prod} = 1$$

$$\text{sum} = 0$$

$$234$$

$$\rightarrow \text{digit} \rightarrow 4$$

$$\text{product} \Rightarrow \text{product} \times \text{digit}$$

$$\text{sum} = \text{sum} + \text{digit}$$

ans  $\rightarrow$  product - sum ;

code



`cin >> n;`

`int product = 1;`

`int sum = 0;`

`while (n != 0) {`

`int single_digit = n % 10;`

`product = product * single_digit;`

`sum = sum + single_digit;`

`n = n / 10;`

`}`

`int answer = product - sum;`

`cout << answer;`

\* (\*) LeetCode Question 2  $\rightarrow$  191. Number of 1 bits.

$$n \rightarrow 11$$

$$011 \rightarrow 3$$

$$(1011)$$

$$11 \rightarrow \underline{\hspace{1cm}} \rightarrow 1011.$$

Convert to binary & count -> 1011

100  $\rightarrow$  right shift  $\rightarrow$  32 bit

32 baar right shift

$$\begin{array}{r} 00 \rightarrow 001 \rightarrow n \neq 1 \\ n! = 0 \end{array}$$

Code :-

`int count = 0;`

`while (n != 0) {`

`// checking the last bit.`

`if (n & 1) {`

`count++;`

`}`

`n = n >> 1;`

`}`

`cout << "Number of 1's in << n << " : << count;`

3.

॥ भी गणेशाय नमः ॥

## LECTURE - 6 : BINARY & DECIMAL NUMBER SYSTEM.

### ★ Decimals & Binary →

Decimals Number Sys → 10, 11, ...  
Binary 1, 1, 1 → 0, 1.

### ★ Convert Decimal to Binary →

① 1st Approach →

②  $n = 10$ .

→  $n / 2$ .

→ store ~~ans~~ remainder in answer.

→ repeat above 2 step until  $n! = 0$ .

→ reverse ~~ans~~ answer.

$n = 10$	Division	Remainder	$2 \overline{) 10 } \quad 5$
divisor → 2	$10_2 \rightarrow 5$	0	$\frac{1}{0}$
answer.	$5_2 \rightarrow 2$	2	
<del>ans</del>	$2_2 \rightarrow 1$	0	
$\downarrow$	$1_2 \rightarrow 0$	1	

$n = 0$  सके जाना.

③  $n = 7$

divisor → 2	Div	Rem.	⑦
	$7/2 \rightarrow 3$	1	$4 + 2 + 1$
answer.	$3/2 \rightarrow 1$	1	$2^2 \quad 2^1 \quad 2^0$
<del>ans</del>	$1/2 \rightarrow 0$	1.	1 1 1

### ★ 2nd Approach →

$[n \rightarrow 5] \rightarrow 101$

n

5

101

1

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

$$1, 2, 3 \rightarrow \underline{321}$$

reverse flow

$$\text{ans} = 0$$

$$\begin{aligned}\text{ans} &= \text{digit} \times 10^0 + \text{ans} \\ &= 1 \times 10 + 0 \\ &= 1.\end{aligned}$$

∴ if we use →

$$\begin{aligned}\text{⑩ } \text{ans} &= 2 \times 10^1 + 1 \\ &= 20 + 1 \\ &= 21\end{aligned}$$

$$\boxed{\text{ans} = \text{digit} \times 10^i + \text{ans}}$$

we will get  
rev integer.

$$\begin{aligned}\text{⑪ } \text{ans} &= 3 \times 10^2 + 21 \\ &= 300 + 21 \\ &= 321\end{aligned}$$

$$\boxed{\text{rev} = \text{digit} \times 10^i + \text{rev}}$$

$$1, 2, 3 \rightarrow 123$$

same flow

$$\text{ans} = 0$$

$$\begin{aligned}\text{⑫ } \text{ans} &= (\text{ans} \times 10) + \text{digit} \\ &= 0 \times 10 + 1 \\ &= 1\end{aligned}$$

$$\begin{aligned}\text{⑬ } \text{ans} &= (1 \times 10) + 2 \\ &= 10 + 2 \\ &= 12.\end{aligned}$$

$$\boxed{\text{ans} = \text{ans} \times 10 + \text{digit}}$$

$$\begin{aligned}\text{⑭ } \text{ans} &= (12 \times 10) + 3 \\ &= 120 + 3 \\ &= 123\end{aligned}$$

Code to convert Decimal Binary to Binary →  
#include <math.h>

int n;  
cin >> n;

float rev;  
rev = 0;

int i = 0;

while (n != 0){

int bit = n & 1;

~~rev~~ = (bit \* pow(10, i)) + ~~rev~~;

n = n >> 1;  
i++;

}

cout << "Answer is " << ans << endl;

n = 123      1000      bit → 0

1st rev = (0 × 10<sup>0</sup>) + 0      0 + 0  
= 0      bit → 0

2nd rev = (0 × 10<sup>1</sup>) + 0      00 + 1  
= 0      bit → 1

3rd rev = (1 × 10<sup>2</sup>) + 0      000 + 1  
= 100 + 0      bit → 1.  
= 100

4th rev = (1 × 10<sup>3</sup>) + 100

= 1000 + 100      ↳ n >> 1  
= 1100      ↳ 0000

exits loop ↳,

int range  $\rightarrow (-2^{31}, 2^{31}-1)$

binary of n = -6

① ignore negative.

6

0000 0110

$\rightarrow 1001 \leftarrow 2^3$   
+ 1  $\leftarrow$  Add 1.

$\rightarrow 111010$

② Convert -ve Decimal to binary

③ Binary to Decimal  $\rightarrow$

10101

↓ ↓ ↓ ↓ ↓

$2^4 2^3 2^2 2^1 2^0$

↙ ↘ ↘ ↘ ↘

$2^4 + 2^2 + 2^0 \rightarrow 16 + 4 + 1 \rightarrow 21$

Div	2	21	rem
2	10	1	
2	5	0	
2	2	1	
1	0		

>>  $\rightarrow$  /

<<  $\rightarrow$  x

Ex. n = 2

$2 \times 2$

$\ll 1$       4

10

$\downarrow 100 \rightarrow 4$

$\ll 1 \rightarrow 8$

1000  $\rightarrow$

$\ll 1 \rightarrow 16$

10000  $\rightarrow 16$

int n; ans=0; i=0;

cin >> n;

while (n != 0) {

int digit = n % 10;

if (digit == 1)

ans = pow(2, i) + ans;

}

n = n / 10;

i++;

}

cout << ans;

## LECTURE → 7

### ① LEETCODE PROBLEM SOLVING SESSION .\*

7. Reverse a integer : 32 bit int → signed (+/-) kuch bhi.

I/P → x = 123

O/P → 321

( $-2^{31}, 2^{31}-1$ )

If over range then 0,

I/P → x = -123

O/P = -321

I/P → x = 120

O/P = 21

Cases →

case 1 → normal case → reverse.

case 2 →  $\{ \begin{array}{l} 1 \\ -3 \end{array} \}$

$\{ \begin{array}{l} 3 \\ -1 \end{array} \}$  → over range.

→ return 0.

ans = 0

num = 123

$$123 \% 10 = 3$$

$$12 \% 10 = 2$$

$$1 \% 10 = 1$$

$$\text{ans} = \text{ans} \times 10 + \text{digit}.$$

$$= 3 \times 10 + 2$$

$$= 32 \times 10 + 1$$

$$= 320 + 2$$

$$= 322$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$\begin{array}{r} 546 \\ \downarrow \quad \downarrow \\ 5 \times 100 + 4 \times 10 + 6 \times 1 \end{array}$$

$$= 321$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

$$= 320$$

$$= 321$$

$$= 321$$

&lt;math display

$\text{INT\_MIN} = 2^{31}$ ,  $\text{INT\_MAX} \rightarrow 2^{31}-1$

$\text{if } \left( \frac{\text{ans} > \text{INT\_MAX}}{10} \right) \text{ || } \left( \frac{\text{ans} < \text{INT\_MIN}}{10} \right)$

Number.  
का करेगा.

return 0;

3

दो नं. जिसे बड़ा करेगा

$\text{ans} \times 10 > \text{INT\_MAX} \times 10$ ,  $\text{ans} \times 10 < \text{INT\_MIN} \times 10$

10, 10,

$\text{ans} > \frac{\text{INT\_MAX}}{10} \text{ || } \text{ans} < \frac{\text{INT\_MIN}}{10}$ .

मैं से नहीं कर सकता ↴

$\text{ans} \times 10 > \text{INT\_MAX}$        $\text{ans} \times 10 < \text{INT\_MIN}$ .

इसलिए 10 को उधर Shift किया →

$\text{ans} > \frac{\text{INT\_MAX}}{10} \text{ || } \text{ans} < \frac{\text{INT\_MIN}}{10}$

return 0;

$\text{ans} \times 10 - (2^{31}-1) \times 10 \rightarrow$

\* code →

```
int n;
cin >> n;
long long ans = 0;
```

while ( $n \neq 0$ )

int digit =  $n \% 10$ ;  
 $\text{ans} = (\text{ans} * 10) + \text{digit}$ .

if ( $(\text{ans} > \text{INT\_MAX}) \text{ || } (\text{ans} < \text{INT\_MIN})$ )

return 0;

n = n / 10;

ans =  $\text{ans} \leq 0 ? -\text{ans} : \text{ans}$ ;

`cout << ans << endl;`

Leetcode

II) 1009. Complement of Base 10 Integer

$$n = 5 \rightarrow 101$$

$$010 \rightarrow 2 \leftarrow \text{Ans.}$$

$$n = 7 \rightarrow 111$$

$$1's \text{ comp} \rightarrow 000 \rightarrow 0 \dots$$

$$n \rightarrow 5 \rightarrow 101$$

$$000 \quad 0101$$

$$\sim n = \sim 5$$

$$\begin{array}{r} 11111 \\ - 1010 \\ \hline \text{ignore.} \end{array}$$

$$\sim \text{mask}$$

$$\begin{array}{r} 11111 \\ - 00000 \\ \hline 1111 \quad \text{← Mask} \\ \sim \sim \sim 0010 \end{array}$$

$$\text{mask.}$$



right shift  $\gg$

$$\begin{array}{l} 1^{\text{st}} \rightarrow 000 \quad 0010 \rightarrow !=0 \\ 2^{\text{nd}} \rightarrow 000 \quad 0001 \rightarrow !=0 \\ 3^{\text{rd}} \rightarrow 000 \quad 0000 \rightarrow =0. \end{array}$$

$\ll$

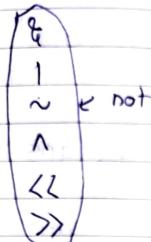
$$0000 \ 0 \ 0$$

$$0000 \quad 1$$

$$0000 \quad 11$$

$$0000 \sim 111$$

bits के साथ खेलते हैं.



cin >> n;

int m = n;

int mask = 0;

// edge case.

if ( $n == 0$ )

return 1;

while ( $m \neq 0$ )

{

    mask = (mask << 1) | 1,  
    m = m >> 1,

}

    int ans = ( $\sim n$ ) & mask;

return ans;

3.

Leetcode →.

231. Power of two →

(iii)  $n = 1, \text{ op} \rightarrow \text{true}, 2^0 = 1$ . $\therefore \text{No } \rightarrow ? \text{ } \times$ .

$$n = 16 = \frac{8}{2} = \frac{4}{2} = \frac{2}{2} = 1 \rightarrow \text{Yes}, \ldots \times$$

$$\frac{14}{2} \rightarrow \frac{7}{2} \rightarrow \frac{3}{2} \rightarrow 1 \rightarrow \text{No } \times$$

$2^0$   
 $2^1$   
 $2^2$   
 $\vdots$   
 $2^{31}$

→ equal → Yes

→ Not equal → No.

```
int n;
cin >> n;
```

```
#include <iostream>
using namespace std;

bool powerOfTwo(int n) {
    if (n == 1) return true;
    if (n < 1) return false;
    int ans = 1;
    for (int i = 0; i < 30; i++) {
        ans *= 2;
        if (ans == n)
            return true;
    }
    return false;
}
```

$$\begin{aligned}
 1^{\text{st}} \rightarrow \text{ans} &\Rightarrow 1 \times 2 & \text{ans} = 2 \times 2 & \text{ans} = 4 \times 2 \\
 &= 2. & &= 4 \\
 & & &= 8.
 \end{aligned}$$

Code →

```
int ans = 1;
for (int i = 0; i < 30; i++) {
    cout << ans << endl;
    if (ans == n)
        break;
}
```

```
return true;
if (ans < INT_MAX / 2)
```

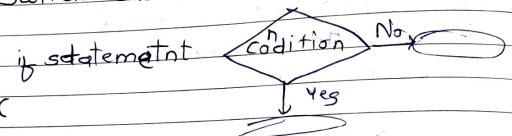
```
ans = ans * 2;
```

```
return false.
```

## LECTURE - 8

### SWITCH STATEMENT & FUNCTIONS.

Switch Case → Control Statement



switch (exp) → int / char / float / string

case constant: → Yes / No

break → Yes / No

case constant 2: → Yes / No

break; → Yes / No

default: → Yes / No

3.

int num = 2;  
switch (num){

case 1: cout << "First" << endl;  
break;

case 2: cout << "Second" << endl;  
break;

default: cout << "It is default case" << endl;

3. ↳ second

char ch = '1';  
switch (ch){

case 1: c -  
b -

case '2':

default: ↳ 1

3.

break; → अगर इस Case के आगे का कुछ भी लाइनेक्स क्यूट मत करना.

④ Nested switch → Not necessary

char ch = '1';

int num = 1;

switch (ch){

case '1': cout << "char" << endl;

break;

case '2': switch (num){

case - 1: cout << "Value" << num << endl;

break;

break;

default: →

3.

H.W  
 while(1){  
 switch( ){  
 exit(0);

}  
 } → इस loop से बाहर कैसे आयेंगे ?

switch  
 continue → is not valid.

H.W  
 i/p → a, b, operation  
 ex → 3 4 \* + / - %

o/p → answer =  $3 * 4 \rightarrow 12$ .

calculator प्यारासा →

```
int a,b;  

cin>>a;  

cin>>b;
```

char op;  
cout << "Enter operation you want to perform" << endl;

```
switch(op){  

    case '+':  

        break;  

    case '-':  

        break;  

    case '*':  

        break;  

    case '/':  

        break;  

    default : cout << "Wrong I/p" << endl;
```

H.W Due →  
 Total Amount = 1330 → 100 (13)  
 50  
 20 (1)  
 ↘ notes. (10)

13 → 100 ₹

1 → 20 ₹

10 → 1 ₹

$$\frac{1330}{100} = 13, 1330 - 1300 = 30$$

$$\frac{30}{20} = 1, 30 - 20 = 10$$

$$\frac{10}{1} = 10, 10 - 10 = 0.$$

100-X  
 50-Y  
 20-Z  
 1-a.

Code →

```
int n;  

int notes_100 = 0,  

notes_50 = 0,  

notes_20 = 0,  

coins_1 = 0;
```

cin >> n;

```
while (n > 0){  

    switch(n/100){
```

case 0:

switch(n/50){

case 0:

switch(n/20){

case 0:

coin\_1 = m;

n = 0;

break;

default :

notes\_20 = n/20;

n = n% 20;

break;

```

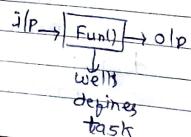
break;
default:
notes_50 = n / 50;
n = n % 50;
break;
}
break;
default:
notes_100 = n / 100;
n = n % 100;
break;
}
cout << notes_100 << endl;
cout << notes_50 << endl;
cout << notes_20 << endl;
cout << coins_1 << endl;

```

## ④ Functions →

- ↳ If not function ↳
- ↳ Bulky code → if we not use functions .....  
(Buggy)
- ↳ It is not readable

↳ A program जिसका well defined task ..



Syntax →

```

int power
<return type> <function-Name> (<arguments />
                                     ^/p parameters)
{
    3 return ans;
          ^ int

```

code →

#

```

int power(int a, int b) {
    int ans = 1;
    for (int i = 1; i <= b; i++) {
        ans = ans * a;
    }
    return ans;
}

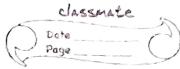
int main() {
    int num1, num2;
    cin >> num1 >> num2;
    int output = power(num1, num2);
    cout << output << endl;
    return 0;
}

```

}

1.

46.20



$nCr$  Program →

$$nCr = \frac{n!}{x! * (n-x)!}$$

$i \rightarrow n, x$

$0 \rightarrow nCr$ .

$nC_0 \rightarrow 1.$

$$8C_0 \rightarrow \frac{8!}{0! * (8-0)!} \rightarrow \frac{8!}{1 \times 8!} = 1.$$

$$8C_2 \rightarrow \frac{8!}{2! * (8-2)!} = \frac{8!}{2! \times 6!} = \frac{8 \times 7}{2} = 28$$

```
int factorial(int n){  
    int fact = 1;  
    for(int i=1; i<=n; i++){  
        fact = fact * i;  
    }  
    return fact;
```

```
int nCr(int n, int x){  
    int numerator = factorial(n);  
    int denominator = factorial(x) * factorial(n-x);  
    int nCr = numerator / denominator;  
    return nCr;
```

}

\* Counting Program →

void → यह Function कुछ भी Return नहीं करते वाला.

// Function Signature ↗

void printCounting(int n){

// Function body.

for(int i=1; i<=n; i++){  
 cout << i << " ";

}

cout << endl;

}

int main(){

int n;

cin >> n;

// function call.

printCounting(n);

return 0;

}

\* Is prime →

bool isPrime(int n){

for(int i=2; i<n; i++){

// divide hogaya, not prime.

if(n % i == 0){

return 0;

}

return 1;

}

```

int main(){
    int n;
    cin >> n;
    if (isPrime(n)) {
        cout << "Prime";
    } else {
        cout << "Not prime";
    }
}

```

return - types → *i/p arguments/param.*  
name( ) {

void / int / char /  
float / string.  
3 return.

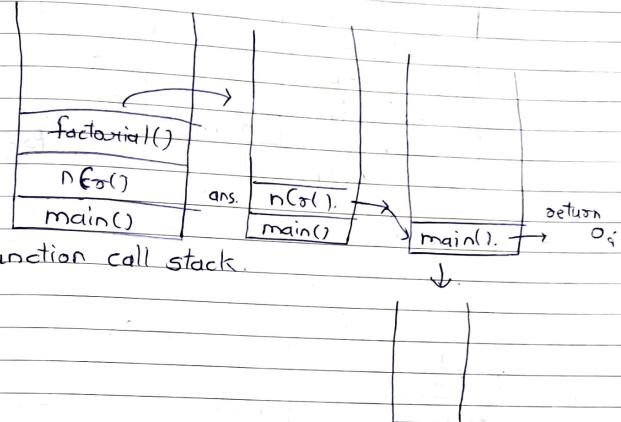
void f( ){

3 return;

यहांसे सिर्फ वापस चले जाओ !!!

$$\begin{array}{r}
 0, 1, 1, 2, 3 \\
 1 + 1 \\
 \hline
 2 \quad \text{ans}
 \end{array}$$

(Shadi ki ~~टेक्स्ट~~)  
Function Call Stack →



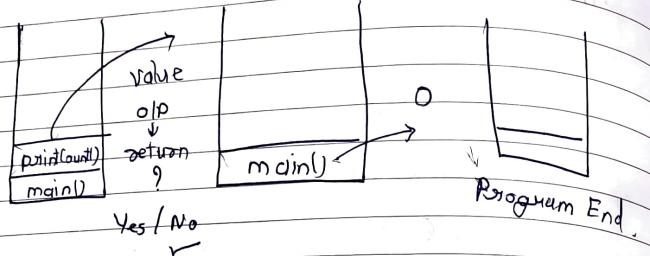
① (H/W) → A.P =  $(3 * n + 7)$   
*i/p* =  $n \rightarrow 3 \rightarrow 3 * 3 + 7$   
*o/p* →  $n^{\text{th}}$  term. →  $9 + 7 \rightarrow 16$ ,

② (H/W) → a & b.  
*o/p* → total no. of set bits in a & b.

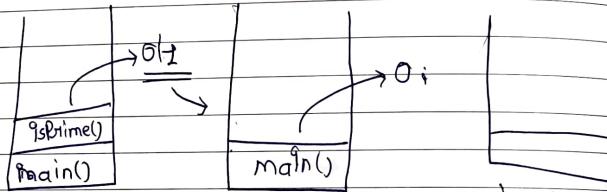
$$\begin{array}{lcl}
 a = 2 & \rightarrow 10 & \text{o/p} \rightarrow 3 \\
 b = 3 & \rightarrow 11 & \\
 \end{array}$$

③ (H/W) →  
*i/p* →  $n$ .  
*o/p* →  $n^{\text{th}}$  Fibonacci.  $n=5 \rightarrow$   
 $0, 1, 1, 2, 3, 5, 8, 13$ .  $\text{o/p} \rightarrow 3$ .

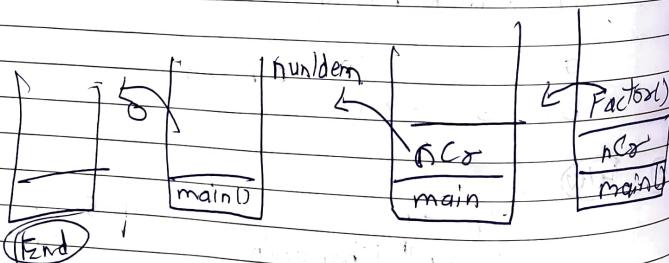
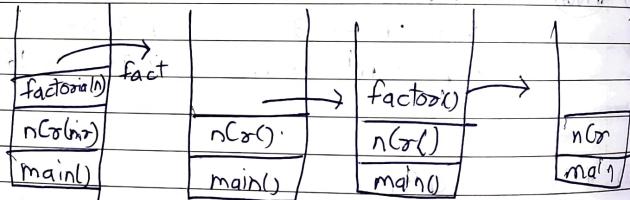
Counting



② isPrime →



③ nCr →



\* Pass by Value →

```
void dummy (int n){ n++;  
cout << "n is " << n; } → 15
```

```
int main(){ int n;
```

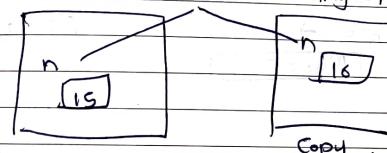
```
cin >> n;
```

```
dummy(n);
```

```
cout << "number n is " << n << endl;
```

```
}
```

main different dummy n++ ?

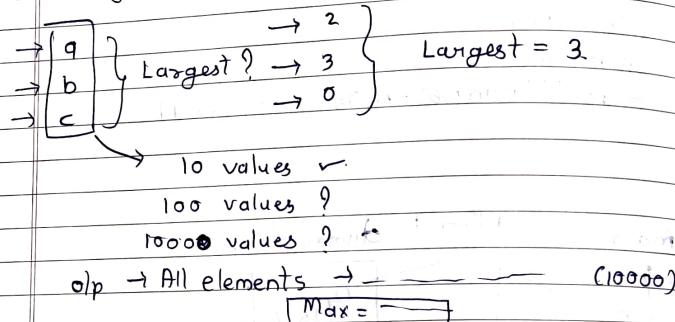


dummy(n) → (function के अंदर n की copy बन गयी)

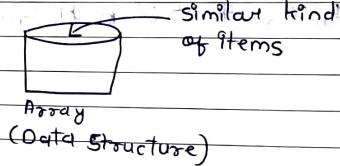
## LECTURE 9

### INTRO TO ARRAY IN C++

⊕ Arrays →



Array → same type के items डाल सकते हैं. (list)



int → 100

array of int  
100

Arrays →   
① Similar type of Item.  
② Values stored will be in contiguous location. (सन्तान)

Array 5 int.

3	15	9	1	2
107	108	109	100	110

107 108 109 100 110.

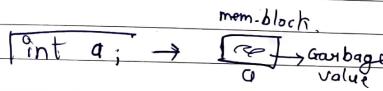
⊕ "Index" के द्वारा Elements Access कर सकते हैं.

⊕ Why array?

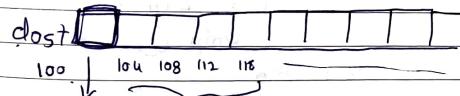
↪ 10000 values → 10000 variables.  
↓  
1 variable.

↪ जब हम Multiple values एक Single variable में Store करना है.

⊕ Implementation →



int dost[10] //Declaration.



IMP (14:06)  
dost 100 location की दर्शा कर रहा है!!!

## ④ Indexes in Array.

int v[5]

0	1	2	3	4
2	6	8		

100 104 108 112 116

v[1] → 100 + 1 × 4  
 ↓      ↓      ↓      ↓  
 v[1]  
 @ress      Bytes      ⇒ 6

v[0] → 2

v[3] → 100 + 3 × 4

→ 112 → 8

index

starts from '0'  
 ends with n-1;

Declaration ✓  
 Access ✓

## ⑤ Initialization →

int dosti[3] = {5, 7, 11};

int array[100000] = {0};

↳ इस Element '0'

Only ①

H.W. Initialize array to  
 hisi bhi value se initialize करें।

code →

// declare

int number[15];

cout << number[0];

cout << number[20]; // error

// new

int second[3] = {5, 7, 11};

int third[15] = {2, 7}, everything is find.

// printing the arr.

int n = 3;

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

cout << third[i] << " ";

}

int fourth[10] = {0}.

सब = 0 initialize

- ④ Making print array function instead calling multiple times. →

```
void printArray(int arr[], int size)
{
    cout << "Printing the array !! " << endl;
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << "Printing Done [ ] !! " << endl;
}
```

- ⑤ Finding the size of array ← how many element →

```
int size = sizeof(arr) / sizeof(int);
```

```
cout << "Array size is " << size;
```

- ⑥ Any datatype array can be created →

```
char ch[5] = {'a', 'b', 'c', 'b', 'b'};
```

```
double dou[5];
float flo[5];
bool boo[5];
```

- ⑦ Maximum & minimum in an array →

```
int size;
cin >> size;
```

int num[size];

टटीया Practice.

(Bcz of constraints)

```
int num[100]; // taking input
for (int i = 0; i < size; i++)
    cin >> num[i];
```

```
}
```

```
int getMin(int arr[], int n)
```

```
int min = INT_MAX;
for (int i = 0; i < n; i++)
    if (arr[i] < min)
```

```
    min = arr[i];
}
```

```
return min;
```

```
}
```

Date \_\_\_\_\_  
Page \_\_\_\_\_

```

int getMax(int arr[], int n)
{
    int max = INT_MIN;
    for (int i=0; i<n; i++)
    {
        if (arr[i] > max)
            max = arr[i];
    }
    return max;
}

```

5.1.1 b) OR using inbuilt function max(-, )

Var arr

```

int maxi = INT_MIN;
for (int i=0; i<n; i++)
{
    maxi = max(maxi, num[i]);
}

min = min(min, num[i]);

```

5.7.5 # Scopes in Arrays →

```

void update (int arr[], int n)
{
    cout << "Inside Function update" << endl;
    arr[0] = 120; // updating 0th element.
}
```

// printing the array.  
for (int i=0; i<n; i++)

{ cout << arr[i] << " ";  
} cout << endl;  
cout << "Going back to main function" << endl;
}

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```

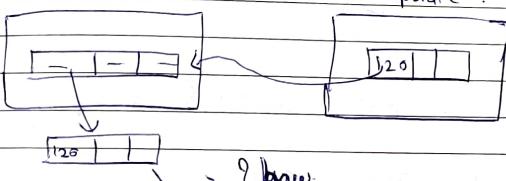
int main()
{
    int arr[] = {1, 2, 3};
    update(arr, 3);
    // printing the array.
    cout << "Printing in main function" << endl;
}

```

y.

Why arr[0] changed in main function

main()



(IMP) In case of variable these we'll be copy of that variable.

In case of arr[3] 

--	--	--	--

 the will share

the address arr → which can → add, remove, update  
100

we affect actual

int main arr .

④ In case of array base address will be passed to the function.

(H.W)

$\text{I/p} \rightarrow \text{size} \rightarrow 5$   
 $\boxed{1 \ 2 \ 3 \ 4 \ 5}$

$\text{o/p} \rightarrow \text{sum} \rightarrow 15$ , ...

print sum of all element in an array.

VScode Laptop

### {Linear Search} →

5	11	-5	4	17	Key = 4
0	1	2	3	4	
2	3	4	5	6	

bool LinearSearch(int arr[], int size, int key)

for (int i=0; i<size; i++)

{ if (arr[i] == key)

return true;

}

} return false;

}

int main()

int key;

<i>> key;

int array[] = {10, 12, 13, -128, 5, 8, 101, 17, 19, 20},

bool isfound = LinearSearch(array, 10, key);  
 if (isfound)

cout << "Element Found !!!" << endl;

else

cout << "Element not found !!!" << endl;

}

}

Reverse an array →

$\text{I/p} \rightarrow \text{arr} \rightarrow \{2, 7, 5, 9\}$   
 $\text{rev. arr} \rightarrow \{9, 5, 7, 2\}$

2 ↙ 3 4 5 9 ↘

9 ↙ 3 4 5 ↘ 2

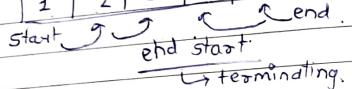
9 5 ↗ 4 3 2

9 5 4 3 2

Observation → arr odd len.  
arr even len.

(Ex1)

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



[start > end] ← terminating condition

Algo

→ swap → (Start wale ko, end wale ko)  
↓  
start++, end--

(Ex2)

1	2	3	4	5
s		e		

5	2	3	4	1
s		e		

5	4	3	2	1
s		e		

0	1	2	3	4
5	4	3	2	0

s > e. terminating condition,  
3 > 1

idhaun zuk jana.

अगर Start, end से बड़ा हुआ तो Stop\_swapping

void reverseArr(int arr[], int n)

{  
int start = 0;  
int end = n-1;  
while (start <= end)

    swap(arr[start], arr[end]);

// OR my swap code

// int temp = arr[start];

// arr[start] = arr[end];

// arr[end] = temp;

    start++;

    end--;

}

}

void printArr( )

{  
}

}

int main()

{  
int array[5] = {1, 5, 6, 2, 9};

reverseArr(array, 5);

}

# LECTURE - 10

## SOLVING LEETCODE / CODESTUDIO

### QUESTIONS [Arrays]

Q1) Swap Alternate.

① i/p → arr[5] = { 1, 2, 7, 8, 5 }  
                     ↑      ↑  
      o/p → { 2, 1, 8, 7, 5 }

② i/p → arr[6] → { 1, 2, 3, 4, 5, 6 }  
      o/p → { 2, 1, 4, 3, 6, 5 }

1	3	2	7	11	8
6	1	2	3	4	5

algo → for (index → 0 - 5), i+2 )  
 ↘

if (i+1 < size) // क्या अगला गला Index Range में आता है

swap (arr[i], arr[i+1])

3

1,

0 1 2 3 4

void swapAlternate(int arr[], int size)  
 ↘  
 for (i=0; i<size; i+=2)

swapping  
 if (i+1 < size)

Swap (arr[i], arr[i+1]);  
 ↘  
 ↘

3  
 ↘

breakdown:  
 Swap function →

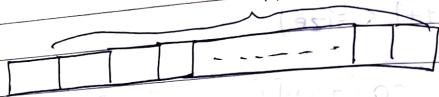
5	2
0	1

temp = arr[1]  
 arr[1] = arr[0].  
 arr[0] = temp.

## LEETCODE - Q.

136.

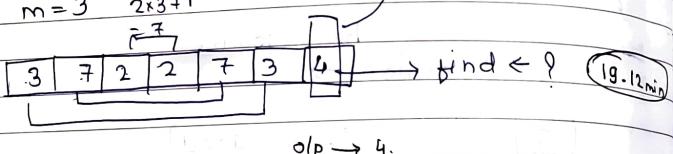
\* Find Unique Element →



"m" number → twice.

1 number → appear only once

$$m = 3 \quad 2 \times 3 + 1$$

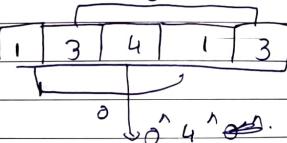


O/P → 4.



$$10 - 10 + 10 - 10 + 1$$

$$\begin{aligned} \text{XOR} \rightarrow a \wedge a &= 0 \\ 0 \wedge a &= a \end{aligned}$$



$$2^7 3^1 1^6 3^1 6^1 2^1 = 1$$

सारे के सारे Element को XOR कर देंगा.

```
int findUnique(int arr[], int size) {
```

```
    int ans = 0;
```

```
    for (int i = 0; i < size; i++) {
```

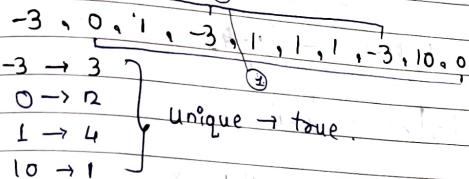
```
        cout << ans = ans ^ arr[i];
```

```
    return ans;
```

A.W.

## LEETCODE → Q.

Find unique no. of occurrences.



bool uniqueOccurrences(int arr[], int size)

int pivot = 0;

int counter = 0;

int prev = 0;

॥ श्री गणेशाय नमः ॥

Find Duplicates →



$[N=5]$

$[2, N-1] \rightarrow [1-5]$

Sol<sup>n</sup> →

$$\boxed{1 \ 2 \ 3} \times \underset{\text{sum}}{|} \underset{\text{N-1}}{+} \underset{\text{cancel}}{N-1} \quad \left\{ \begin{array}{l} a \leftarrow 0 \\ b \leftarrow 1 \\ c \leftarrow 0 \end{array} \right.$$

$1, 2, 3, \dots, x, \dots, N-1, x$

(A) Cancel Out

$\boxed{1, 2, 3, \dots, x, \dots, N-1}$

$$x^x = 0$$

$$0^x = x$$

Vector → dynamic array

```
int findDuplicate(int arr[], int size)
```

{

    int ans = 0;

    // XORing all array elements.

    for (int i = 0; i < size; i++)

}

    ans = ans ^ arr[i];

}

    // XORing [1, n-1]

    for (int i = 1; i < size; i++)

{

        ans = ans ^ i;

}

    return ans;

Leetcode .

442. Find all duplicates in an Array →

Code 360-

⊕ Intersection of two arrays →

Intersection → Common Element.

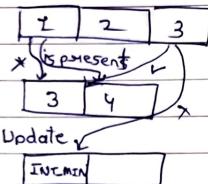
↪ Not present → ⊕ Ans.

पहले से हैं.

Sorted [non-dec order]

Eg.

{1, 2, 3} → ③  
{3, 4}



Check for every.

print - 3

↪ agar check nhi karna break

Sol → ① Check if present with  
② Update the value, INT-MIN. if present

③ Brute-force approach →

```
{  
vector<int> ans;  
for(int i=0; i<n; i++){  
    int element = arr1[i];  
    for(int j=0; j<m; j++){  
        if(element == arr2[j]){  
            ans.push_back(element);  
            arr2[j] = -12223;  
        }  
    }  
}  
return ans;
```

if (element < arr2[j])  
break;

Optimization-2 →

New code →

j=0

1	2	2	2	3	4
2	2	3	3		

j=0

arr1[i] < arr2[j]

j++;

arr1[i] < arr2[j]

↓

arr2[j] = arr2[j]

↓

print / vector

↓  
i++ / j++.

Code →

```
vector<int> ans;  
int i=0, j=0;  
while(i<n && j<m){
```

i (arr1[i] == arr2[j])

ans.push\_back(arr1[i]); i++; j++;

}

else if (arr1[i] < arr2[j])

<

i++;

} else {

j++;

}

return ans;

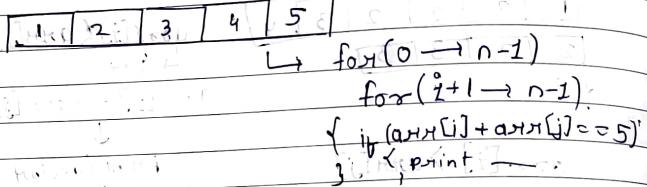
④ Pair sum →



$S = 5 \leftarrow$   
pair → ? sum

$$\begin{matrix} 1 & 4 \\ 2 & 3 \end{matrix}$$

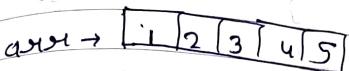
if  $a[i] + a[j] == S$   
 $i \neq j$



code →

```
vector<vector<int>> ans;
for(int i=0; i<arr.size(); i++) {
    for(int j=i+1, j<arr.size(); j++) {
        if (arr[i] + arr[j] == 5) {
            vector<int> temp;
            temp.push_back(min(arr[i], arr[j]));
            temp.push_back(max(arr[i], arr[j]));
            ans.push_back(temp);
        }
    }
}
sort(ans.begin(), ans.end());
return ans;
```

⑤ Triplet with given sum →



$K \rightarrow 12$

$$\text{Triplet} \rightarrow 5+4+3 \rightarrow 12.$$

for( $i \rightarrow h-1$ )

  ↳ for( $i+1 \rightarrow n-1$ )

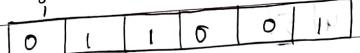
    ↳ for( $j+i \rightarrow n-1$ )

$\downarrow$   
 $\downarrow$   $= k$   
 $\downarrow$  print  
    }

⑥ H.W.

## 2 pointer approach

④ Sort 0 : 1



$$0|_p \rightarrow [0 | 0 | 0 | 1 | 1 |]$$

Sol's →

Counter → 0.

0 → 3

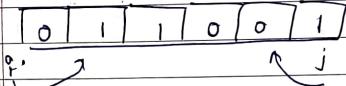
1 → 3

Traverse

0 | 1 .

Sort ()

n log n :



$\text{arr}[i] == 0$

$\hookrightarrow i++$

$\text{arr}[j] == 1$

$j--$

$\text{arr}[i] == 1 \& \text{arr}[j] == 0$

$\text{swap}(\text{arr}[i], \text{arr}[j])$

$i++;$

$j--;$

$i >= j \rightarrow \text{loop}$  रोक देना !!!

0 → left में रखना  
1 → right '1'

code →

```
void sortOneZero(int arr[], int n){  
    int left = 0;  
    int right = n-1;
```

while ( $\text{arr}[left] == 0 \&& left < right$ )

$left++;$

    } while ( $\text{arr}[right] == 1 \&& left < right$ )  
         $right--;$

}

अब यहाँ पहिले गये तीव्र इसका मतलब .

$\text{if } (\text{arr}[left] == 1 \text{ and arr}[right] == 0)$ .

$\text{if } (left < right)$ {

$\text{swap}(\text{arr}[left], \text{arr}[right]);$

$left++;$

$right--;$

}

}

};

④ Sort - 0, 1, 2.

(HW)

11.