

DSA

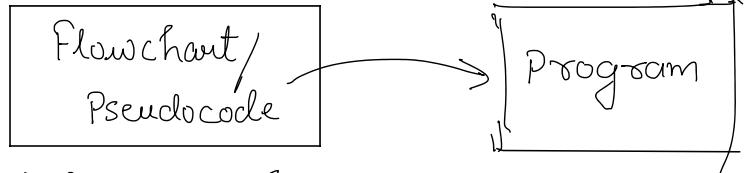
23 May 2024 15:01

→ getting started

- [P]
 - ↳ Understand
 - ↳ given values
 - ↳ Approach
 - ↳ Program

problem
solving

[P] → Sol'n

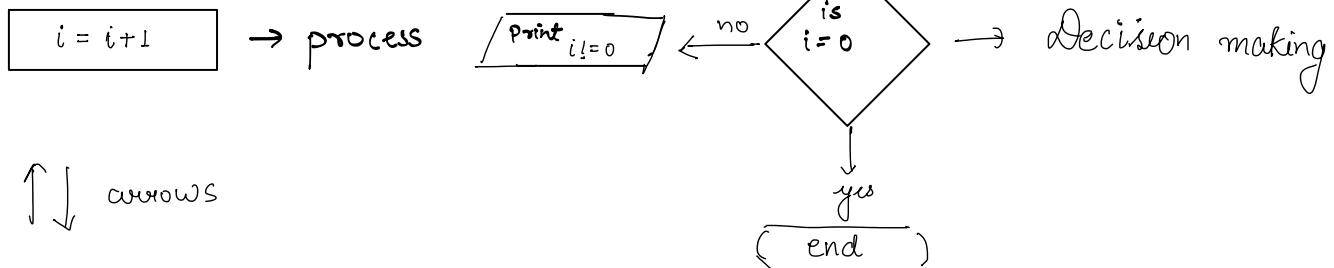


Jo hamne socha

→ Flowchart :- A diagrammatic approach

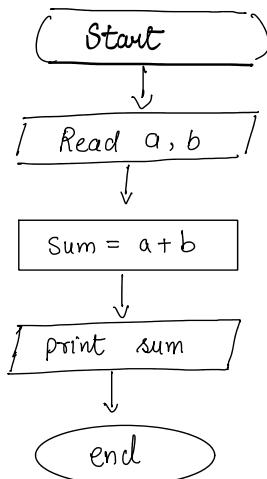
(Start / End) → Terminator
↙ ↘ start end

/ Read a / → Input / Output



(A) → connector

Example:- Sum of 2 numbers



Pseudocode

→ read a, b.

→ let sum = 0.

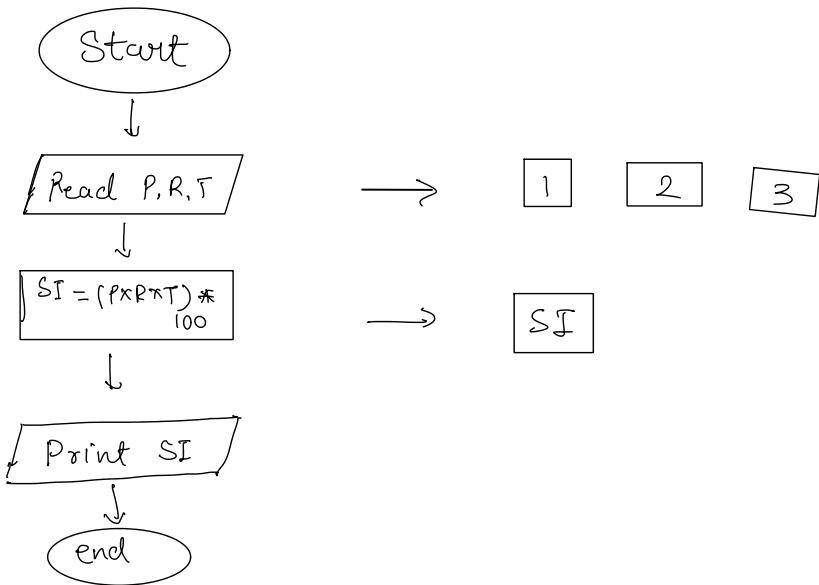
→ sum = a + b.

→ print sum

→ length code

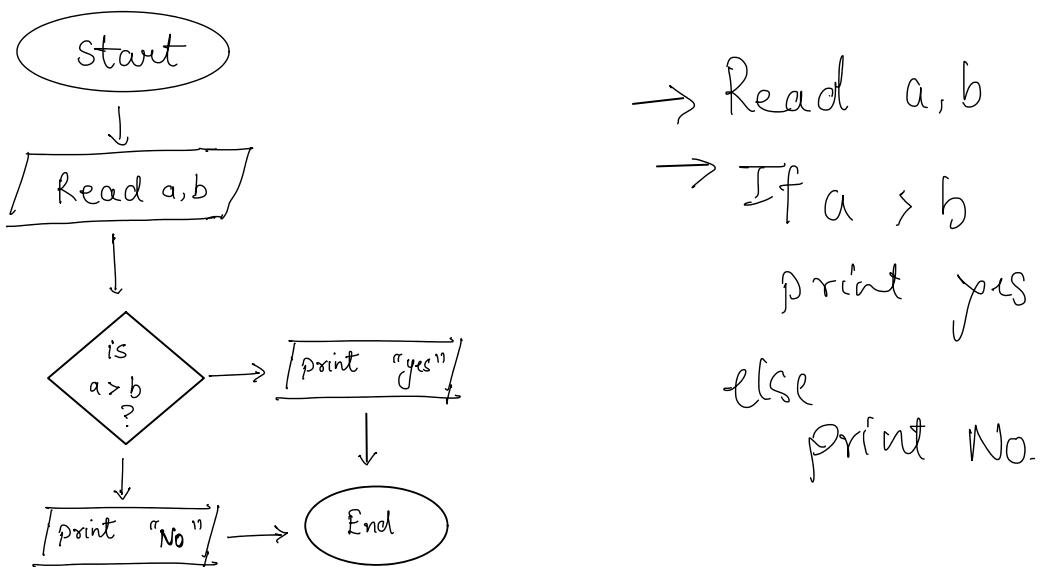
Ques: Write a program to get Simple Interest.

$$SI = (P \times R \times T) / 100$$



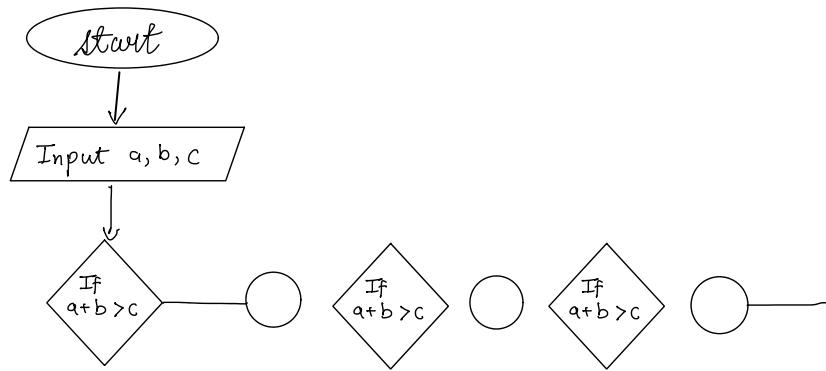
Ques:- $a > b$

Pseudo code



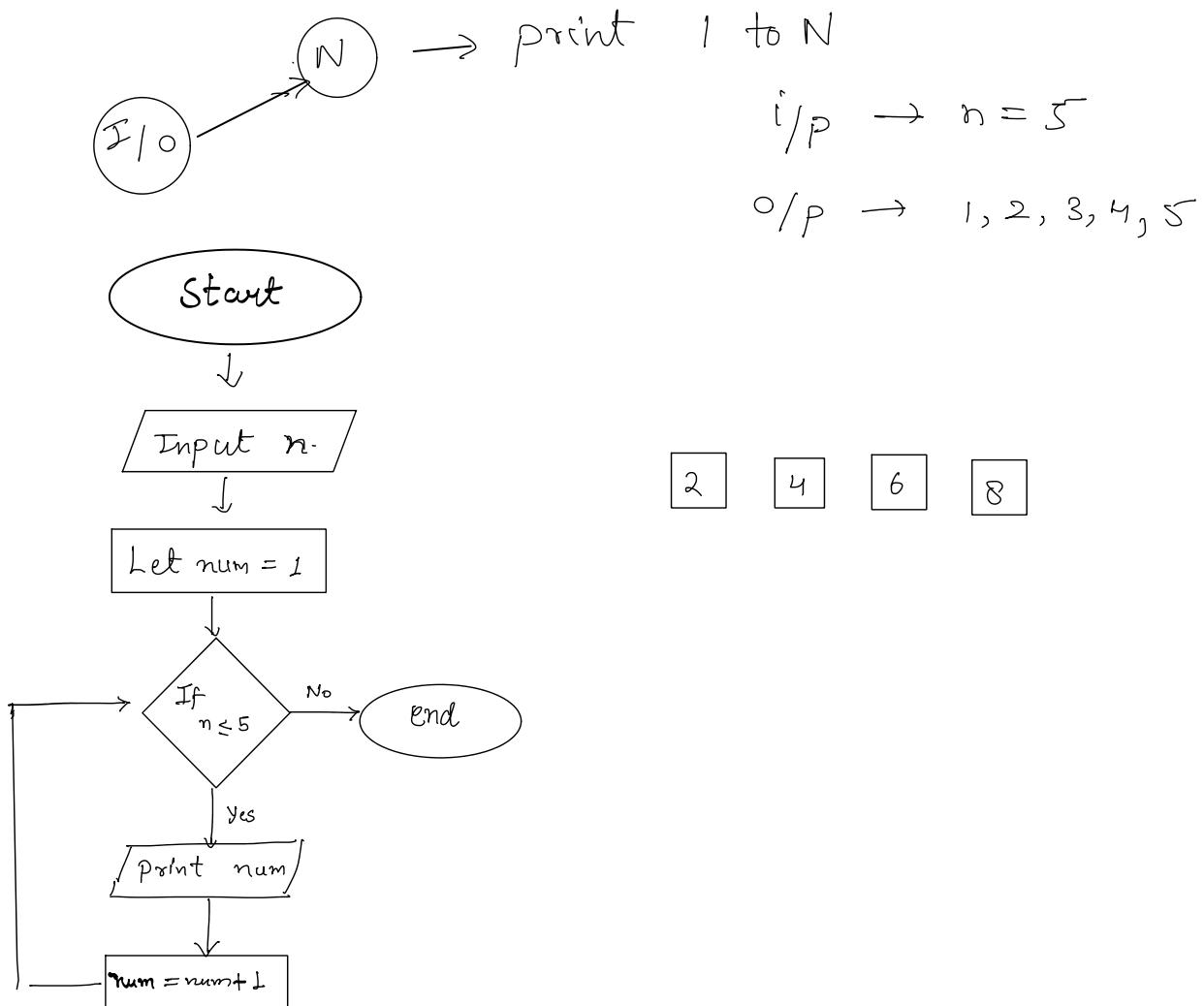
Ques:- Given triangle is legal or not.

Homework
Ques



Loops Loops

23 May 2024 15:01



for :- I to N $(I, N) \rightarrow$ Inclusive

Sum of n numbers

Homework

→ find factorial of n

$$5! = 5 \times 4 \times 3 \times 2 \times 1 \\ = 1 \times 2 \times 3 \times 4 \times 5$$

→ Check prime or not

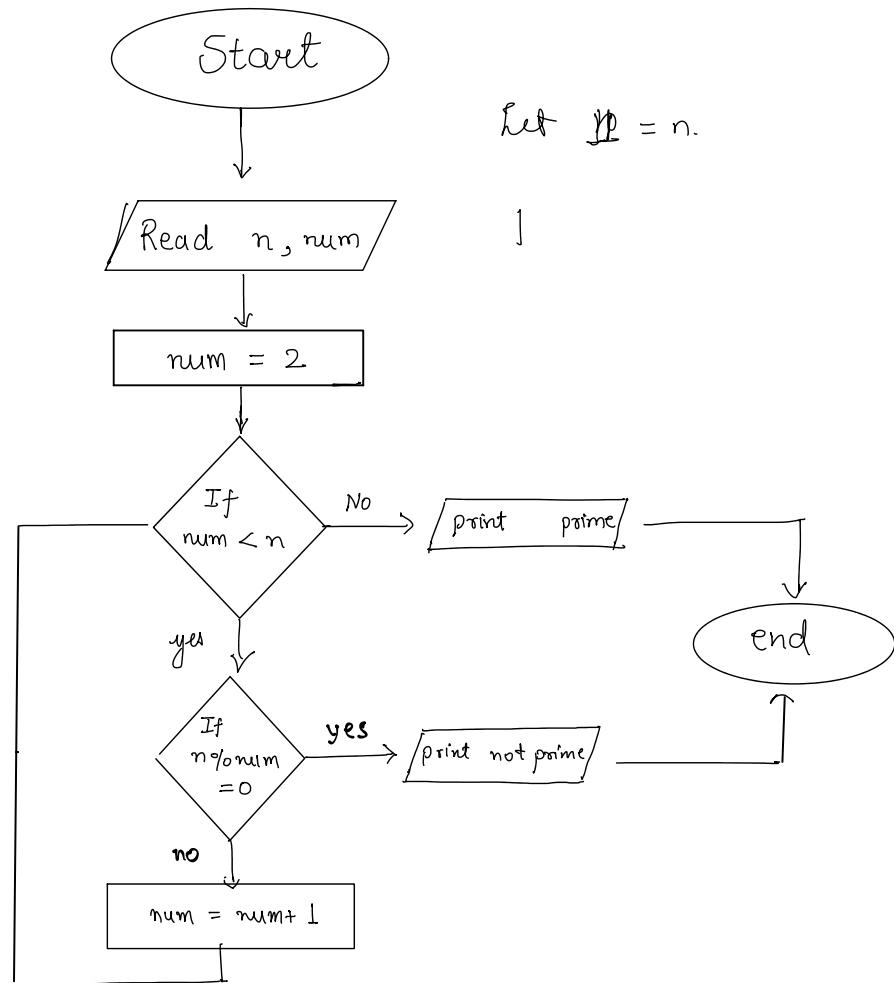
i/p → n

prime or n

ex:- i/p → 7 prime → 2, 3, 5, 7, 11, 13, 17
prime or not

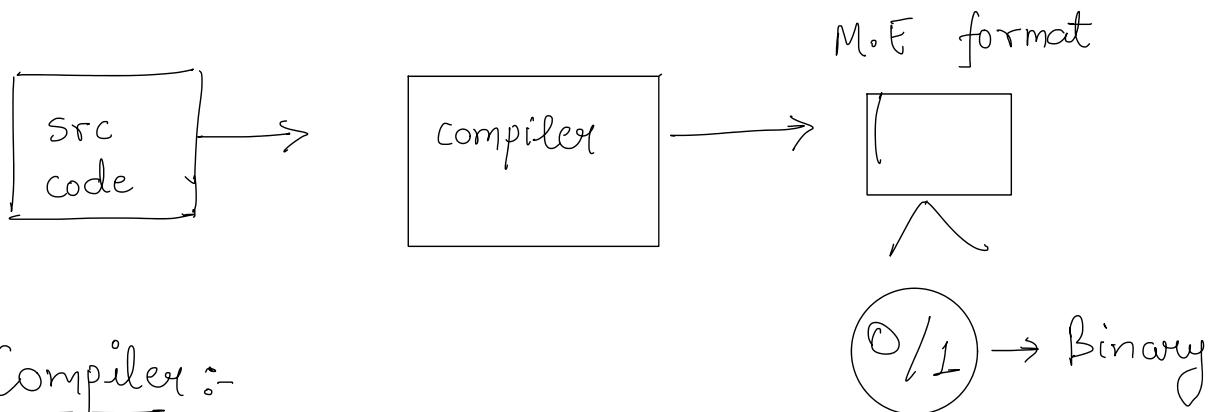
(n) → p or N

1 ----- n
Rem != 0



#2 Flowchart

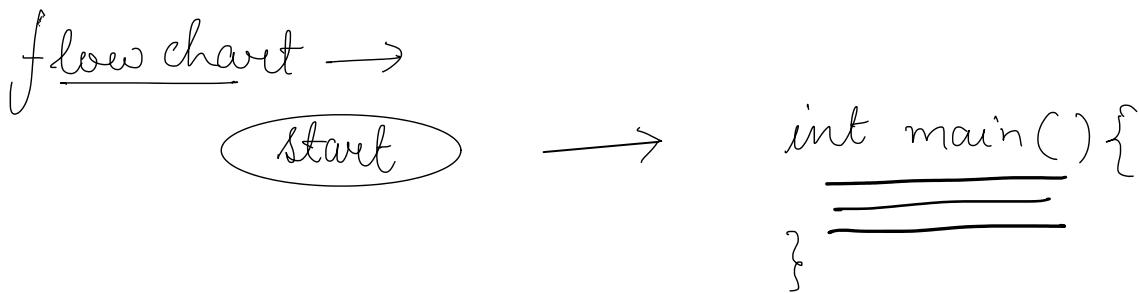
23 May 2024 15:01



Compiler :-

- ↳ Translation
- ↳ Errors

IDE :- VS Code



for printing →
'cout' is used

⇒ #include <iostream>
using namespace std;

int main (){
 cout << "Hello World" << endl;
}

"<<" → to write in std
(display operator).

cout << → output point.

cin >> → Input take

endl → new line

"\n" → also for new line

;" → to end line.

Data Types and Variables

Ex :- int a; → 4 bit → 32 bit

char for character

ex; Char ch = 'a'; Variable name

(iii) `bool` general convention

iv) float f = 1.2;

General conventions

v) double d = 1.23;

general convention

`Ex. - int size = sizeof(a);` → store the size of a

How data is stored

int a = 8 → Binary 1000

ooooo - - - - - - |oooo

32 bit

For char :- ASCII Value is used

The diagram shows a horizontal line representing memory. Below it, there are eight vertical tick marks, each pointing down to a small square box. This visualizes a single character being stored in an 8-bit byte.

`char = 'a';` $\xrightarrow{\text{ASCII} \rightarrow \text{int} \rightarrow \text{To Binary}}$

* Compiler use datatype to differentiate b/w both values.

Type Casting :-

int a = 'a'; → type casted to 97

char ch = 'Q' → type casted to 'b'

How -ve numbers are stored :-

First bit

- +ve 0
- -ve 1

Ex:- -ve no \rightarrow (-5)

S1 \rightarrow Ignore the -ve sign

S2 \rightarrow Convert into binary rep. — 101

S3 \rightarrow Take 2's complement and store

\Rightarrow Add 1 to binary value after 1's complement.

1's complement \rightarrow

1111111111.....010

2's comp. \rightarrow $+ 1$

$\begin{array}{r} 1111111111\dots\dots 010 \\ + 1 \\ \hline 1000000000\dots\dots 011 \end{array}$

shows
-ve

Range -2^{31} to 2^{31}

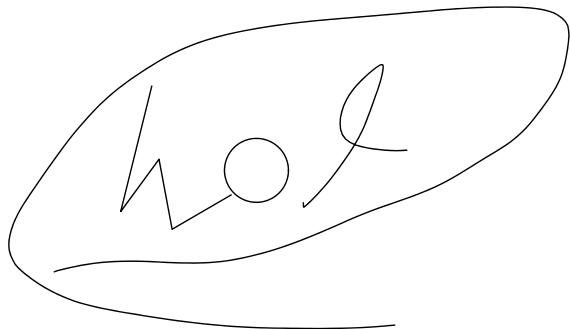
Default

int = 1, -1

* unsigned int a = 112; \rightarrow only store +ve values

Operators :-

int a = 215 i.e. 0



#3 Loops

23 May 2024 15:01

Conditionals and Loops

Ex:- if ($a > b$) {
 cout << "Answer is A";
}

Multiple inputs in CPP

cin >> a >> b;

press enter after every
value:-

cin doesn't read " " space
"\t" tab
"\n" enter

cin.get can be used to
get these values as input.

print numbers

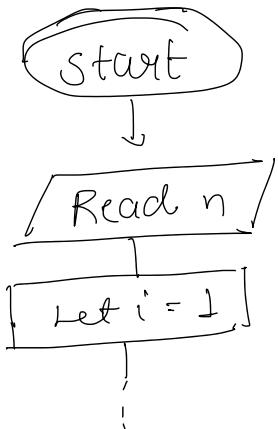
1 to N

→ while loop

while () {

 =====

}

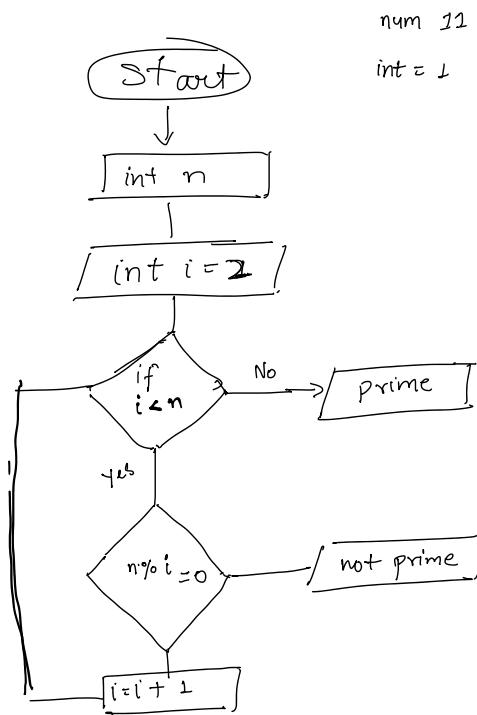


code examples that
i can do (easy)

Prime or Not Using Code

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, world!" << endl;
7     int n;
8     cin >> n;
9     int num = 2;
10
11    while(num<n){
12        if(num%n==0){
13            cout<<"Not prime"<<endl;
14        }
15        else{
16            cout<<"Prime"<<endl;
17        }
18        num = num+1;
19    }
20
21    return 0;
22 }
```



Pattern

$\begin{array}{cccc} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{array}$
} 4 row
4 column.

1st row \rightarrow 4
 2nd row \rightarrow 4
 3rd row \rightarrow 4
 4th row \rightarrow 4

```

int n;
cin>>n;
int i = 0;
//shifting through columns
while(i<n){
    int j=0;
    //printing rows
    while(j<n){
        cout<<"* ";
        j++;
    }
    cout<<endl;
    i++;
}
```

Pattern code

*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

output

Patterns

23 May 2024 15:01

Ques :-

1 1 1
2 2 2
3 3 3

```
// 2.2 Pattern
int n;
cin>>n;
int i = 1;
//shifting through columns
while(i<n){
    int j=1;
    //printing rows
    while(j<n){
        cout<<i<<" ";
        j++;
    }
    cout<<endl;
    i++;
}
```

1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4

output

code

Ques:-

3 2 1
3 2 1
3 2 1

n + i
3 0

```
// 2.3 Pattern
int n;
cin>>n;
int i = 1;
while(i<n){
    int j = 0;
    while(j<n){
        cout<<n-j;
        j++;
    }
    cout<<endl;
    i++;
}
```

5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1

Output

Que

J=1 J=2 J=3
1 2 3
4 5 6
7 8 9

print numbers
1 to n.

```
// 2.4 Pattern
// Counting
int n;
int count = 1;
cin>>n;
int i = 1;
while(i<n){
    int j = 0;
    while(j<n){
        cout<<count<<" ";
        count++;
        j++;
    }
    cout<<endl;
    i++;
}
```

1 2 3 4
5 6 7 8
9 10 11 12

output

Star Pattern

23 May 2024 15:01

Ques:-

* → row 1
 * * → row 2
 * * * → row 3
 * * * * → row 4

jumlah row utnne star.

```
// 2.7 Pattern
int n;
cin>>n;
int i = 1;
while(i<n){
    int j = 0;
    while(j<i){
        cout<<"* ";
        j++;
    }
    cout<<endl;
    i++;
}
```

```

*
* *
* * *
* * * *

```

output

code

Ques:-

1
 2 3
 4 5 6
 7 8 9 10

```
// 2.5 Pattern
int n;
int count = 1;
cin>>n;
int i = 1;
while(i<n){
    int j = 0;
    while(j<i){
        cout<<count<<" ";
        count++;
        j++;
    }
    cout<<endl;
    i++;
}
```

```

1
2 3
4 5 6
7 8 9 10

```

output

code

Ques:-

1
 2 3
 3 4 5
 4 5 6 7

```
// 2.6 Pattern
int n;
int count = 1;
cin>>n;
int i = 1;
while(i<n){
    int j = 0;
    while(j<i){
        cout<<count<<" ";
        count++;
        j++;
    }
    cout<<endl;
    i++;
    count=i;
}
```

```

1
2 3
3 4 5
4 5 6 7

```

output

code

Ques :-

2 1

3 2 1

4 3 2 1

ABC wala

EK koi

kar lewa

Ques :-

★

★ ★

★ ★ ★

★ ★ ★ ★

#5 Operators and For loop

23 May 2024 15:01

Operators & For loop :-

Bitwise Operators :-

AND $\rightarrow \&$

OR $\rightarrow |$

Not $\rightarrow \sim$

XOR $\rightarrow ^$

Left Shift :-

$\boxed{5 \ll 1}$ ↙
00000 - - - 00101 shift it by 1 bit
 $\Rightarrow 00000 - - - 01010$ i.e 10

$\boxed{3 \ll 2}$ \rightarrow 2 bar left shift

0 0 0 0 - - - 0 0 1 1
0 0 0 0 - - - 1 1 0 0 \Rightarrow i.e 12

$\rightarrow 01000 0010$ ↙
1 0 0 1 0 0 \rightarrow Isme 2 se multiply wala case nhi chalega.

Right Shift :-

$15 >> 1$ ↗
00000 - - - 00101 shift it by 1 bit
 $\Rightarrow 00000 - - - 00010$ i.e 10

```
for(int a = 0 , b = 1, c = 2; a>=0 && b>=1; a--, b-- ){  
    cout<<a <<" " <<b << endl;  
}
```

break;

to break the loop,
function etc.

continue;

to skip the iteration.

Variable and scopes

23 May 2024

15:01

Operator precedence

```
int a = 3  
cout << a << endl;  
  
if (true){  
    int a = 5;  
    cout << a << endl;  
}  
  
cout << a << endl;
```

⇒ Done

Ques Find the difference of sum and product of a number.

1. Take a integer. $a = 234$;
2. for loop laga ke % of integer by 10
 $\text{int } x = 234 \% 10$, $y = 234 / 10$ i.e 23
to get last value i.e 4
3. Now $\text{sum} = x + \text{sum}$; $\text{int sum} = 0$
 $\text{product} = 1 * \text{product}$; $\text{in product} = 0$
4. Now if condition ke ander dono ko daal do
and check it.

```
#include<iostream>  
using namespace std;  
  
int main(){  
    int a;  
    cin >> a;  
    int sum = 0;  
    int product = 1;  
  
    while(a != 0){  
        int x = a % 10;  
        int y = a / 10;  
  
        sum = x + sum;  
        product = x * product;  
        a = y;  
    }  
    cout << sum << " " << product << endl;  
    if(sum > product){  
        cout << sum - product;  
    }  
    else if(product > sum){  
        cout << product - sum;  
    }  
    else{  
        cout << 0;  
    }  
    return 0;  
}
```

Ques 191 LeetCode:

Ques 7 Leetcode :Reverse Integer

① int x; $x = 123$
while ($x \neq 0$)
char y = $x \% 10$; $y = 3$
sum = 0;

① Decimals & Binary

$$2 \rightarrow 10$$

$$5 \rightarrow 101$$

$$10 \rightarrow 1010$$

$$8 \rightarrow 1000$$

① $n = 10$

i) Divide by 2

ii) Store remainder

iii) Repeat above 2 steps until $n=0$

$$n = 10$$

$$\text{divisor} = 2$$

	Division	Remainder
	$10/2 \rightarrow 5$	0
iv) reverse stored remainder	$5/2 \rightarrow 2$	1
	$2/2 \rightarrow 1$	0
	$1/2 \rightarrow 0$	1

↑ reverse

②nd Approach

$$\boxed{n \rightarrow 5} \rightarrow 101$$

$$\begin{array}{r} \times \times \times \\ \& \downarrow \\ \hline \end{array}$$

$$n = 5$$

$$n \neq 0$$

$$\{ \quad \text{bit} = n \& 1;$$

$$n = n \gg 1; \rightarrow \text{right shift kardo.}$$

while ($n \neq 0$) {

$$\text{bit} = n \& 1$$

$$n = n \gg 1;$$

num & 1
↓ → 0 (Even)
↓ → 1 (Odd)

How to make a number:-

$$\text{ans} = 0;$$

$$\textcircled{1} \quad \text{ans} = (10^0 \times \text{digit}) + \text{ans}; \\ 10^0 \times 1 + 0$$

$$\textcircled{2} \quad \text{ans} = (10^1 \times \text{digit}) + \text{ans} \\ 10^1 \times 0 + 1$$

$$\textcircled{3} \quad \text{ans} = (10^2 \times \text{digit}) + \text{ans} \\ 100 + 1 \\ \Rightarrow 101$$

two make 123

$$\text{ans} = 0;$$

$$\text{ans} = \text{digit} \times 10^0 + \text{ans} \\ 1 \times 10^0 + \text{ans} \\ = 1$$

$$\text{ans} = \text{digit} \times 10^1 + \text{ans} \\ 2 \times 10^1 + 1 \\ = 21$$

$$\text{ans} = 3 \times 10^2 + 21 \\ 300 + 21 = 321$$

formula for reverse:-

$$(\text{digit} * 10^i) + \text{ans}$$

for printing straight \rightarrow

let num = 123, ans = 0;

$$\textcircled{1} \quad \text{ans} * 10 + \text{digit}$$

$$0 + 10 + 1 = 1$$

$$\textcircled{2} \quad \text{ans} * 10 + \text{digit} \\ 1 \times 10 + 2 = 12$$

$$\textcircled{3} \quad \text{ans} * 10 + 3 \\ 12 \times 10 + 3 = \underline{\underline{123}}$$

formula :- $\boxed{\text{ans} * 10 + \text{digit}}$

-ve num ko binary me krne ka code
+ve ko bhi phle

$$\begin{array}{r} 1000 \\ \swarrow \quad \searrow \\ 0 \times 10^0 + 0 \\ 0 \times 10 \end{array}$$

For positive integer :-

```
// Decimal to binary
int n;
cin>>n;
//for positive integer
int number = 0;
int i = 0;
while(n!=0){
    int bit = n&1;
    number = bit * pow(10,i) + number;
    // number = bit+number*10;
    i++;
    n = n>>1;
}
cout<<number<<endl;
```

For negative integer :-

Binary to Decimal

23 May 2024 15:01

$\begin{array}{r} 10101 \\ 2^4 2^3 2^2 2^1 2^0 \end{array} \rightarrow \text{Ignore zero and take } 1 \\ \Rightarrow 16 + 4 + 1 = \dots$

$n = 110$, int $i=0$

while ($n \neq 0$) {

 int digit = $n \% 10$;
 if (bit == 1) {
 ans = ans + something(2^i)
 }
 i++;

cout < ans;

* Isme agar & use
kiya to wo decimal
value pe kaam
karega

```
// Binary to Decimal
int n;
cin >> n;
int i=0;
int ans=0;
while(n!=0){
    // int bit = n&1;  Nhi krna hai
    int digit = n%10;
    if(digit==1){
        ans = ans+ pow(2,i);
    }

    n = n/10;
    i++;
}
cout << ans;
```

#7 Leetcode Questions

23 May 2024 15:01

```
// Leetcode problem 7(med).
int x;
cin>>x;
int num = 0;
while(x!=0){
    int digit = x%10;
    if(num>INT_MAX/10 || num<INT_MIN/10){
        return 0;
    }
    num = (num*10) + digit;
    x = x/10;
}
return num;
```

Ques:- Complement of Base 10.

num = 5
↳ to binary 101 } code

0000 - - - - > ... 0101

n = ~5
1111 - - - - . 0010

Mask → 0000 - - - - . 0111

Want
0000 0 - - - - , 0/0

(~n) & Mask ✓

int mask = 0

while →

(num != 0)
(0000 - - - - . 0000) | 1
(0000 - - - - . 111)

else

```
int m = n;
int mask = 0;
if(n==0){
    return 1;
}
while(m!=0){
    mask = (mask<<1)|1 ;
    m = m>>1;
}
int num = (~n)& mask;
return num;
```

Question 231 Leetcode

23 May 2024 15:01

```
int n;
cin>>n;
int count = 0;
while(n!=0){
    if(n&1){
        count++;
    }
    n= n>>1;
}
if(count==1){
    return true;
}
else{
    return false;
}
return 0;
```

My solution

→ ye ve
me prob dega
solve karne.

```
class Solution {
public:
    bool isPowerOfTwo(int n) {
        for(int i =0;i<=30;i++){
            int ans = pow(2,i);
            if(ans==n){
                return true;
            }
        }
        return false;
    }
};
```

Babbar

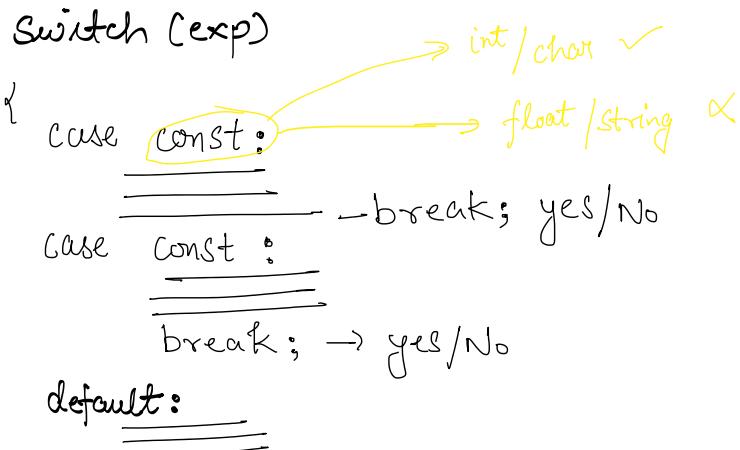
```
class Solution {
public:
    bool isPowerOfTwo(int n) {
        if(n<=0) return false;
        if( (n&(n-1))==0) return true;
        return false;
    }
};
```

Leetcode best

From <<https://leetcode.com/problems/power-of-two/>>

#8 Switch & functions

23 May 2024 15:01



Ques: while () {
 switch () {
 case:
 }
 → infinite loop
How to exit infinite loop
without break using exit().

* Continue is not valid in switch Case statement.

Calculator using switch case :-

```
int a, b;
cout << " Enter the value of a " << endl;
cin >> a;
cout << " Enter the value of b " << endl;
cin >> b;
char op;
cout << " Enter the Operation you want to perform" << endl;
cin >> op;
switch( op ) {
    case '+': cout << (a+b) << endl;
                 break;
    case '-': cout << (a-b) << endl;
                 break;
    case '*': cout << (a*b) << endl;
                 break;
    case '/': cout << (a/b) << endl;
                 break;
    case '%': cout << (a%b) << endl;
                 break;
    default: cout << "Please enter a valid Operation " << endl;
}
```

Rupee distribution

23 May 2024 15:01

Functions

23 May 2024 15:01

```
#include<iostream>
using namespace std;
int factorial(int n) {
    int fact = 1;
    for(int i = 1; i<=n; i++ ) {
        fact = fact * i;
    }
    return fact;
}
int nCr(int n, int r) {
    int num = factorial(n);
    int denom = factorial(r) *
factorial(n-r);
    return num/denom;
}

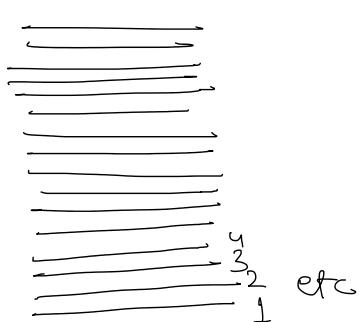
int main( ) {
    int n, r;
    cin>> n >> r ;
    cout <<" Answer is " << nCr(n,r)
<< endl;
    return 0;
}
```

```
void printCounting(int num) {                                Void for no return value
//    cout << n << endl;
    //Function Body
    for(int i=1; i<=num; i++) {
        cout<< i << " ";
    }
    cout<<endl;
}
int main() {
    int n;
    cin >> n;
    //function Call
    printCounting(n);

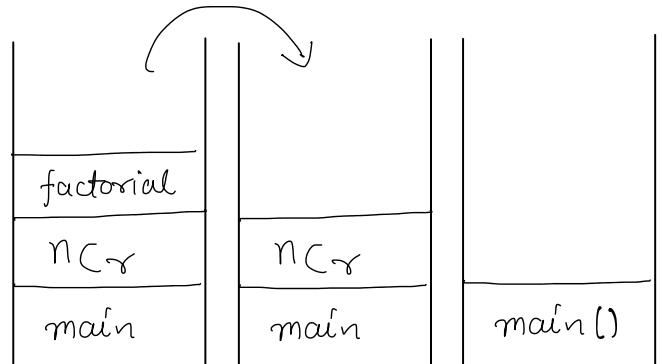
    return 0;
}
```

Function call Stack →

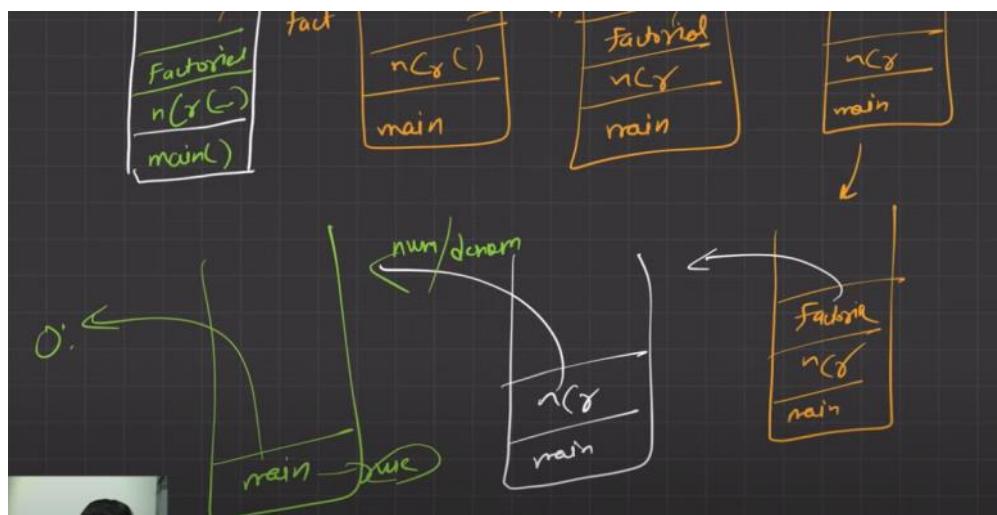
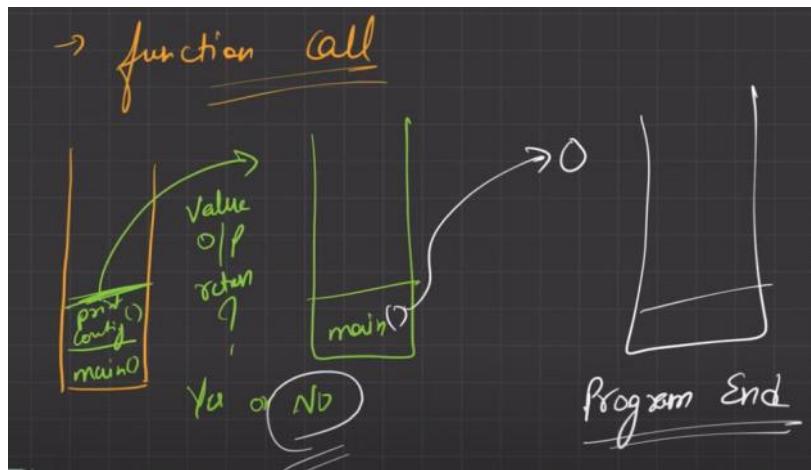
23 May 2024 15:01



ex:- factorial ques-



function call
stack

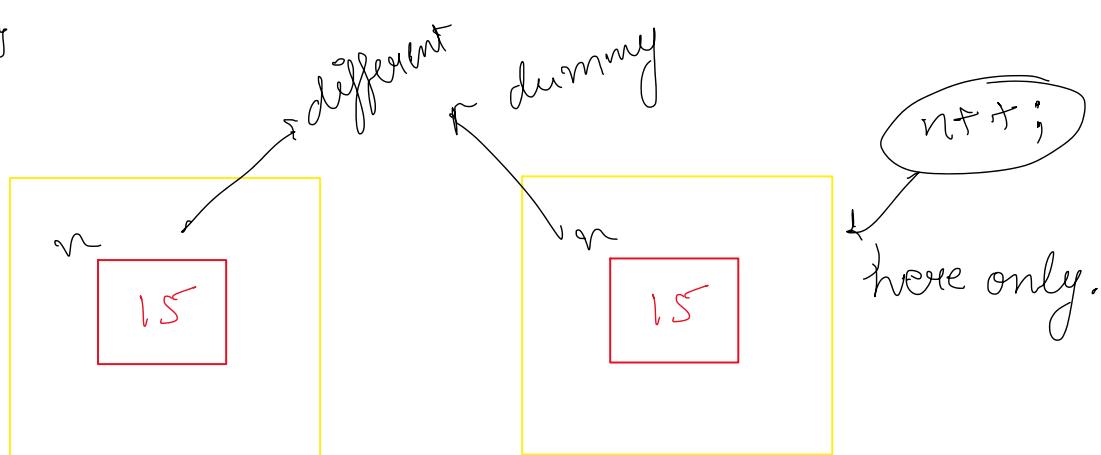


Pass by value

23 May 2024 15:01

```
void dummy(int n){  
    cout << "n is" << n << endl;  
}
```

```
int main(){  
    int n;  
    cin >> n;  
    dummy(n); → when this is called this value of n  
    cout << "number n is" << n << endl;       wont effect main  
    return 0;  
}
```

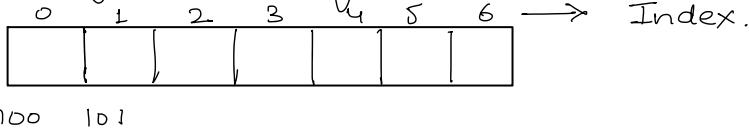


#9 Arrays

23 May 2024 15:01

⇒ Array is collection of same data types.

⇒ Contiguous memory location



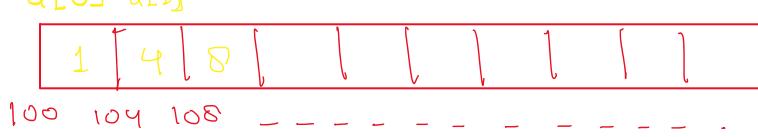
Address

⇒ To store multiple values in single variable.

Declarations-

⇒ `int a[10];` → 10 memory blocks.

`a[0] a[1]` name of array (also gives address of first location).



Integer takes 4 byte to store.

$$a[1] = 4$$

Initialization

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



`int arr[10000] = {0};`



* Only for zero

All have zero values.

* general

`int arr[size]; std::fill(arr, arr + size, 5);`

* If we put some values rest all index set zero by default.

Arrays with Function

* Like len Avi

```
#include<iostream>
using namespace std;
int get_Max(int arr[], int size){
    // int x = 0;
    int i;
    int max = arr[0];
    for(i=0; i<size; i++){
        if(arr[i] > max){
            max = arr[i];
        }
    }
    return max;
}
int get_Min(int arr[], int size){
    // int x = 0;
    int i;
    int min = arr[0];
    for(i=0; i<size; i++){
        if(arr[i] < min){
            min = arr[i];
        }
    }
    return min;
}

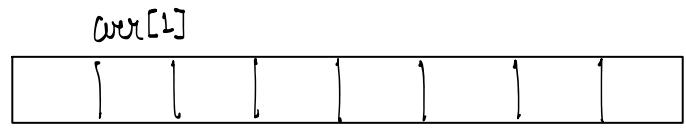
int main(){
    int n;
    cin >> n;
    int arr[5];
    for(int i=0; i<n; i++){
        cout << "Enter " << i << " number : ";
        cin >> arr[i];
    }
    for(int i=0; i<n; i++){
        cout << i << " number is ";
        cout << arr[i] << endl;
    }
    cout << "Maximum value of the array is " << get_Max(arr, n);
    cout << "Minimum value of the array is " << get_Min(arr, n);

    return 0;
}
```

Types of Array

- 1) int
- 2) char
- 3) bool
- 4) String
- 5) -----

Max & Min value of Array :-



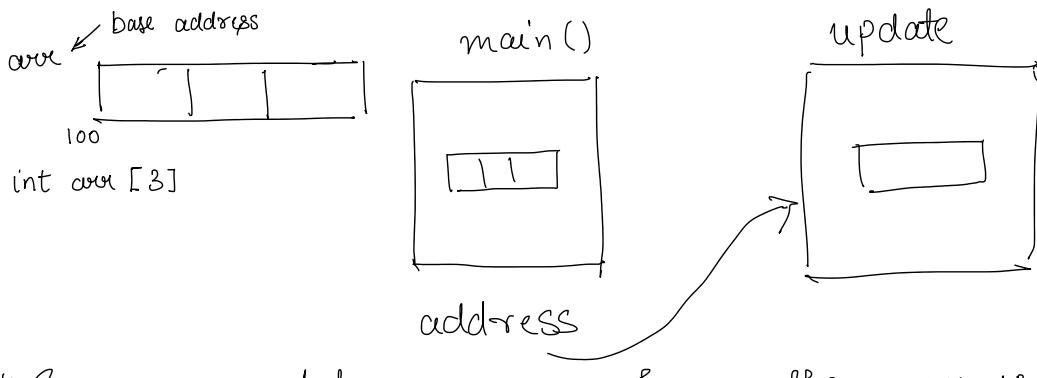
- `arr[0]`
- ① compare
② store larger value

Alternative :-

$\text{maxi} = \max(\text{maxi}, \text{num}[i]).$

↓

predefined
(max, min).



* So any update would change the array;

```
#include<iostream>
using namespace std;
void update(int arr[], int n) {
    cout << endl << "Inside the function" << endl;

    //updating array's first element
    arr[0] = 120;
    //printing the array
    for(int i=0; i<3; i++) {
        cout << arr[i] << " ";
    } cout<<endl;

    cout << "Going back to main function" << endl;
}

int main() {
    int arr[3] = {1,2,3};
    update(arr, 3);
    //printing the array
    cout<< endl << "Printing in main function" << endl;
    for(int i=0; i<3; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```

Sum Of array function:

```
int sumArray(int arr[], int size){
    int sum = 0;
    for(int i =0; i<size;i++){
        sum = sum + arr[i];
    }
    return sum;
}
```

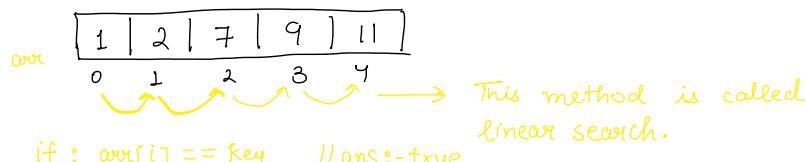
Linear Search

23 May 2024 15:01

```
int arr [10] = { };
```

// whether 1 is present in it or not.

```
arr[5] = {1, 2, 7, 9, 11};
```

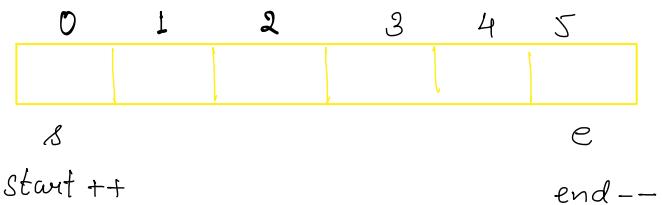


```
bool linearSearch(int arr[], int size, int key){  
    for(int i=0; i<size; i++){  
        if(arr[i]==key){  
            return true;  
        }  
    }  
    return false;  
}  
  
cout << linearSearch(arr, n, 7);
```

Reverse of an Array

```
int arr [] = {2, 7, 5, 9};
```

exchanging pairs



```
void reverseArray(int arr[], int size){  
    int start = 0;  
    int end = size - 1;  
  
    while(start <= end){  
        swap(arr[start], arr[end]);  
        start++;  
        end--;  
    }  
}
```

```
    }  
    while(s <= e){  
        exchange  
    }
```

Array Problems

23 May 2024 15:01

① Swap alternate

I/P → arr[5] = {1, 2, 3, 4, 5}

O/P → arr[5] = {2, 1, 4, 3, 5}

3	1	2	7	11	8
---	---	---	---	----	---

```
for(int i=0; i<n : i=i+2){  
    if(i+1 < n){  
        swap(arr[i], arr[i+1]);  
    }  
}
```

what does swap do :-

```
#include<iostream>  
using namespace std;  
  
void swapAlternate(int arr[], int size){  
    for(int i=0; i<size; i=i+2){  
        if(i+1 < size){  
            swap(arr[i], arr[i+1]);  
        }  
    }  
}  
void PrintingArray(int arr[], int size){  
    for(int i=0; i<size; i++){  
        cout << arr[i] << " ";  
    }  
}  
int main(){  
  
    int even[8]={5,2,9,4,7,6,1,0};  
    int odd[5] = {11,33,9,76,43};  
    swapAlternate(even, 8);  
    swapAlternate(odd, 5);  
    PrintingArray(even, 8);  
    cout << endl;  
    PrintingArray(odd, 5);  
  
    return 0;  
}
```

```
int x = 5;  
int y = 4;  
int temp;  
  
temp = x;  
x = y; // x = 4  
y = temp; // y = 5 } swapped
```

2

$$n = 2m+1$$



$m = \text{number} \rightarrow \text{twice}$

1 number \rightarrow appear once

ex:- [3 | 7 | 2 | 2 | 7 | 3] 4

↳ find(4) it appears one time.

Sol:- $a \wedge a = 0$

$$3 \wedge 3 \quad 7 \wedge 7 \quad 2 \wedge 2 = 0$$

left ans;

```
int gettingUniqueNum(int arr[], int size){
    int ans = 0;
    for(int i=0; i<size; i++){
        ans = ans ^ arr[i];
    }
    return ans;
}
```

```
int mirror[7] ={5,3,4,4,5,3,1};
cout<<gettingUniqueNum(mirror, 7)<<endl;
```

* XOR Mostly used
for sorted array &
continuous arrays.

3
Meth
Mine

```
for(i=0; i<n; i++){
    for(j=0; j<i; j++){
        if(arr[i] == arr[j]){
            return true;
        }
    }
}
{ 1, 2, 2, 3, 4 }^ 0 = ①
```

$O(n^2) \Rightarrow$ Double
linear search

* 0 to n hai jo
* ki linear hai

Meth2

Teacher

Sabka XOR hogya
ek bar \rightarrow { } \rightarrow Same value zero
for (ans = ans ^ arr[i];) {
 Ex:- {
 }
* left values \rightarrow

$[i] \wedge \text{ans}$ given ans

$O(2n)$

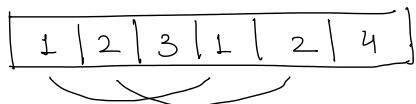
```
int findDuplicate(int arr[], int size){
    int ans = 0;
    for(int i=0; i<size; i++){
        ans = ans ^ arr[i];
    }
    for(int i=0; i<size; i++){
        ans = ans ^ i;
    }
    return ans;
}
```

```
int arr[6] ={1,2,3,5,4,5};
cout<<findDuplicate(arr, 6);
```

Que - 442

Leetcode

(4)



```

for (           )
    for (           ) {
        arr[j] = arr[i]
    }
for

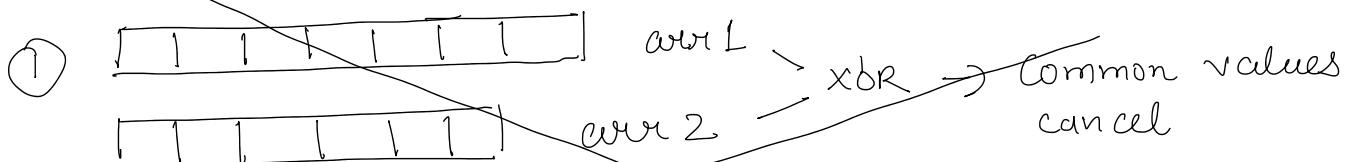
```

Que :- 2 sorted arrays (In ascending)

A & B

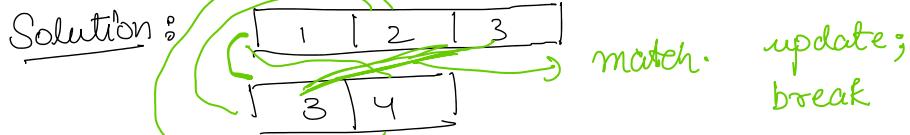
size N & M

If no intersection return '-1'



② for $[arr1 \sim i]$ get the value

③ arr ek if condition for '-1'



Brute force

Previous question with 2 pointer approach:

[1 | 2 | 2 | 2 | 3 | 4 |]

[2 | 2 | 3 | 3 |]

whole in
while loop

{ if ($\text{arr}[i] < \text{arr}[j]$)
 $i++$
 if $\text{arr}[i] == \text{arr}[j]$
 update;
 $i++, j++$
 if $\text{arr}[i] > \text{arr}[j]$
 $j++$

Initialization:

```
std::vector<int> vec = {1, 2, 3, 4, 5};  
Accessing Elements:
```

```
int value = vec[2];           // Access element at index 2  
int value = vec.at(2);        // Access element with bounds checking  
Modifying Elements:
```

```
vec[2] = 10;                  // Modify element at index 2
```

Iterating Over Elements:

```
cpp  
Copy code  
for (int elem : vec) {  
    std::cout << elem << " "  
}  
std::cout << std::endl;
```

Getting Size:

```
cpp  
Copy code  
int size = vec.size();  
Checking if Empty:
```

```
cpp  
Copy code  
bool isEmpty = vec.empty();  
Adding Elements:
```

```
cpp  
Copy code  
vec.push_back(6);           // Add element at the end  
Inserting Elements:
```

```
cpp  
Copy code  
vec.insert(vec.begin() + 2, 99); // Insert 99 at index 2  
Deleting Elements:
```

```
cpp  
Copy code  
vec.erase(vec.begin() + 2);    // Remove element at index 2  
Clearing the Vector:
```

```
cpp  
Copy code  
vec.clear();                 // Remove all elements  
Resizing the Vector:
```

```
cpp  
Copy code  
vec.resize(10);              // Resize vector to contain 10 elements  
Swapping Contents:
```

```
cpp  
Copy code  
std::vector<int> vec2 = {7, 8, 9};  
vec.swap(vec2);              // Swap contents of vec and vec2
```

Accessing Front and Back:

```
int front = vec.front();      // Access first element
int back = vec.back();       // Access last element
```

Reserving Capacity:

```
vec.reserve(20);             // Reserve capacity for at least 20 elements
```

Summary

Static Arrays: Fixed size, manual management required **for** operations like insertion and deletion.

std::vector: Dynamic size, many built-in methods **for** easy array manipulation.

Here's a complete example **using** `:std::vector`

cpp

Copy code

```
#include <iostream>
#include <vector>
void printVector(const std::vector<int>& vec) {
    for (int elem : vec) {
        std::cout << elem << " ";
    }
    std::cout << std::endl;
}
int main() {
    std::vector<int> vec = {1, 2, 3, 4, 5};
    // Add element at the end
    vec.push_back(6);
    printVector(vec);
    // Insert element at index 2
    vec.insert(vec.begin() + 2, 99);
    printVector(vec);
    // Remove element at index 2
    vec.erase(vec.begin() + 2);
    printVector(vec);
    // Access and modify element
    vec[2] = 10;
    printVector(vec);
    return 0;
}
```

#11. Time & Space Complexity

Time Complexity

23 May 2024 15:01

- It is the amount of time taken by an algorithm to run.
- as a function of length input.

Why? → For making better programs
 ↳ Comparison of Algo.

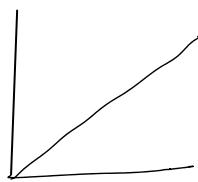
→ Big O notation → Theta Θ → Omega Ω
 Upper Bound for avg-case complexity ↓ Lower Bound

→ Constant time $\rightarrow O(1)$ → `for (i=0 → 10) { "Hello" }`
 → Linear time $\rightarrow O(n)$ → `for (i=0 → n { " " } // Input dependent.`

→ Logarithmic time $\rightarrow O(\log n)$ → Binary Search

→ Quadratic time $\rightarrow O(n^2)$
 → Cubic time $\rightarrow O(n^3)$

→ $O(n)$



$O(N!)$ — Highest.

$O(2^n)$

$O(N^3)$

$O(N^2)$

$O(N \log N)$

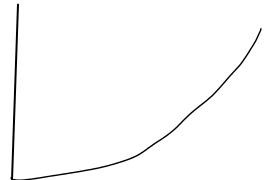
$O(N)$

$O(\log N)$

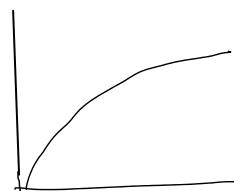
$O(1)$

↳ Least

→ $O(n^2)$



→ $O(n \log n)$



Question :-

→ lower degree & constants are ignored

$$f(n) = \frac{2n^2 + 3n}{n} \Rightarrow O(n^2)$$

$$f(n) = 4n^4 + 3n^3 \Rightarrow O(n^4)$$

$$f(n) = N^2 + \log N \Rightarrow O(n^2)$$

$$f(N) = 12001$$

$$f(n) = 3n^3 + 2n^2 + 5 = O(n^3)$$

$$f(n) = \frac{n^3}{300} \quad O(n^3)$$

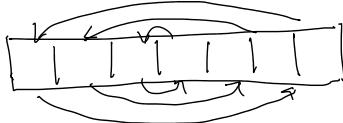
$$f(n) = 5n^2 + \log n \quad O(n^2)$$

$$\begin{array}{ll} f(n) = n/4 & O(n) \\ f(n) = \frac{n+4}{4} & O(n) \end{array} \quad \left. \begin{array}{l} \\ \end{array} \right\} \rightarrow \text{linear time}$$

Example :- i) printing of array →.

$\Rightarrow O(n)$ to n iteration $\Rightarrow O(n)$

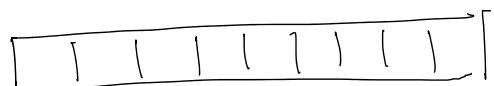
ii) Swapping



g) 0 to n length over

$\Rightarrow n/2$ swaps so, $O(n)$

iii) Linear Search :



i) 0 to n so (n comparisons)

$\Rightarrow O(n)$

* for (0 to n) {

$O(n)$

```

for (0 to n) {
    for(0 to m) {
    }
}

```

$O(n)$

$\overbrace{\hspace{10em}}$ $O(m)$

}

So $\hat{A} \in O(n \times m)$.

Steck in TLE

10^8 operation rule \rightarrow Most modern machine can perform upto 10^8 operations/sec

If constraints given

$1 < n < 10^6$
$1 < n < 1000$

	<u>Time Complexity</u>
$\leq [10 \dots 11]$	$O(n!)$, $O(n^6)$
$< [15 \dots 18]$	$O(2^n * n^2)$
< 100	$O(n^4)$
< 400	$O(n^3)$
< 2000	$O(n^2 * \log n)$
$< 10^4$	$O(n^2)$
$< 10^6$	$O(n \log n)$
$< 10^8$	$O(n)$, $O(\log n)$

* Learn the table for constraints.

Space Complexity :-

\Rightarrow Memory that is used by program

① $\text{int } a, \text{int } b;$ $\rightarrow O(1)$

* Variable is always $O(1)$

② $\text{func } ()$

{ $\text{int arr[5]} = \{1, 2, 3, 4, 5\}$

* Fixed size array

$\Rightarrow O(1)$

}

③ $\text{int } n;$

$\text{cin};$

$\text{vector<int>} v(n);$

$\rightarrow S.C \rightarrow O(n)$

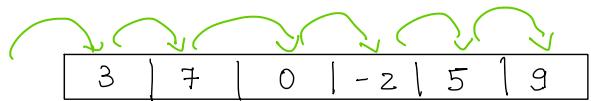
$\hookrightarrow \text{length}(n)$

* For loop memory nhi khata time khata hai until usme i/p na do.

#12. Binary Search

Binary Search

⇒ Linear Search:-

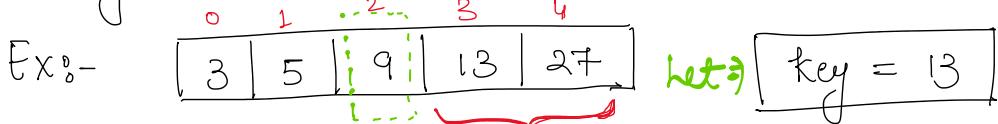


$n \rightarrow$ comparison.

* worst case n
when element not present -

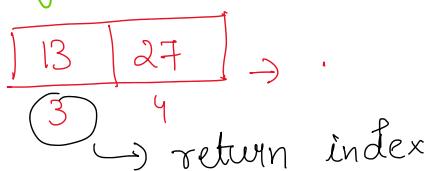
Binary Search:- Element should be in increasing /
Sorted OR Monotonic OR decreasing order

* Binary search me mid value find karte hai.

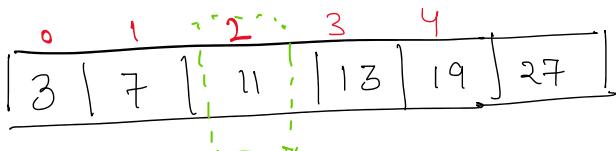


↳ Compare with key

⇒ 13 is greater than 9



New Search 27

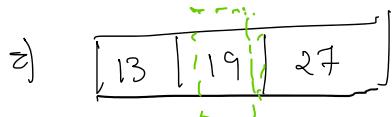


$$\text{mid} = \left(\frac{s+e}{2} \right) = \frac{0+5}{2} = \frac{5}{2} = 2$$

key = 27

11 = 27

27 > 11



19 = 27
27 > 19

$$\text{mid} = \frac{5+5}{2} = 5$$

z) $\text{mid} \left(\frac{3+5}{2} \right) = \frac{8}{2} = 4$

27

return index 8-5

Optimization

$$\text{mid} = \left(\frac{\text{start} + \text{end}}{2} \right) \quad \text{int} \rightarrow 2^{31} - 1$$

$$\begin{aligned} \text{start} &= 2^{31} - 1 \\ \text{end} &= 2^{31} - 1 \end{aligned}$$

value

$$\Rightarrow \left\{ \begin{array}{l} \text{mid} = s + \left(\frac{e-s}{2} \right) \\ \text{mid} = s + \frac{e}{2} - \frac{s}{2} \\ \text{mid} = \frac{s}{2} + \frac{e}{2} = \left(\frac{s+e}{2} \right) \end{array} \right.$$

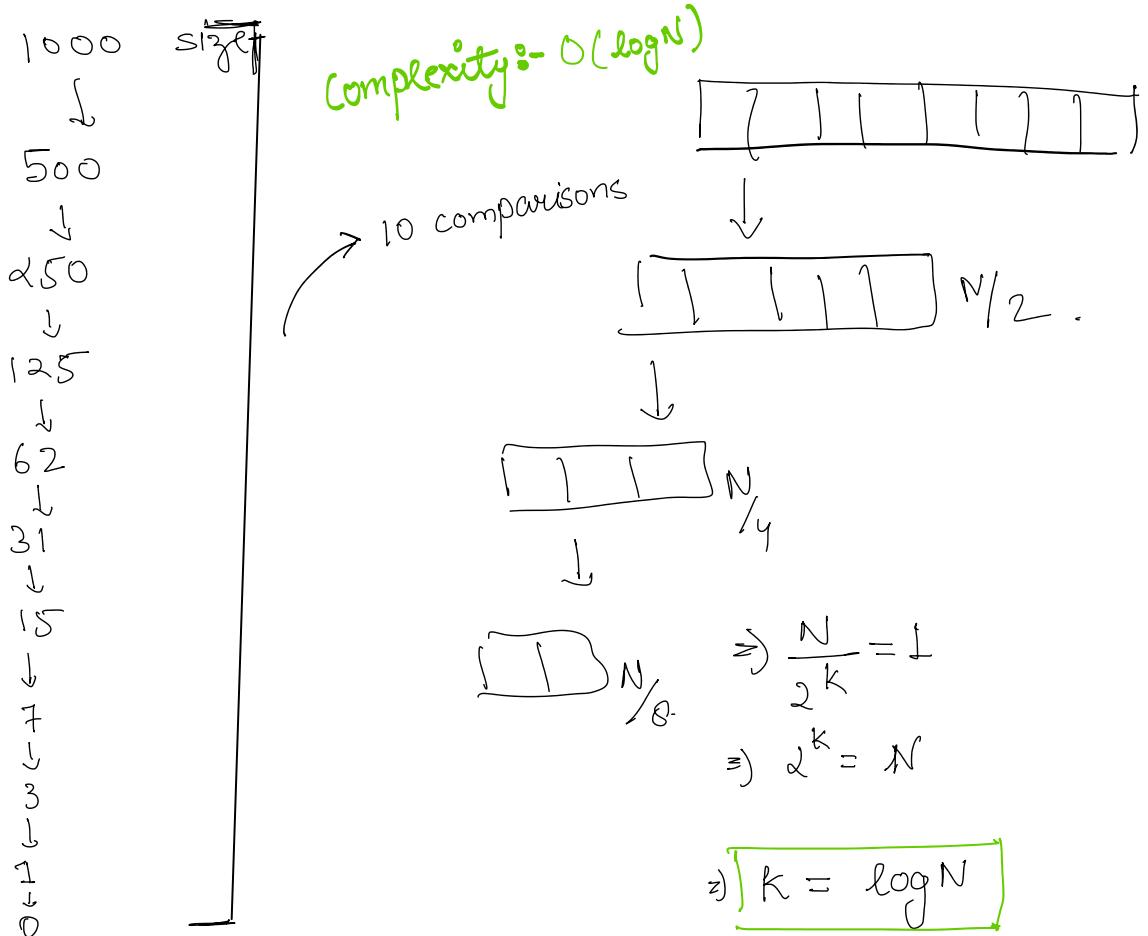
→ To remove run time error for larger integers.

Linear Search vs Binary Search

for Sorted array →

L.S → 1000 values — array
Worst case → 1000 comparisons

Now, B.S → 1000 value → sorted array.

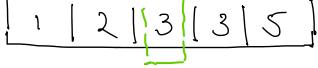


#13 Binary Search Questions

B.S Questions

- ① First and last position of K element in N elements.
Sorted Array Given

Eg: 0 1 1 2 2 2 2
Return index of both.

Sol:-  key = 3

mid = $\frac{\text{start} + \text{end}}{2}$
3 = 3 = true
store it

Now left most occurrence :-

```
if (arr[mid] > key)
    end = mid - 1;
if (arr[mid] < key) {
    start = mid + 1;
```

return int index.

durra function int ans = 0;
 \downarrow For last occurrence

```
if (arr[mid] == key)
    ans = mid;
    s = mid + 1;
if (arr[mid] > key){
    end = mid - 1;
if (arr[mid] < key) {
    start = mid + 1
}
```

Now pair these integers.

return int index.

* Pair is a datatype

```
pair<int, int> p;
p.first = func1(____);
p.second = func2(____);
return p;
```

```
int firstOcc(vector<int>& arr, int n, int key) {
    int s = 0, e = n-1;
    int mid = s + (e-s)/2;
    int ans = -1;
    while(s <= e) {
        if(arr[mid] == key) {
            ans = mid;
            e = mid - 1;
        }
        else if(key > arr[mid]) { // Right me jao
            s = mid + 1;
        }
        else if(key < arr[mid]) { // Left me jao
            e = mid - 1;
        }
        mid = s + (e-s)/2;
    }
    return ans;
}
```

```
int lastOcc(vector<int>& arr, int n, int key) {
    int s = 0, e = n-1;
    int mid = s + (e-s)/2;
    int ans = -1;
    while(s <= e) {
        if(arr[mid] == key) {
            ans = mid;
            s = mid + 1;
        }
        else if(key > arr[mid]) { // Right me jao
            s = mid + 1;
        }
        else if(key < arr[mid]) { // Left me jao
            e = mid - 1;
        }
        mid = s + (e-s)/2;
    }
    return ans;
}
```

② Total no. of occurrences of key.

Meth 1: Make both functions.

First Index | Last Index.
Better

Formula : (Last Index - First Index) + 1

OR

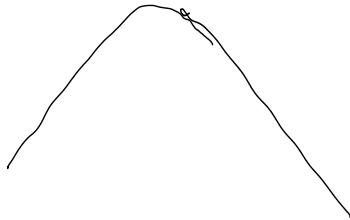
M-2 Get First Index by function →

then :- if in side if:-

1 1 2 1 3 1 3 1 5
l not equal

int count = 0
count ++;

Peak element in mountain array :-



0 1 1 0, 0 2 1 1 0



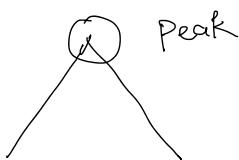
0 1 0 1 5 1 2

3 1 4 1 5 1 1

Max → L.S
→ O(n)

Que: Find maximum element:-

if (arr[i] > ans){
ans = arr[i];
}



Now, with B.S

0 1 2 4 3 1

$$\frac{s+e}{2} = 3$$

if (arr[mid] < arr[mid+1]) {

x = arr[mid];

else {
end = mid;

return arr[mid];

store it.

```
int peakElement(int arr[], int size){  
    int s = 0;  
    int e = size - 1;  
    int mid = s + (e-s)/2;  
    int x = 0;  
    while(s <= e){  
        if(arr[mid] < arr[mid+1]){  
            s = mid+1;  
        }  
        else{  
            e = mid;  
        }  
        mid = s + (e-s)/2;  
        x = arr[s];  
    }  
    return x;  
}
```

14 Binary Search (More Questions)

① Find Pivot. of an array.

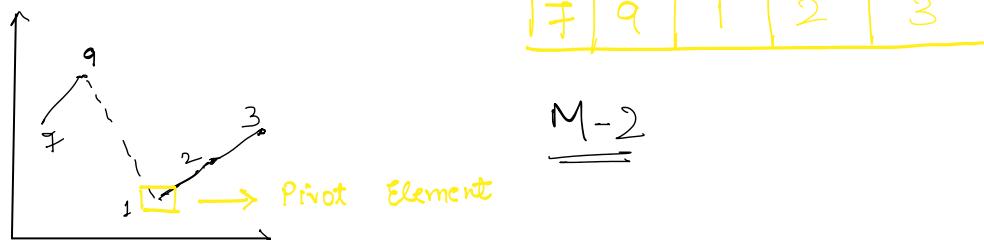
i/p \rightarrow arr[] $\rightarrow \{1, 2, 3, 7, 9\} \rightarrow$ [sorted]

$\xrightarrow{\text{if}} \{7, 9, 1, 2, 3\} \rightarrow$ [rotated]

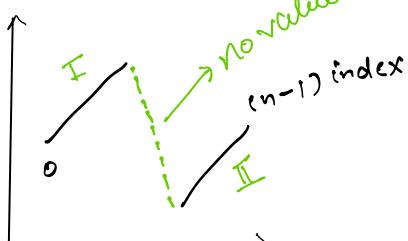
i/p Pivot $\rightarrow ?$

M-1) L-I

O(n)



no value in this line

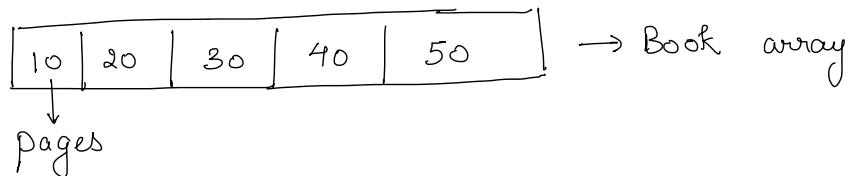


$mid = s + (e-s)/2$
 $arr[mid]$
 I: $\{ \text{if } \text{arr}[mid] > \text{arr}[0] \}$
 $s = mid+1$
 }
 else {
 $e = mid;$

```
int pivotElement(int arr[], int size){
    int s = 0;
    int e = size-1;
    int mid = s + (e-s)/2;
    while(s < e){
        if(arr[mid] > arr[0]){
            s = mid+1;
        }
        else{
            e = mid;
        }
    }
    return s;
}
```

Book allocation

Book Allocation problem



No. of students m.

1. Continuous book
2. maximum number of pages assigned to student is minimum.