

(Part - 1st)

(DSA in C++)

# INDEX

Name: Tushar Singh Rawat Std: \_\_\_\_\_ Sec: \_\_\_\_\_ Roll No: \_\_\_\_\_

S.No.	Date	Title	Pg. No.	Remark
(*)		DSA in C++ ( <u>30 feb 2023</u> ) Start along with C++ Concepts   Phase - 1 *		Part 1
(*)		DSA in C++ (Phase-2)   Phase - 2 * ( <u>6 May 2023</u> ) End		Part 2

# Detail Index

- (\*) Getting started → flow chart
- (\*) Pattern questions
- (\*) Bitwise manipulation & questions
- (\*) Fibonacci & factorial understanding
- (\*) Basic array questions
- (\*) Time complexity, Space complexity
- (\*) Search Space
- (\*) Selection sort
- (\*) Bubble sort
- (\*) Insertion sort
- (\*) Basic C++ STL
- (\*) Merge two sorted array
- (\*) All about Pointers
  - (\*) Subscript & offset Notation
  - (\*) Pointer to constant
  - (\*) Constant Pointer to constant
  - (\*) Reference
  - (\*) L value and R value
- (\*) Rotate an array
- (\*) Leetcode imp questions for array, String and basic use of pointers
- (\*) Leetcode - wave matrix point, Spiral point  
Search 2D matrix - I, search 2D - II
- (\*) Sieve of Eratosthenes
- (\*) GCD / HCF by Euclidean algo
- (\*) Modulo operator properties
- (\*) Heap & Stack memory understanding
- (\*) Dynamic memory Allocation
- (\*) Basic keywords and concepts

## (\*) Recursion

(\*) Stair problem

(\*) Fibonacci, factorial, count n no.

## (\*) Sorting using Recursion

(\*) Linear sort, Binary sort

## (\*) Check for Palindrome

(\*) Bubble sort

(\*) Merge sort

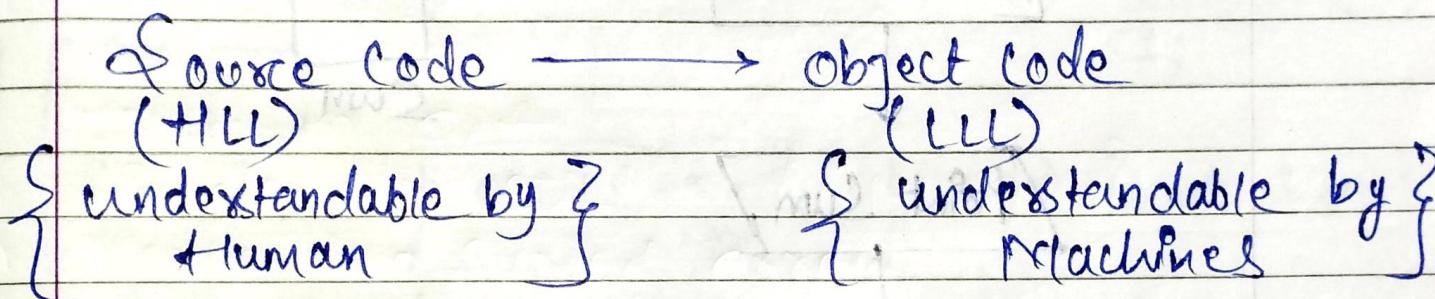
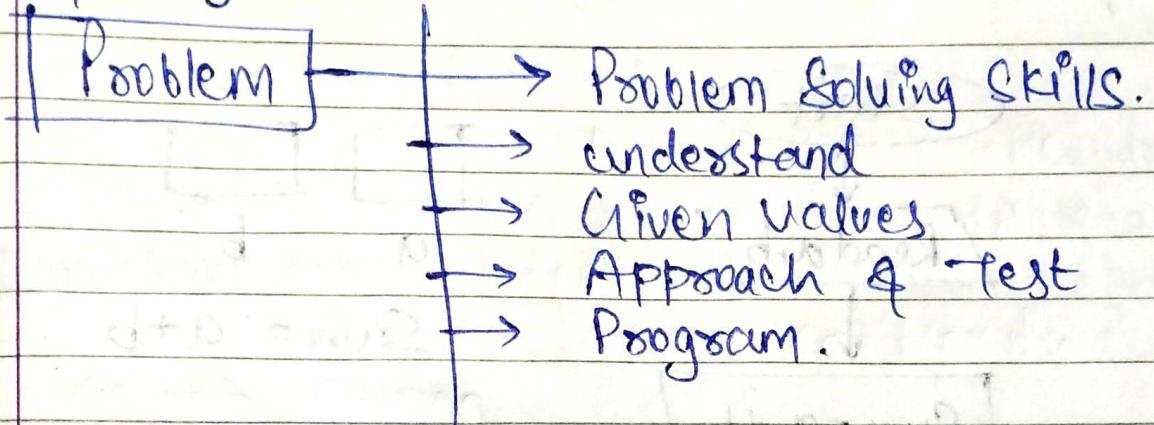
(\*) Quick sort

(\*) Leetcode - Powerset/Subset concept

(\*) Leetcode - phone pad combination

(\*) Rat in a Maze & Recursion

## Getting Started.



### ⑦ flow chart : { Diagram representation }

(1)

→ Terminator (Start, End).

(2)

→ Input / Output.

(3)

→ Processing, Calculation.

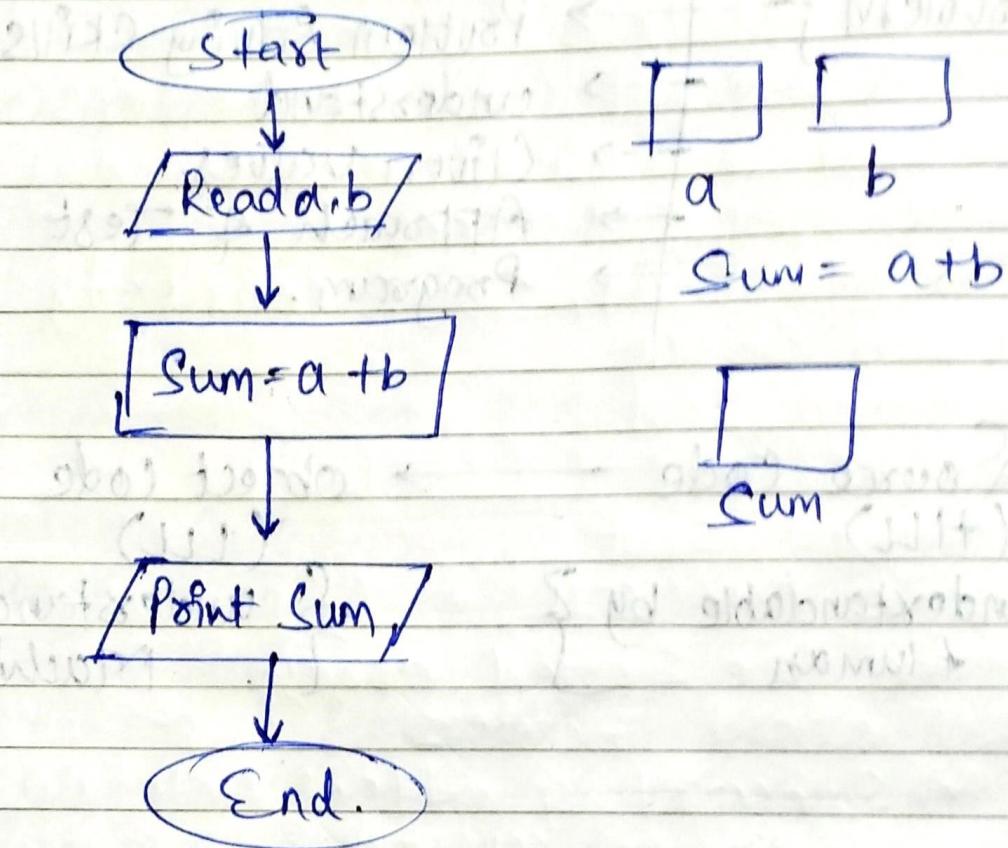
(4)

→ Condition, decision making.

(5)

→ Connector

Eg:- Sum of two Number



# Pseudo Code: General way of representing code & Algorithm.

Eg: Sum of two no.

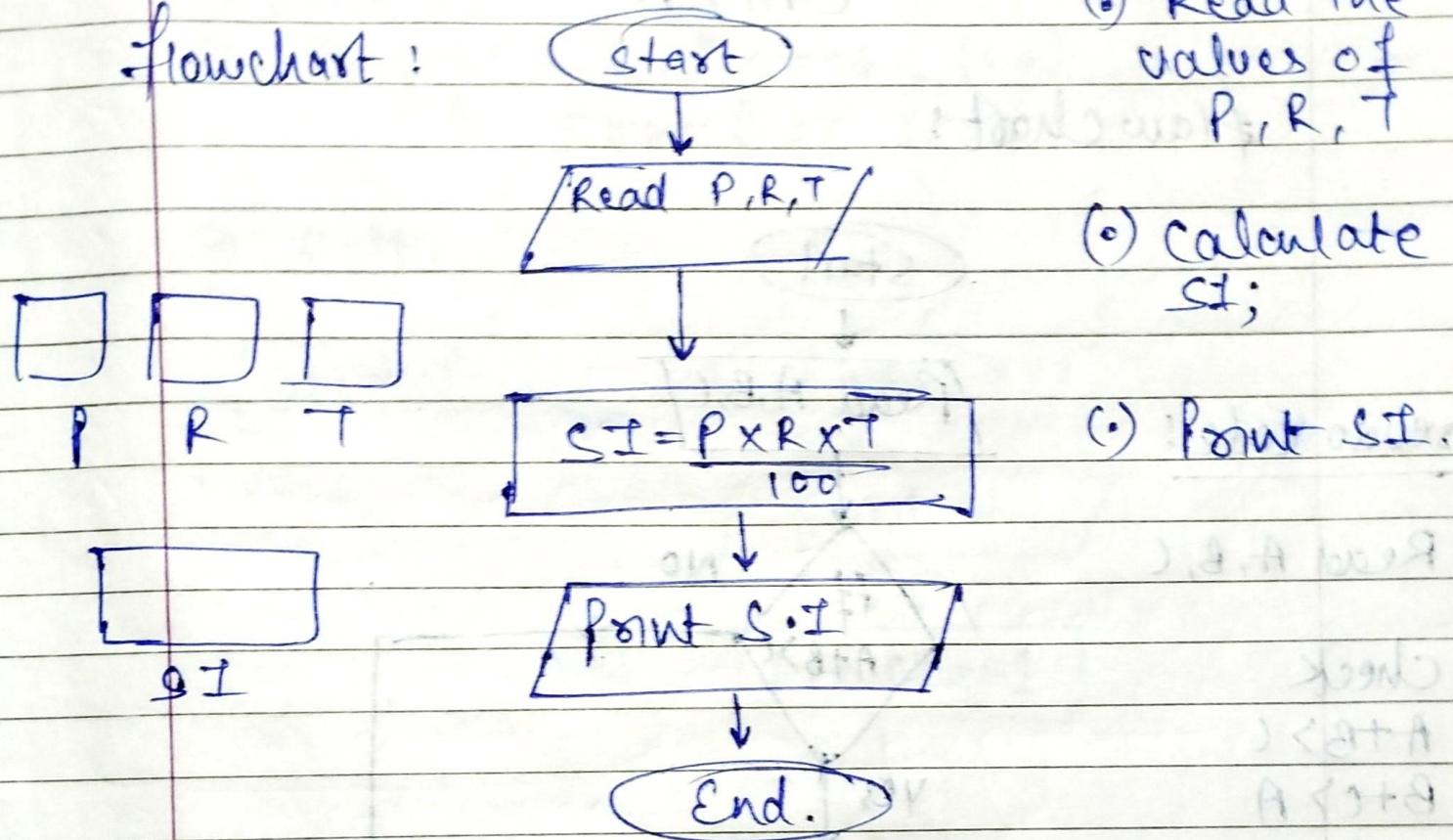
Pseudo code :

- (1) Read two no.(a,b)
- (2) Sum = a+b
- (3) Point Sum
- (4) Exit

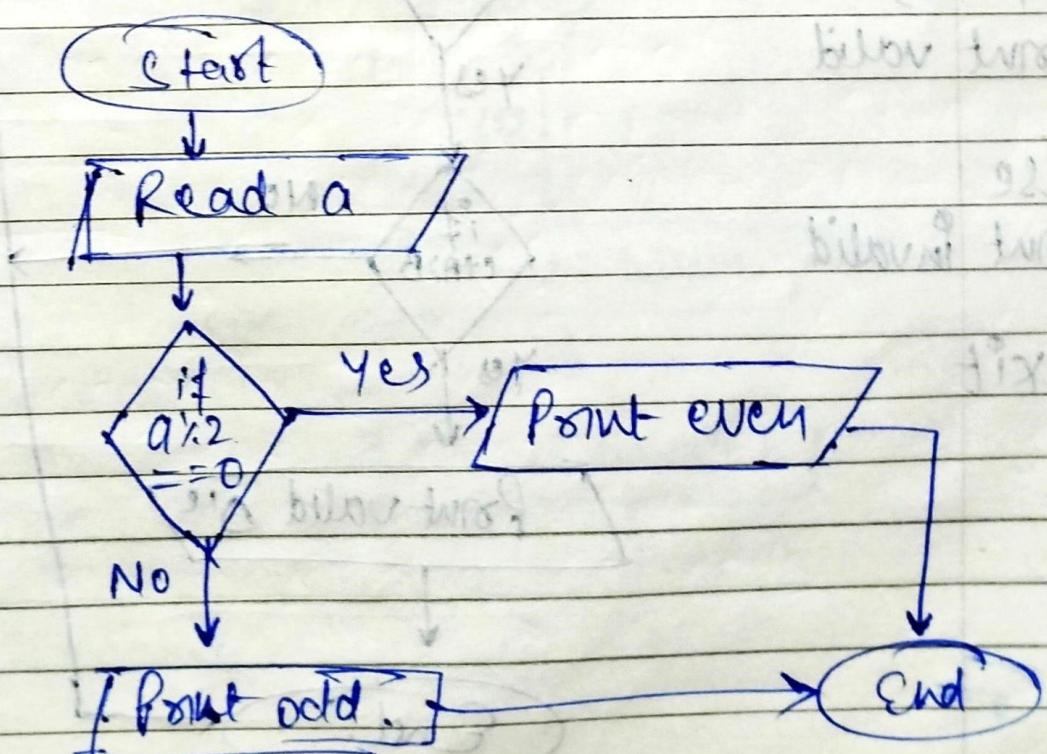
Eg:- Calculate Simple Interest

$$SI = \frac{P \times R \times T}{100}$$

flowchart :



Eg:- Read value & decide even or odd.



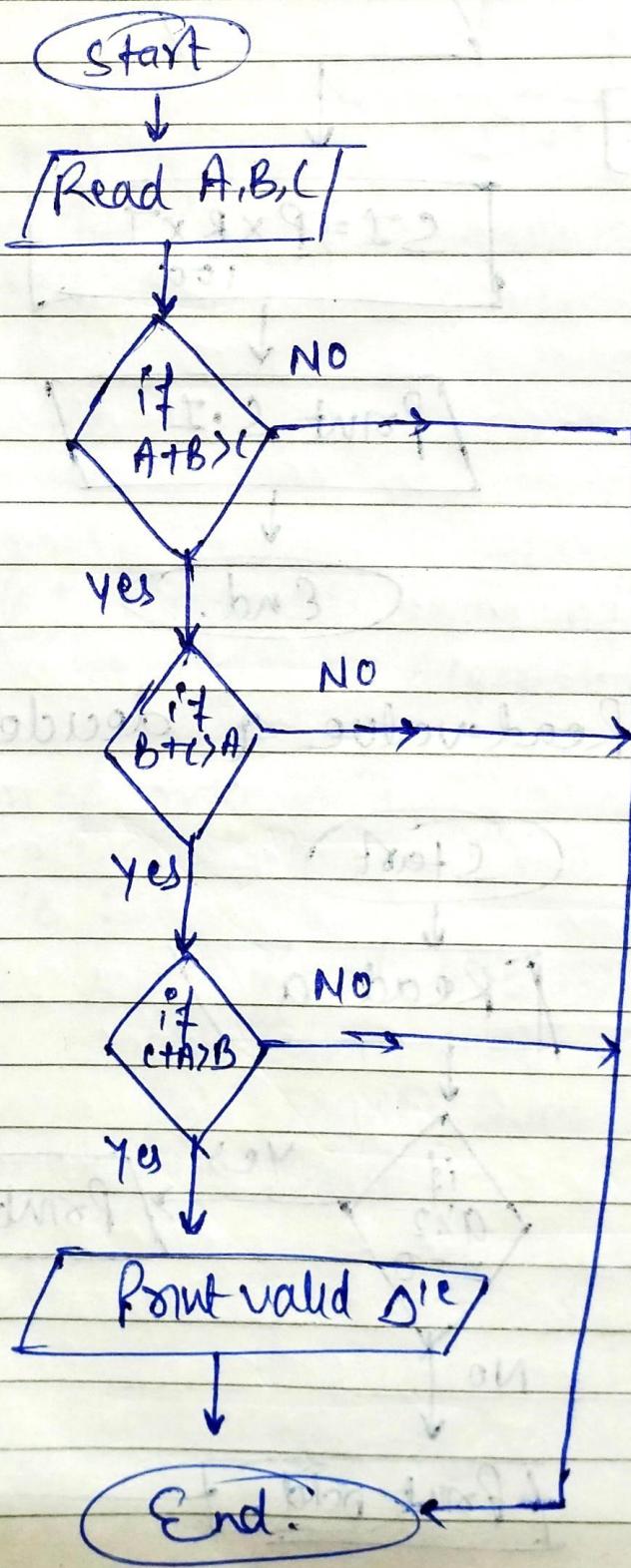
✓ Eg:- valid triangle

Condition :  $A+B > C$

$B+C > A$

$C+A > B$

flow chart:



Pseudo code:

\* Read A, B, C

\* Check

$A+B > C$

$B+C > A$

$C+A > B$

\* If yes

Print valid

\* else

Print invalid

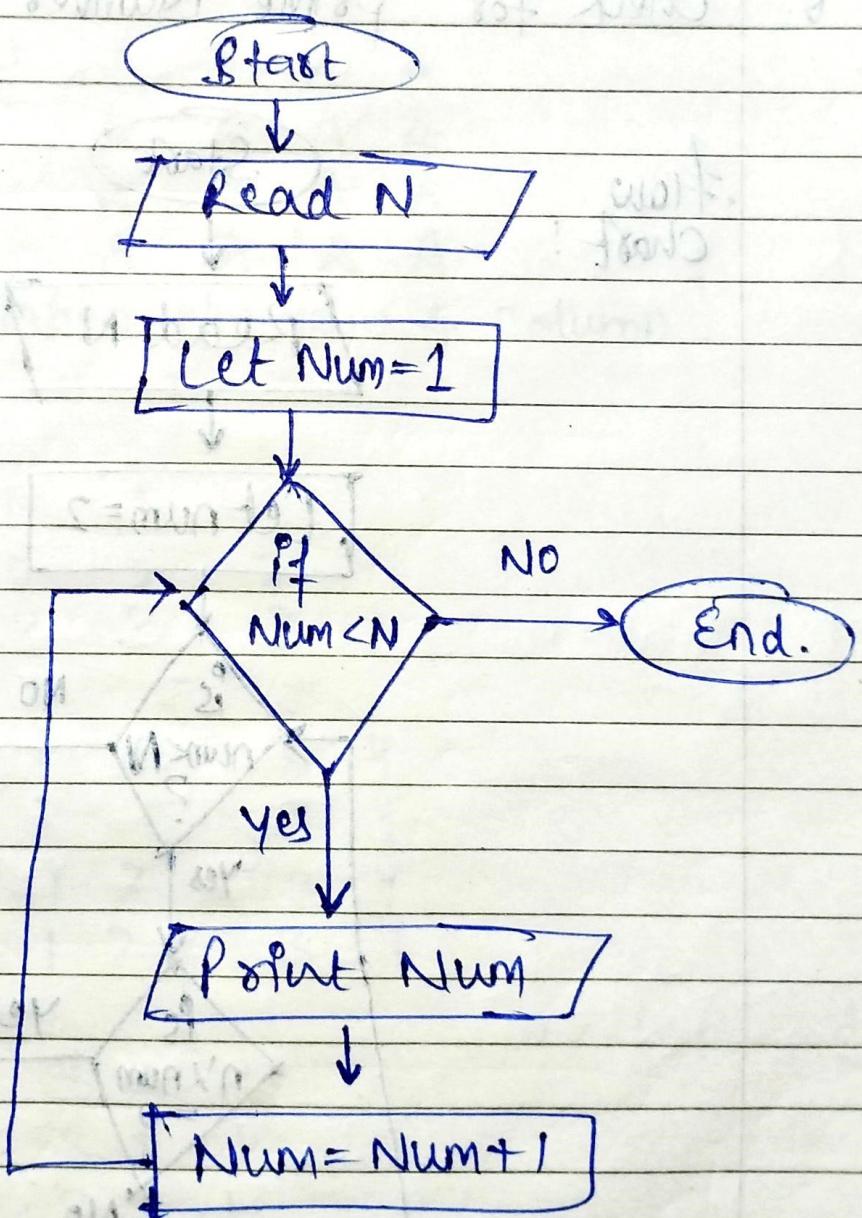
\* Exit

✓ Eg:- Take value from user in N, print upto N  
 (N ∈ Real Number)

Pseudo code :-

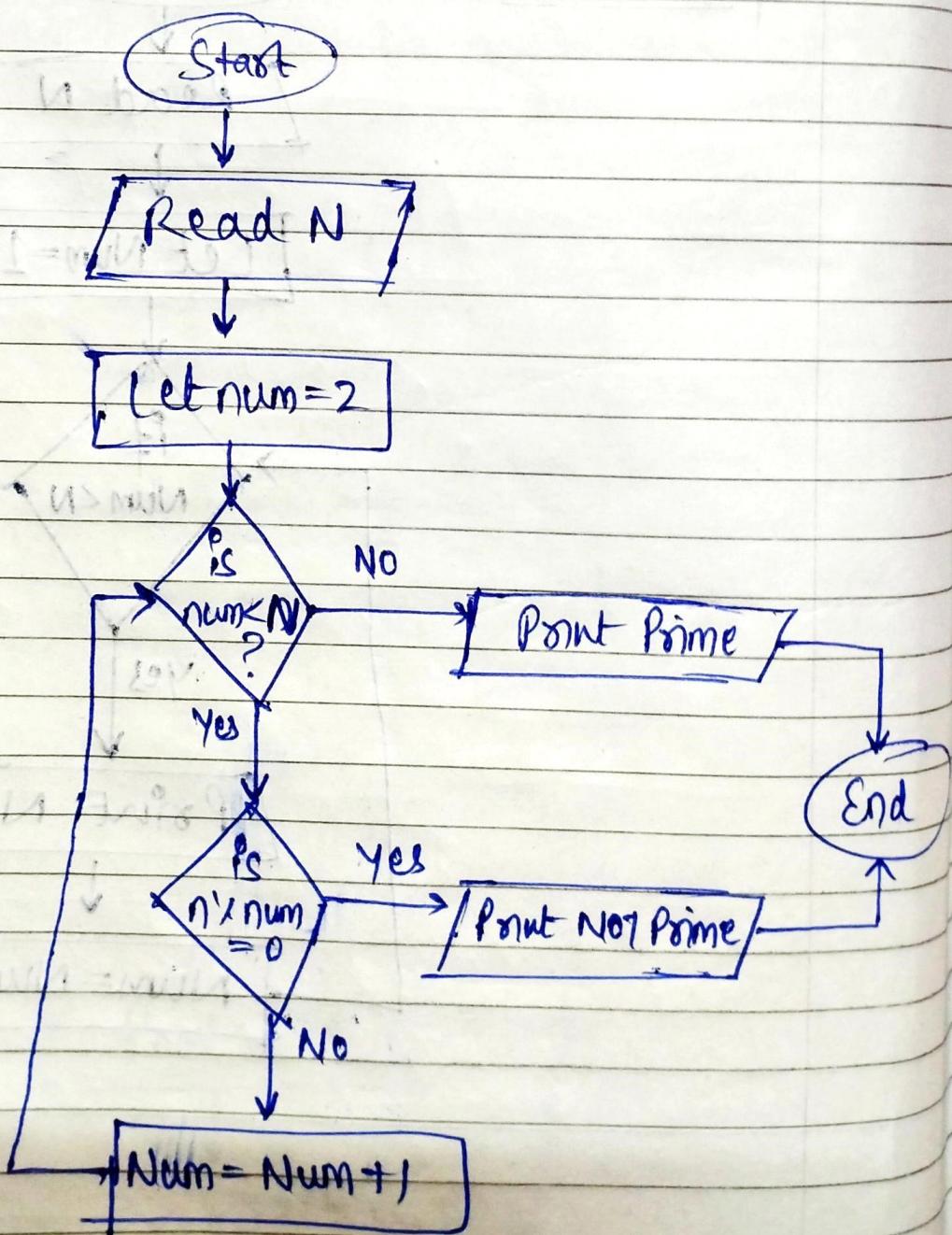
- (1) Read N
- (2) ~~for( i=0, i<=N ; i++ )~~  
     { print i ; }
- (3) Exit.

flowchart :



- ✓ Q. Read value from user in N.  
print only even Number upto N.
- ✓ Q. find sum from 1 to N (Inclusive)  
user will put value of N.
- ✓ Q. find  $n!$ , user will Read n.
- ✓ Q. check for prime Number , Read N.

flow chart :



- Q.  $a-z \rightarrow$  lowercase  
 $A-Z \rightarrow$  uppercase  
 $0-9 \rightarrow$  numeric
- E Read value from user and print accordingly.

- Q. Read value N.  
 Sum of all even no. upto N.

- Q. Read value of temp in f & convert to c.

Q. Print pattern:

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

Hint: find relation b/w row & column.

Q. Pattern :

1	1	1
2	2	2
3	3	3

Q. Pattern :

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

Q. Pattern :

3	2	1
3	2	1
3	2	1
3	2	1

Hint:  $(n-j+1)$

✓ Q. Pattern:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

✓ Q. Pattern:

1*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*

Upper triangle.

Hint:  $(\text{col} \leq \text{row})$

Q. Pattern:

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

✓ Q. Pattern:

1	2	3
4	5	6
7	8	9
11	12	13
15		

Q. Pattern:

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

Hint:  $i+j-1$

Q. Pattern: 1

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

Hint:  $(i-j+1)$ .

Q. Pattern: A A A

✓  
 B B B  
 C C C

Hint: Char ch = 'A' + row - 1

Q. Pattern: A B C

✓  
 A B C  
 A B C

Hint: char ch = 'A' + j - 1

Q. Pattern: A B C

✓  
 B C D  
 C D E

Hint:  $i+j-1=1$

$i+j-1 + 'A'-1 = 1 + 'A'-1$

$'A' + i+j-2 = 'A'$

char ch = 'A' + i + j - 2

Q. Pattern:



A			
B	C		
C	D	E	
D	E	F	G
E			
F			
G			

Hint:  $\text{char ch} = \text{'A'} + i + j - 2$ .

Q. Pattern:



D			
C	D		
B	C	D	
A	B	C	D

Hint: for first char.

$\text{char ch} = \text{'A'} + n - i$  /  $\text{ch} = \text{'D'} - i$   
then increment.

Q. Pattern:

—	—	—	—	*
—	—	—	*	*
—	—	*	*	*
—	*	*	*	*
*	*	*	*	*

↙ while for iteration

Hint: 3 while loop.

(1) 1<sup>st</sup> while → space iteration

(2) 2<sup>nd</sup> while → star space

(3) 3<sup>rd</sup> while → star.

Space =  $n - i$

Q. \* \* \* \* Hint:  $n-i+1$

\* \* \*

\* \*

\*

Q. Pattern: \* \* \* \* Hint:

— \* \* \* Stars =  $n-i+1$

— - \* \* Space =  $i-1$

— — — \*

Q. Pattern :-

Space =  $n-i$

Start =  $i-1$

while ( $row < n$ )

```
{     int space = n - row;
      while (space) {
        cout << " ";
        space--;
      }
    }
```

```
cout << endl;
row = row + 1;
}
```

~~int col = 1;~~

```
while (col <= row)
  { cout << col;
    col++;
  }
```

int start = i - 1;

```
3rd → while (start)
      { cout << start;
        start--;
      }
```

## Q. Pattern:

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

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

→ 8010

→ 9009

(A → 8010) gives

2008 → (A → 9009) for

B (9009) gives

C → 8010

for

1000 → 8010

21 pages = 2008

## (+) Bitwise operator :-

AND      &  
 OR        |  
 NOT      ~  
 XOR      ^

### (i) AND(&) Bitwise : (Both showed as 1)

Input		Output
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

Eg:-  $5 + 7$  $\downarrow$  $\downarrow$ 

101

111

$$\begin{array}{r} 101 \text{ (5)} \\ + 111 \text{ (7)} \\ \hline 101 \end{array}$$

 $\rightarrow 5$ 

### (ii) OR(|) Bitwise : (Anyone showed as 1)

Input		Output
x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

Eg:-  $a=2 \quad 010$  $b=4 \quad 100$  $\underline{110 \rightarrow 6}$ 

### (iii) Not(~) Bitwise : (Invert Bit)

x	z
0	1
1	0

Eg:-  $a=2 \quad (10)$ 

$\therefore$  int a  $\rightarrow$  4 byte =  $4 \times 8 \text{ bit}$   
 $= 32 \text{ bit}$

$a = 000\dots 10$   
 $\sim a = 111\dots 01 \Rightarrow \sim a = -3$

① XOR (^) Bitwise: Give true only at one 1.

Input		Output
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

$$\text{eg!:- } a=2 = 10 \\ b=4 = 100$$

$$a \wedge b = 10 \\ 100 \\ \hline 110 \rightarrow 6.$$

$$\text{eg!:- } a=5 = 101 \\ b=7 = 111 \\ \hline 010 \rightarrow 2.$$

② Left Shift:

Eg!:-  $5 \ll 1$  (Shift 5 by 1 on left)

000...101

000...1010

↑ shift by 1 bit

Eg!:-  $3 \ll 2$

000...011

000...01100

↑ shift by 2 bit

Eg!:-  $n=n \ll 1;$

③ Right Shift:

Eg!:-  $P=P \gg 1;$

Eg!:-  $15 \gg 1$  (Shift 15 by 1 on right)

0000...1111

0000...0111

↑ shift by 1

+ve Padding = 0

-ve Padding = depend on compiler.

15

Page No.

Date

Note : Padding (adding of a number) of a number in +ve number is always start with 0.

In negative no., Padding depend on compiler.

#

## Abonacci series :-

0, 1, 1, 2, 3, 5, 8, 13, 21, 24 ...

Sum = n ) ②  
for( int i = 0 ; i < n ; i++ )  
{  
    int sum = a + b;  
    cout << sum ;  
    a = b;  
    b = sum;

b = 1  
a = 0

## ⊕ Decimal & Binary conversion :-

Eg:-  $(15)_{10} \rightarrow (?)_2$

8 4 2 1  
1 1 1 1 = 15.

0 0 1 0 . 1 0 0 1 0 0

2	15	1	1
2	7	1	
2	3	1	
	1		

$(1111)_2 \rightarrow \{ \text{Approach 1} \}$

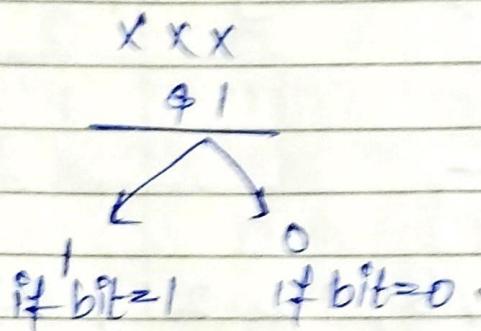
for Decimal to Binary divide by 2

for Binary to decimal use 8 4 2 1 or etc.

(\*) How to find the binary code for any decimal  
 { Approach 2 }

Eg! -  $n = 5 \rightarrow \text{XXX}$

whilst  $n! = 0$   
 {



bit =  $n \& 1$ ;  
 $n >> 1$ ;  
 }

Eg! - 1, 2, 3  
 answer = 123;

$$\begin{aligned} \text{ans} &= (\text{digit} * \text{pow}(10, i)) + \text{ans} \\ \boxed{\text{ans} &= (\text{digit} * 10^i) + \text{ans}} \\ &\downarrow \\ \text{Reverse. (Eg: 321)} \end{aligned}$$

logic?  $\rightarrow$  int ans = 0;

$$\begin{aligned} \text{ans} &= (\text{digit} * 10^0) + \text{ans}; \\ \text{ans} &= 1 * 10^0 + 0 \\ &= 1 + 0 = \underline{1}. \end{aligned}$$

$$\begin{aligned} \text{ans} &= (\text{digit} * 10^1) + \text{ans} \\ &= 2 * 10^1 + 1 \\ &= 20 + 1 = \underline{21} \\ &= \underline{21} \end{aligned}$$

$$\begin{aligned} \text{ans} &= \text{digit} * 10^2 + \text{ans} \\ &= 3 * 10^2 + 21 \\ &= 300 + 21 \\ &= \underline{321}. \end{aligned}$$

ans =  $(\text{ans} * 10) + \text{digit}$   
 } straight (Eg: 123)

Concept to convert ~~number~~ digit into number;

# WAP for Conversion of Decimal to Binary.

(1/2)

Page No.

Date

17

#include <iostream>

#include <math.h>

using namespace std;

int main()

cout << "Enter the no.";

{ float

int

cin >> n;

ans = 0; i = 0; }

while (n != 0) {

int bit = n % 2;

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

n = n / 2;

i++;

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

return 0;

}

Eg:-

423

and if we want 3 from 423

let n = 423

digit = n % 10;

3  
digit

digit\_start = n / 10;

42  
digit\_start

Concept to take the digit out of Number

- ~~H/W~~
- Q. Convert negative decimal to binary code  
eg :-  $n = -10$

~~H/W~~

# WAP to convert Binary code to decimal. (\*2)

```
#include <iostream>
#include <math.h>
using namespace std;
```

```
int main()
```

```
{ int i{0};  
float ans{0};
```

```
cout << "Enter the digit";
```

```
cin >> n;
```

```
while (n != 0)
```

```
{
```

```
int digit = n % 10;
```

```
if (digit == 1) {
```

```
ans = ans + pow(2, i);
```

```
}
```

```
n = n / 10;
```

```
i++;
```

```
}
```

```
cout << "Answer is: " << ans;
```

```
}
```

#

Switch :int | char onlySwitch ( expression ) {break ;↳ To Stop .Case 1 :Statement 1 ;continue ;break ;↳ To skip .Case 2 :Statement 2 ;break ; or continue ; \*default :Statement xyz ;X ↳ we can't

}

use continue in switch

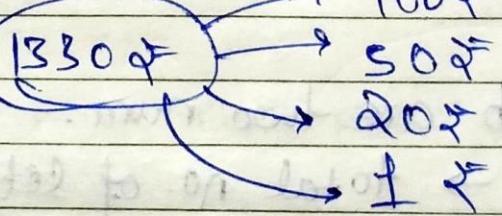
Note : `exit()` :- It is a exit function, declared under `<stdlib.h>`, terminate a C++ program.

Q. WAP for calculator using switch.

Ask user for variable and operator.

Q.

Total amount = 1330 ₹



Amount of notes required to make the total amount ;

`return total amt ;`

(\*) Note:

Even numbers | odd numbers

$$(\text{LSD}) == 0 \quad | \quad (\text{LSD}) == 1$$

Condition: if (num & 1) { condition check  
 {  
 return 0; // odd.  
 } }

return 1; // Even

Q. WAP for  ${}^nC_r = \frac{n!}{r!(n-r)!}$

Hint: Make factorial fun.  
 Then make fun<sup>n</sup> for  ${}^nC_r$ .

$$\text{Q. } A \cdot p = (3 \times n + 7)$$

$\therefore$  If  $p = n$ ;  $A \cdot p = n^{\text{th}}$  term.

Q. a & b are two num.

$O/p \rightarrow$  total no. of set bits in a & b.

$$\text{eg:- } a = 2 \quad 10$$

$$b = 3 \quad 11$$

$$O/p = 3$$

Note:  $\text{size\_t}$  ;

↳ unsigned

int

Page No.

Date

21

based on our system.

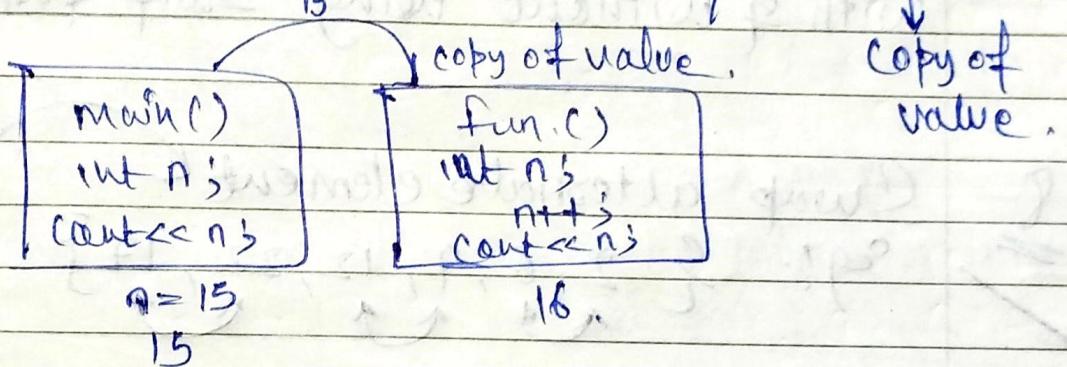
Q. fibonacci series function.

i/p  $\rightarrow n$  o/p  $\rightarrow$   $n^{\text{th}}$  fibonacci no/series.

Eg:- 1, 1, 2, 3, 5, 8, 13, 21, ...

(\*) Note: function call stack;  
→ understand this!

(\*) Pass by value;  $\rightarrow$  (theory !) / (call by value)



(\*) Note: int  $\rightarrow$  range  $(-\varphi^{31}, \varphi^{31}-1)$

$$\text{INT\_MIN} = -\varphi^{31}$$

$$\text{INT\_MAX} = \varphi^{31}-1$$

Q. Calculate the max and min value in an array.

Take elements and size of array from user.

use INT\_MIN and INT\_MAX inbuilt function.

- Take the size & element from the user.  
 Take the key( element needed to found)  
 Perform ~~ba~~ linear search  
 Return true if present , else return false;

- WAP to Reverse an array. → use third variable  
 Take values from the user.  
 With & without using swap function .

Swap alternate element

e.g:- { 2, 6, 9, 12, 22, 17 } → Hint  
{ 6, 2, 12, 9, 17, 22 }

if ( $i + 1 < \text{size}$ )  
Swap(arr[i], arr[i+1])

Hint: Swap ( arr [start] , arr [end] )  
{ }

$$\begin{aligned} \text{Start} &= \text{Start} + 2; \\ \text{End} &= \text{End} + 2; \end{aligned}$$

{ }

int ans = 3;

ans = ans ^ arr[i];

find unique element in an array. → Hint: use XOR (^)  $\text{ans} = \text{ans} ^ \text{arr}[i]$

find duplicate element in an array.

→ Hint: XOR (^) with whole array.

Eg:- 1, 2, 3, ..., N, ..., (N+1), X.

1, 2, 3, ..., X, ..., (N+1) → duplicate

Hint: Two pointer approach.

## Q. Array intersection.

Eg:-  $\text{arr1}[] = \{1, 2, 3, 4\}$

$\text{arr2}[] = \{2, 4, 6, 8\}$

$\text{Op} = \{2, 4\}$

\* Q Pair sum. of array. Hint:-

\* Q Triplet sum of array. → use 3 for loop.  
 $\text{for}(i=0; i<\text{size}; i++)$   
 $\text{for}(j=i+1; j<\text{size}; j++)$

\* Q Sort 0's & 1's. in array. →  $\text{for}(k=j+1; k<\text{size}; k++)$

## B. Leetcode : 1207 (unique number of occurrence).

### unique element

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

$\text{ans} = \text{ans} \wedge \text{arr}[i];$

}

return ans;

}

[Leetcode]

### duplicate element

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

$\text{ans} = \text{ans} \wedge \text{arr}[i];$

}

$\text{for}(int i=1; i<\text{size}; i++)$

{

$\text{ans} = \text{ans} \wedge i;$

}

return ans;

}

Q

Leetcode : 442 ( find all duplicate in array ).

## Sol<sup>n</sup> (\*) Array Intersection .

Approach: Two Pointers

$$\sum_{j=0}^5 j \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 2 & 2 & 2 & 3 & 4 \\ \hline \end{array} = 900$$

$$\rightarrow \boxed{2 \mid 2 \cdot 3 \mid 3} = 0.882$$

$\rightarrow \text{arr1}[i] < \text{arr2}[j]$

$\rightarrow \text{arr1[i]} == \text{arr2[j]}$   
push in vector  
 $i++, j++;$

$\rightarrow arr[i] > arr[j]$   
 $]++;$

Code: int i{3}, j{3};

vector<int>ans;

while ( i < n && j < m ) // In Range of  
size

```

if (arr1[i] == arr2[j])
    ans.push_back (arr1[i]);
    i++;
    j++;
}
else if (arr1[i] < arr2[j])
    i++;
}
else
    j++;
}
return ans;
}

```



PairSum :-

```

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] == s)
        {
            vector<int> temp;
            temp.push_back (min (arr[i], arr[j]));
            temp.push_back (max (arr[i], arr[j]));
        }
    }
}

```

↓  
Inbuilt to find max & min.

ans.push\_back(temp);

} sort is built-in used to sort an array.

} sort(ans.begin(), ans.end());  
return ans;

}

\* Sort 0's & 1's in array.

int left = 0, right = n - 1;

while (left < right) {

    while (arr[left] == 0 && left < right) {

        left++;

}

    while (arr[right] == 1 && left < right) {

        right--;

} if (left < right)

        swap(arr[left], arr[right]);

        left++;

        right--;

}

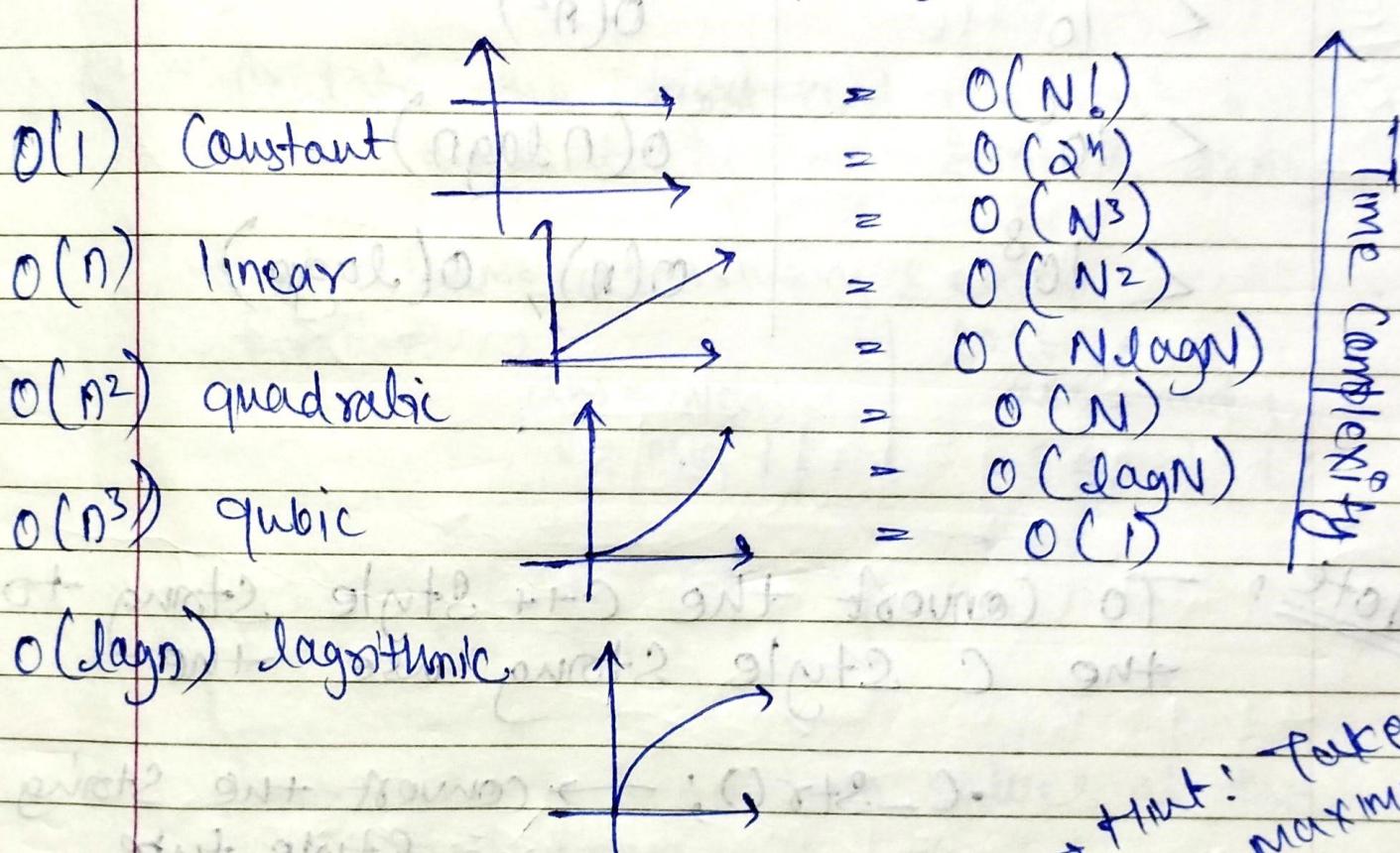
(i) Time Complexity : It is the amount of time taken by an algorithm to run.  
 → as a function of length of the input

Big 'O' notation → Θ → Omega

↓  
Upper bound.

↓  
for avg-case complexity.

↓  
lower bound.



Eg:-  $f(n) = 2n^2 + 3n = O(n^2)$ .

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

$$f(n) = n^3 / 300 = O(n^3)$$

$$f(n) = 1000 = O(1)$$