~~Code~~

Codehelp.in

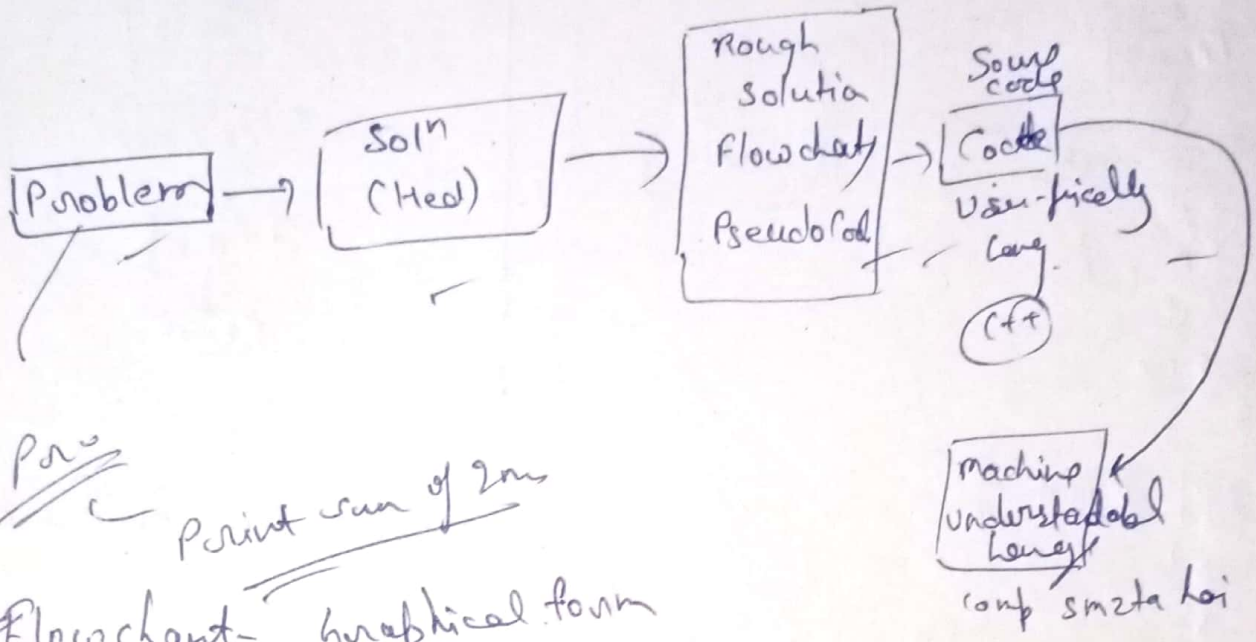
D.S.A → by Love Babbar

Class Schedule ↓

- Friday → 9-1pm
- Saturday → 12-2pm
- Sunday → 12-2pm

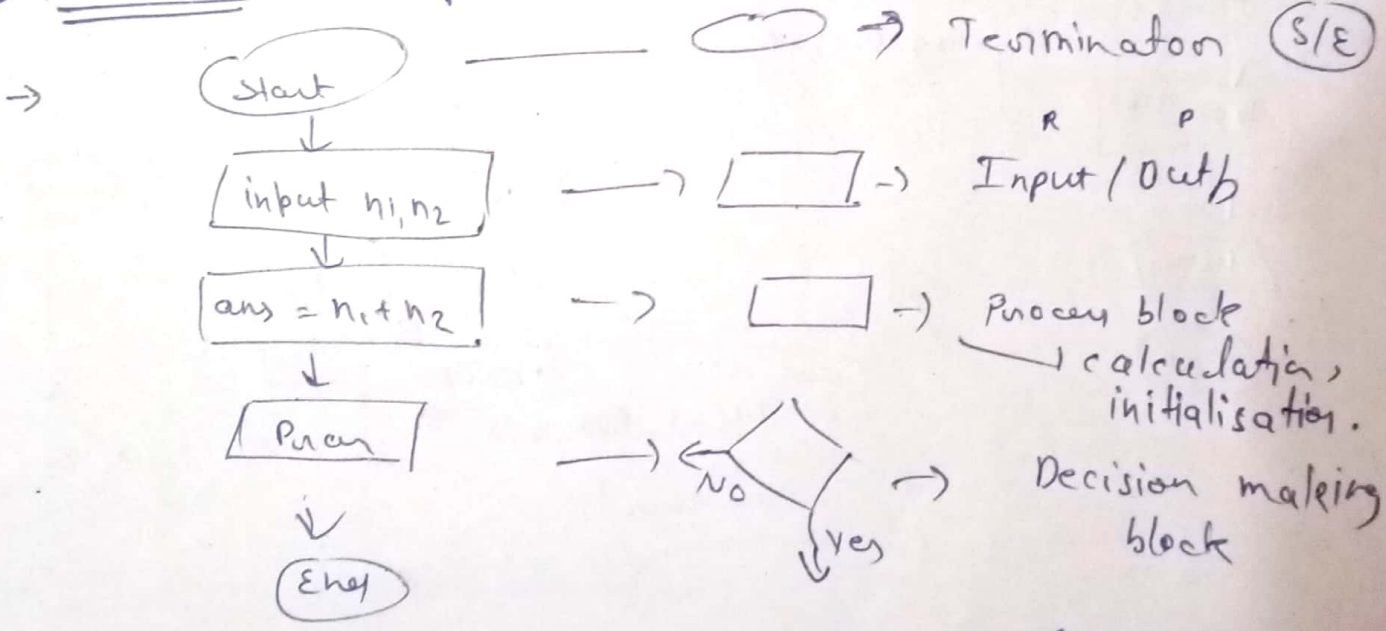
-) understand the prob
→ i/p values
→ Approach

2/Feb/23 Lect-1 (Fundamental of Lang)

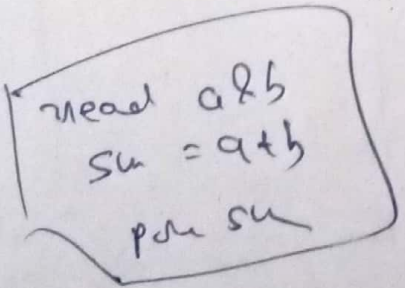


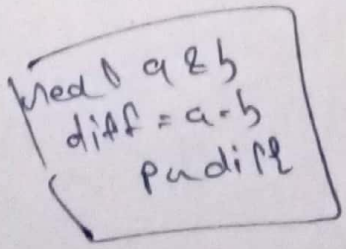
Pro: Print sum of 2 no.

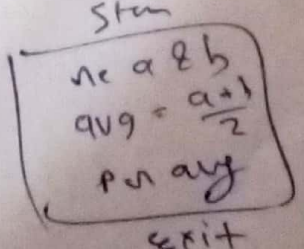
Flowchart- Graphical form



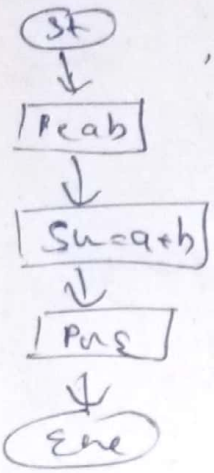
Pseudo Code: Generic way to represent your code in text form

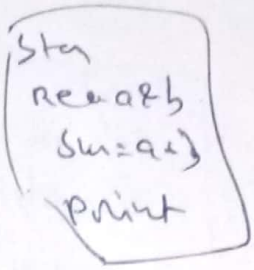




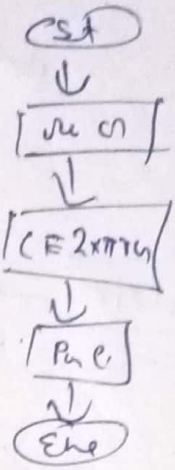


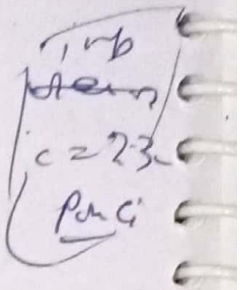
1) Add 2 no by taking input



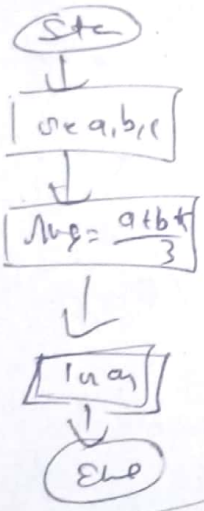


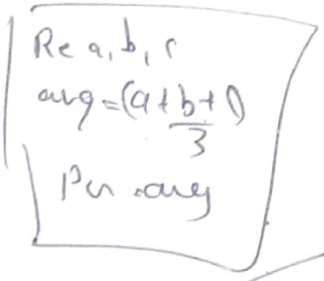
=> Find Circle of circle



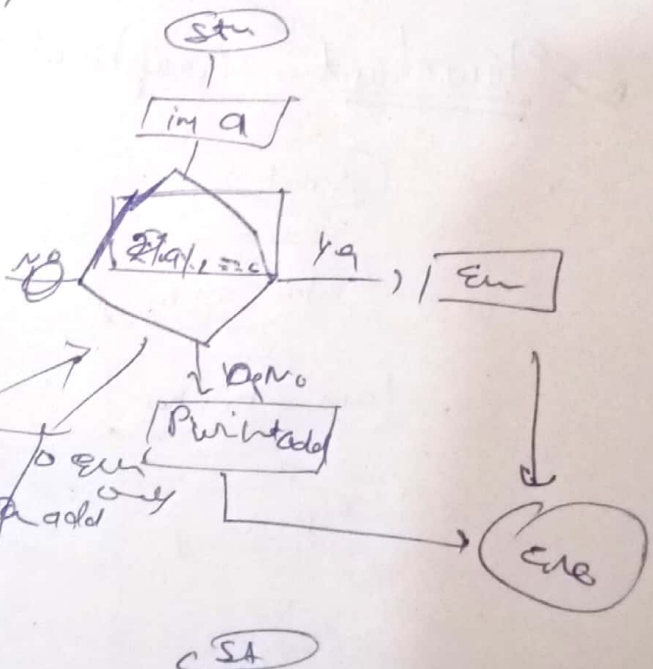


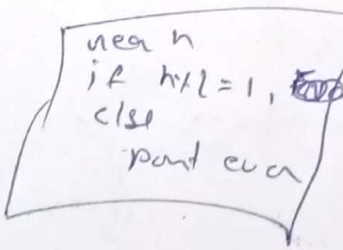
=> Ave of 3 no

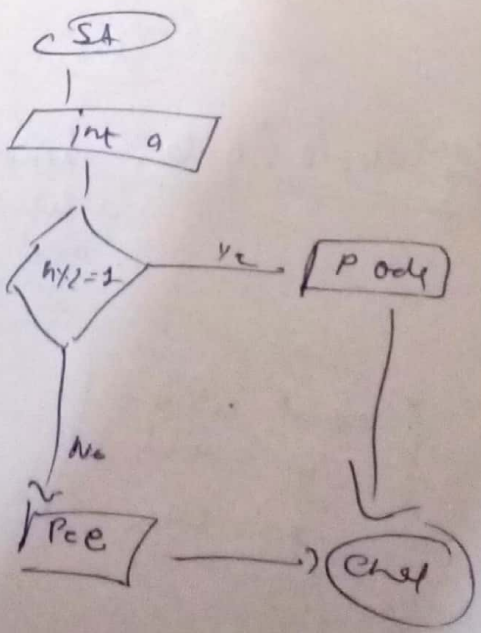




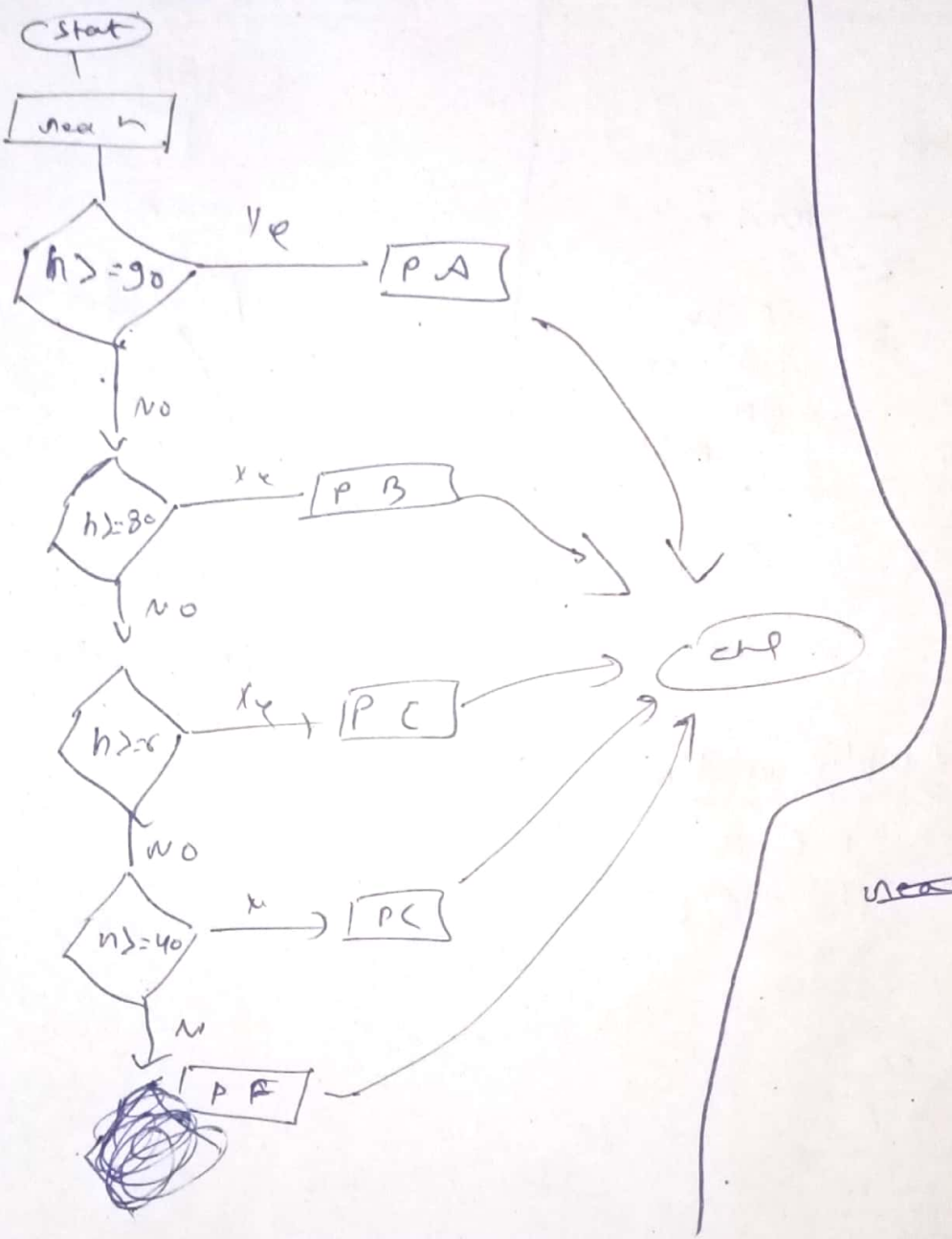
=> check no. even or odd







=> Students & Grade flowchart

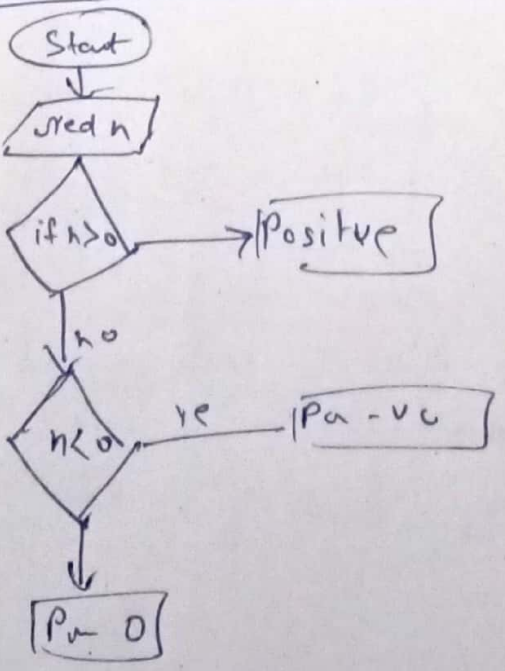


>=90 -> A
>=80 -> B
>=60 -> C
>=40 - D
<40 - E

```
read n
if n>=90
    P. A
elseif n>=80
    P. B
elseif n>=60
    PC
elif n>=40
    PD
else
    PE
```

~~read a, b, c~~

=> check no. is +ve, -ve or 0



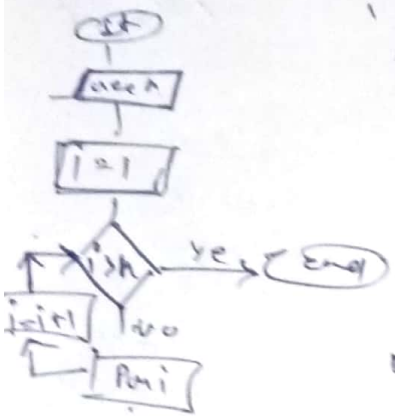

```
read n
if n>0
    Pr +
elseif n<0
    P. -
else
    P 0
```

⇒ Print 1 to N

n=5

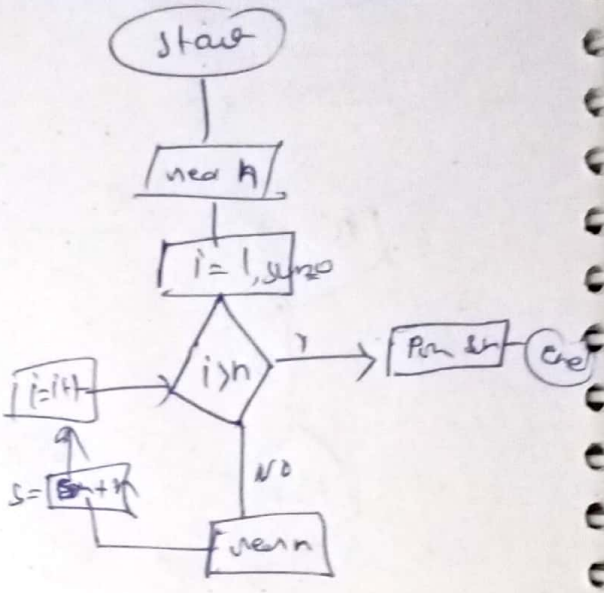
1, 2, 3, 4, 5

Flow



1 → read n
2 → i=1
3 → if i>n
exit
4 → else
print i
i=i+1
go to step ②

⇒ Add n numbers from user



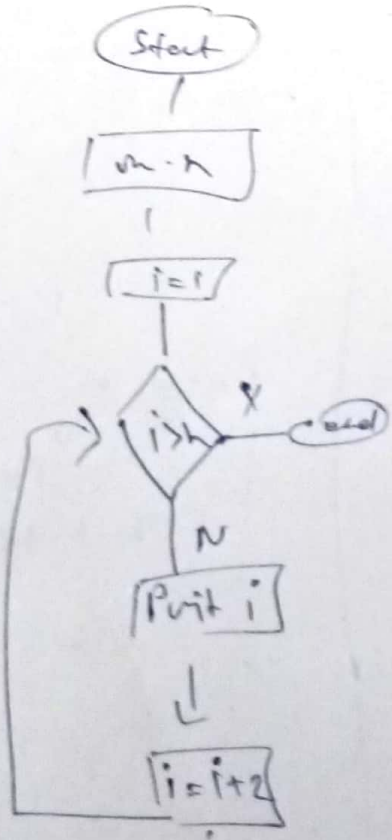
1 → read n
2 → i=1
3 → sum=0
4 → if i>n
print sum
end
else
read n
sum=sum+n
i=i+1
goto step ④

⇒ Print 1 to N only odd no.

n=6

1, 2, 3, 4, 5, 6

odd

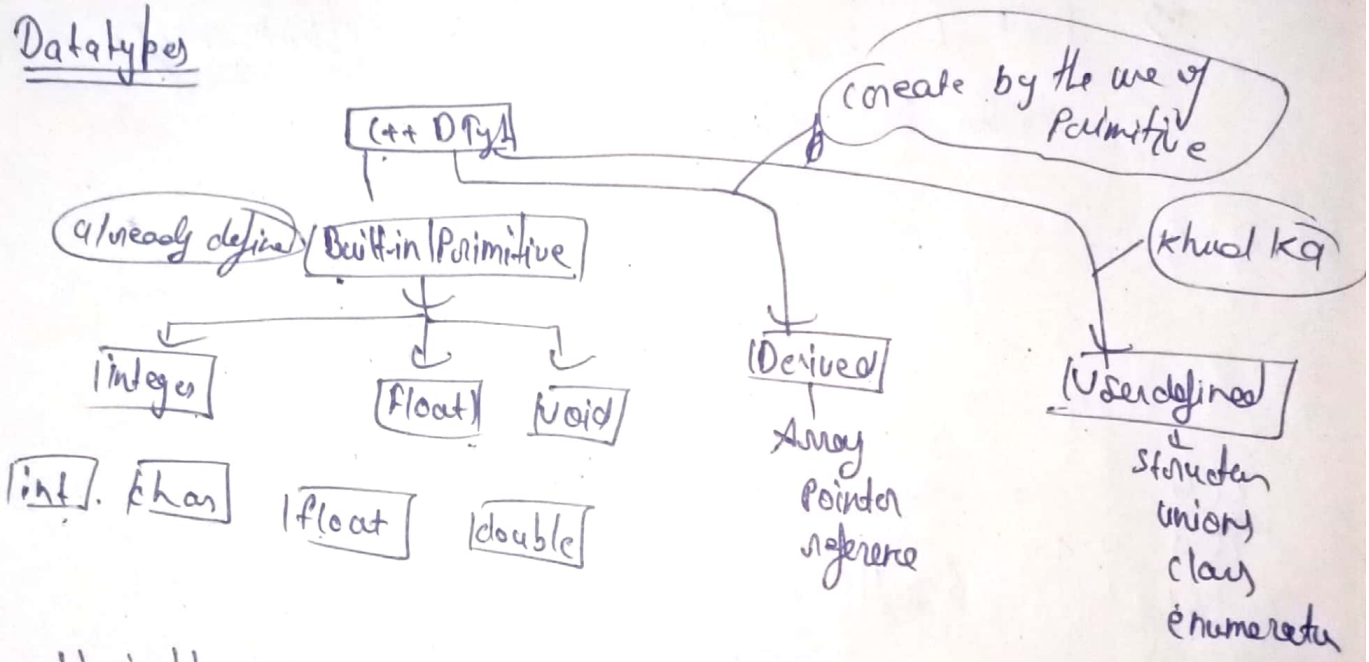


E-2 Create your First Program

```
#include<iostream>
using namespace std;
int main()
{

}
```

cout << <<endl;
cin >>

Datatypes

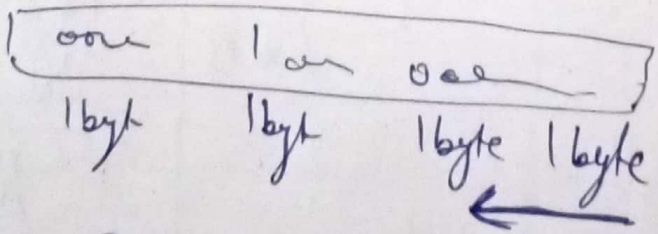


Variable

a=1, b= 10, etc

init

→ int → 4byte 1byte = 8bits
└ 32 bits



Total no of output = 2^{32}

0 -

$2^{31} - 1$

char → 1 byte → 8 bit

char —

1byte

'a'

8bits

2^8 → 128
ya 0-255

2^8 =

Boolean → T — 1
F — 0

1 space

{Smallest addressable memory 1 byte}

→ 1 byte minimum required

bool flag = 1;

float

1.2 float f = 1.2

double f = 1.200;

{Short → 2 byte → 16 bit}

$2^{16}-1$

max $2^{32}-1$

2 | 6
2 | 3 0
1 1

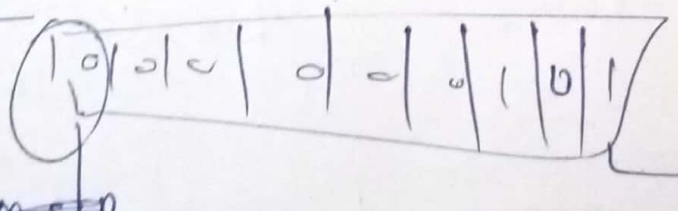
6 = 110

Stored data

5 → 101

int a = 5

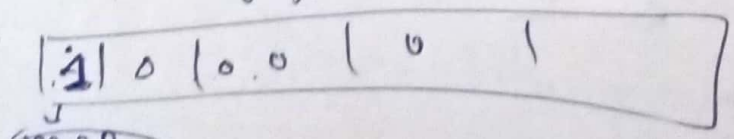
(+) 0 होता है



MSB — Most significant Bit

LSB

(−) 1 होता है



MSB

2's comp
↓
1's complement + 1
↓
1st flip

How to stored −1 number

7 = 111

7 → 00000111

1's → 11111000

+1

2's 11111001

1+1 = 10 in Binary

2|5 1
2|2 1
|1 0

111
0000111
+1
0001000

int a = -5

(i) ignore - sign

(ii) Binary equivalent

000 ____ 00101

(iii) 2's compl

1's → 1111 ---- 11010

1

1111 — 11011 → -5

Read 1's

0000 --- 00100

+1

--- 00101

2 by → 16 bit

T = 2^{16}

US 0 - $2^{16}-1$ S: $-2^{15} \to 2^{15}-1$

6 by = 48

T.C = 2^{48}

U: $0 \to 2^{48}-1$

S $0 \to -2^{47} \to 2^{47}-1$

signed → both +, - store, 0

unsigned → only +ve, 0

n bits

* sign → $(-2^{n-1} \to 2^{n-1}-1)$

* unsign → $0 - 2^n - 1$

int → 4 byte → 32 bits

← 32 →

Total comb → 2^{32}

unsing

↓

$0 \to 2^{32}-1$

signed

2^{31} -ve | 2^{31} +ve

$-2^{31} - 2^{31}-1$

$\frac{2^{32}}{2} = \frac{2^{32}}{2^1} = 2^{32-1} = 2^{31}$

$-2^{31} \to 0 \to 2^{31}-1$

Scanned with CamScanner

Operators

- Arithmetic — H/w Precedence Table
- Relational
- Assignment
- Logical → && → multiple
  || → only one
  ! not
- Bitwise

1.3 Patterns I (HD)

{ Code :-

nested loops

2 loops — outer → rows; inner → col

multiple loops

① 

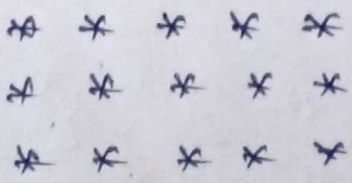
```
for (int row=0; row<3; row=row+1)
{
    for (int col=0; col<5; col=col+1)
    {
        cout << "* ";
    }
    cout << endl;
}
```

① row - observation
total row = 3

② col - Obser

col 0 - 3*
col 1 - 3*
col 2 - 3*
col 3 - 3*
col 4 - 3*

⇓ dry run



② Solid Square Pattern

```
* * * *
* * * *
* * * *
* * * *
```

```
int n;
cin >> n;
for (int row=0; row<5; row++)
{
  for (int col=0; col<5; col++)
  {
    cout << "* ";
  }
  cout << endl;
}
```

③ Hollow rectangle

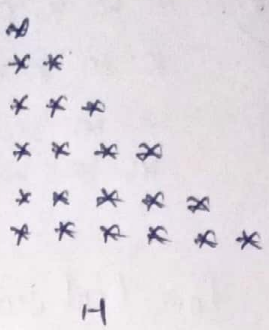
```
for (int row=0; row<3; row++)
{
  for (int col=0; col<5; col++)
  {
```

row 0 * * * * *
row 1 * *
row 2 * * * * *

~~for (int row=0; row<3; row++)~~
~~{~~
~~if (row==0 || row==2)~~
~~{~~
~~for (int col=0; col<5; col++)~~
~~{~~
~~cout << "* ";~~
~~}~~
~~}~~
~~else~~
~~{~~
~~cout << "* ";~~
~~for (int i=0; i<3; i++)~~

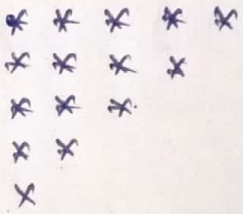
④ Half Pyramid →

```
for (int row=0; row<5; row++)
{
    for (int col=0; col<row+1); col+1
    {
        cout << "*";
    }
    cout << endl;
}
```



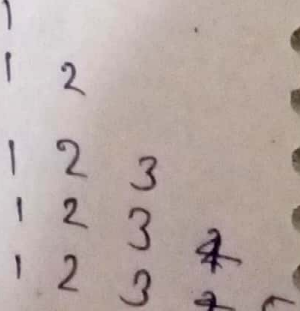
⑤ Inverted Half Pyramid →

```
int n;
cin >> n;
for (int r=0; r<5; r++)
{
    for (col=0; col<n-row; col++)
    {
        cout << "*";
    }
    cout << endl;
}
```



⑥ Numeric Half Pyramid →

```
int n;
cin >> n;
for (int r=0; r<n; row++)
{
    for (int col=0; col<row+1; col++)
    {
        cout << col+1;
    }
    cout << endl;
}
```



⑦ Inverted Half Numeric Pyramid.

```
for (int i=0; i<n; i++)
{
  for( int col=0 ; col < n-i ; col++)
      {
      { cout << col+1;
      }
    cout<<endl;
}
```

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

⑧ Full pyramid

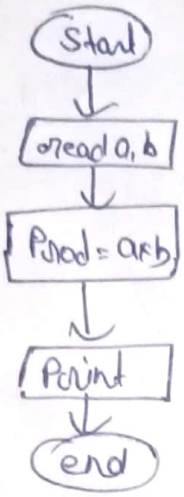
```
    *
   * *
  * * *
 * * * *
```

⇒ Namespace → used to avoid collision. *

⇒ Ternary operator.

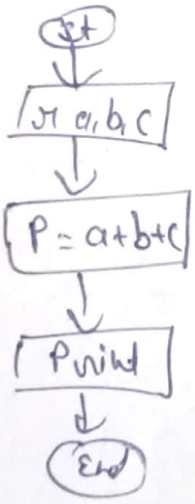
Assignment

① Multiply ~~by~~ 2 numbers.



read a,b
Product = a*b
Print product

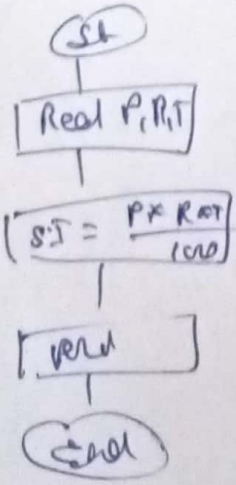
② Perimeter of △



read a,b,c
P = a+b+c
Print

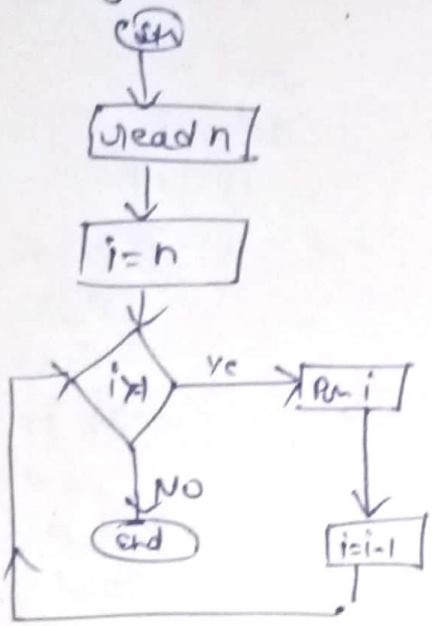
③ Find Simple Interest

$$S.I = \frac{P*R*T}{100}$$



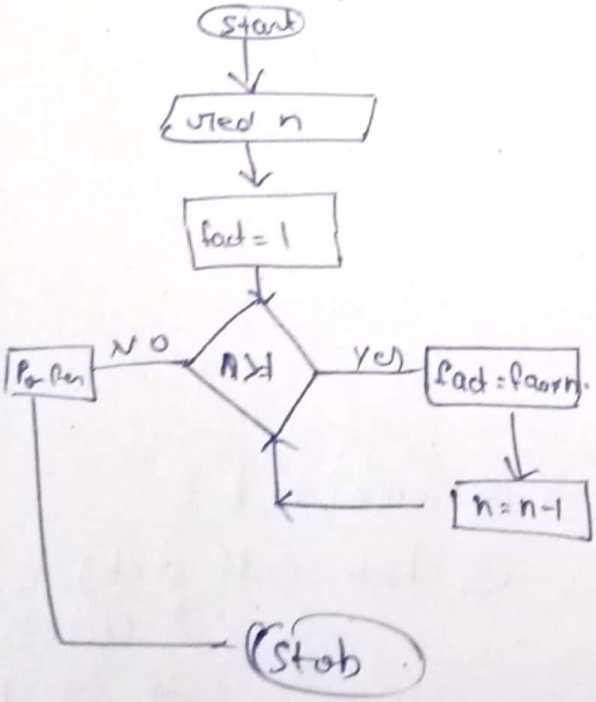
read P, R, T
S.I = P*R*T/100
Print S.I

④ counting from N to 1



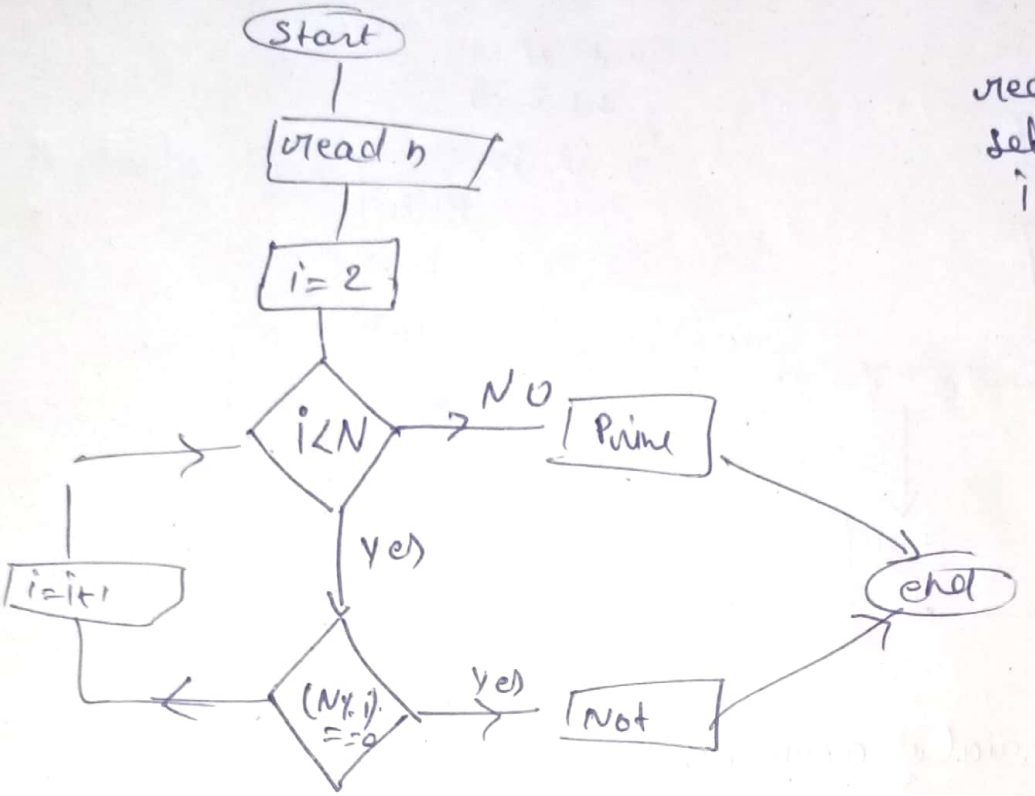
1. read N
   set i=N
2
3 if i>=1
   Print i
   i=i-1
   goto 3
4. else stop

⑤ Find factorial of a number.



① Read n
② fact = 1
③ if N>=1
   fact = fact * n
   n = n - 1
   goto 3
Print Print fact

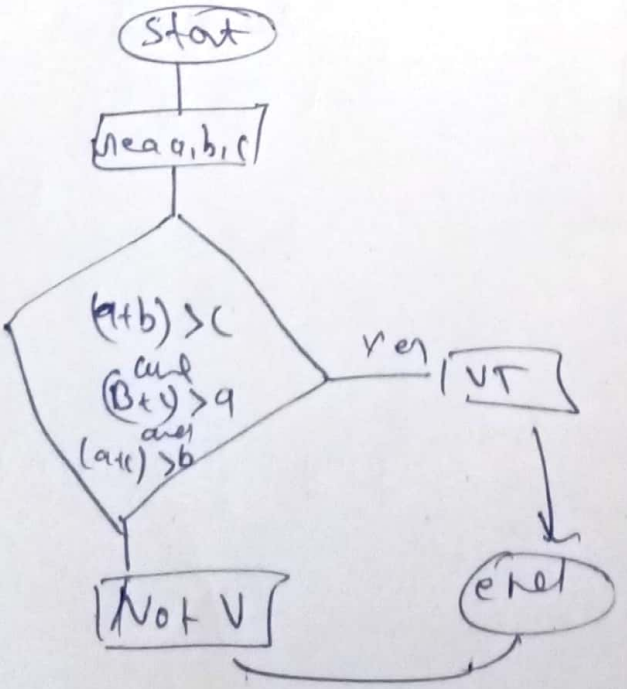
⑥ Check a number is Prime or not.



read n
Set i=2
if i<n
if N%i-2=0
Print Not
Not 5
else i=(+)
Not}
Print Pri
End

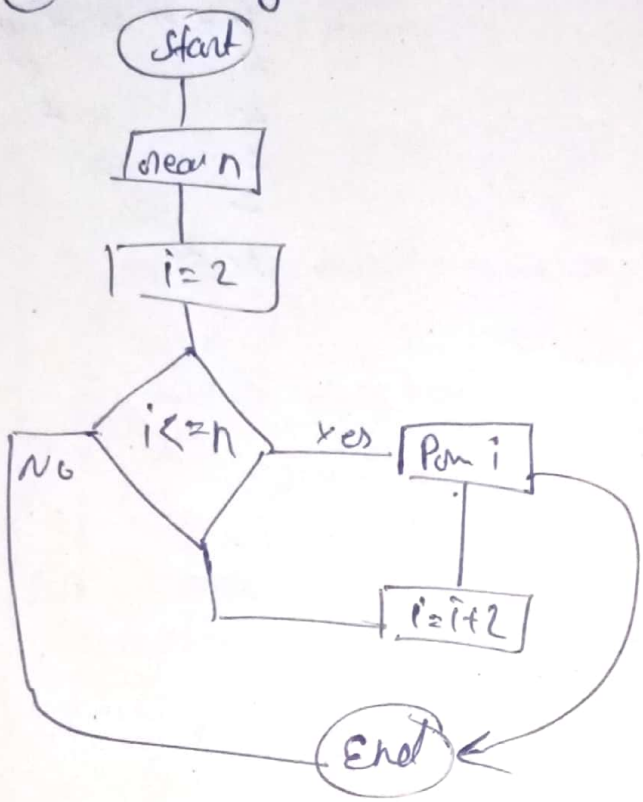
⑦ Check given triangle is valid or not

(a+b) > c
(b+c) > a
(a+c) > b



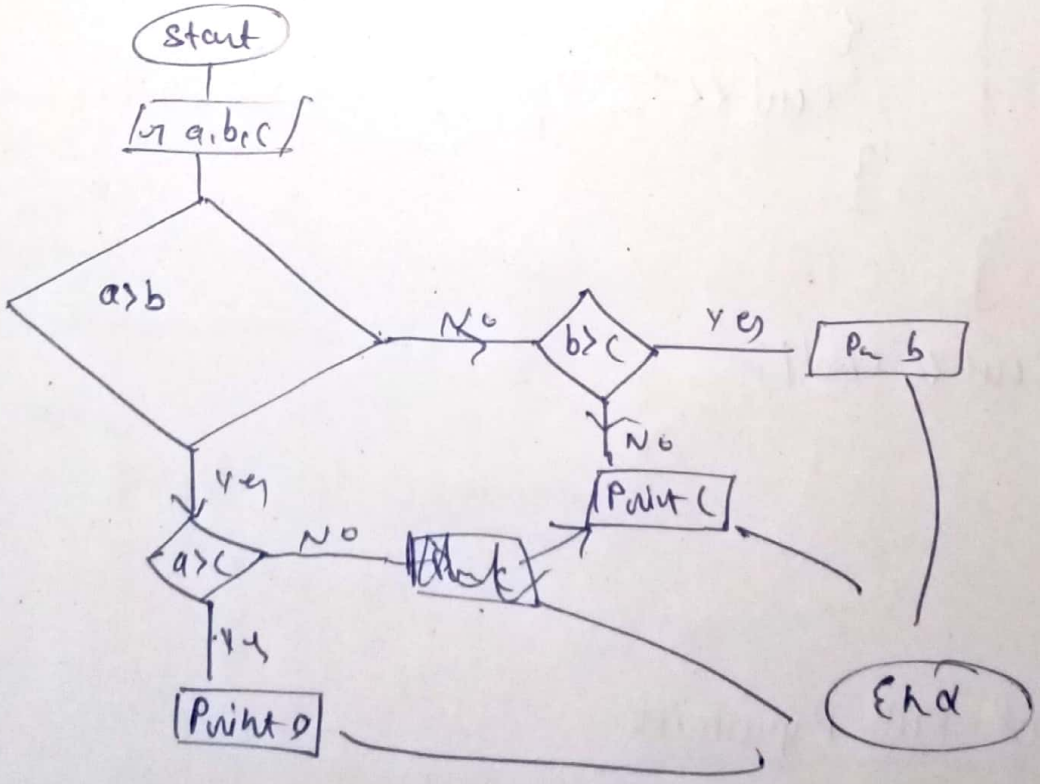
① read a, b, c
② check cond (a+b)>c
& (b+c)>a
& (a+c)>b
Pri val
Not o ⑨
③ Pri Not val
④ End

⑧ Print only even number from 1 to N



① read n
② st i=2
③ if i<=n
Prin i
i=i+2
goto 3
④ ~~Print~~ End

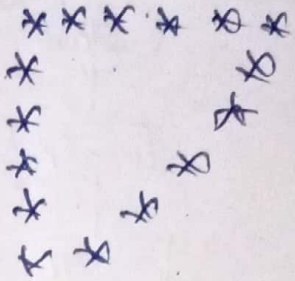
⑨ Print maximum of three numbers



① read a,b,c
② if a>b
A. a>c
APrint a
B. Print c
Else if b>c
Prn c
else
Print c

→ Solid Square Pattern

→ Hollow inverted Half Pyramid



```
int main()
  {
    int n;
    cin>>n;
  for(int i=0; i<n; i++)
  {
  for (int j=0; j<n; j++)
      {
      if(i==0 || j==0 || j==n-i-1)
          {
            cout<<" * ";
          }
      else
          {
          cout<<"_ ";
          }
      }
    cout<<endl;
    }
}
```

→ Hollow & full Pyramid

Full Pyramid

```
int main()
{
  int n;
  cin >> n;
  for(int row=0; row<n; row=row+1)
    {
      //space
      for (int col=0; col<n-row-1; col=col+1)
        {
          cout << " ";
        }
      // stars
      for(int col=0; col<row+1; col=col+1)
        {
          cout << "* ";
        }
      cout << endl;
    }
}
```

n-(row+1)

Inverted Full pyramid

```
int n;
cin >> n;
for ( int row = 0; row<n; row++)
  {
    //space
      for( int col=0; col<row; col+1)
          {
            cout << "_";
          }
      // star
      for(int col=0; col<n-row ; col++)
          {
            cout<<"* ";
          }
      cout << endl;
  }
```

Solid Diamond

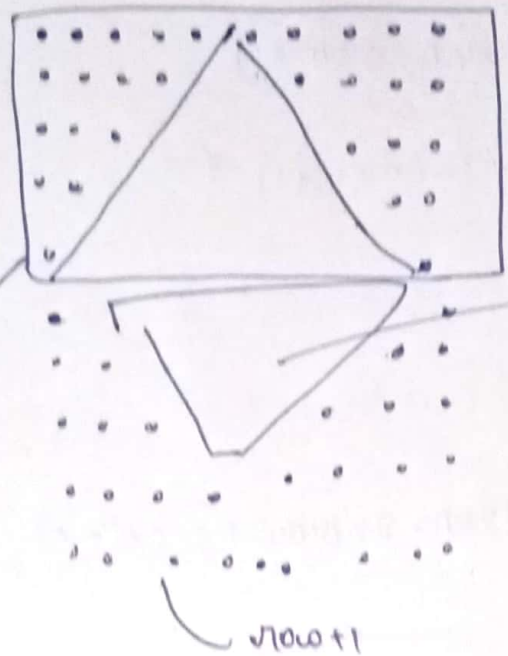
Full + Reverse

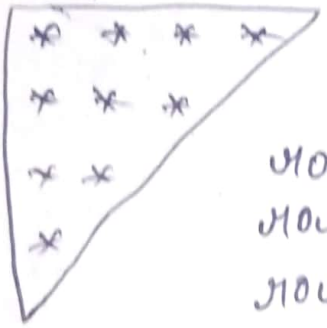
Hollow Diamond

```
int n;
cin >> n;
for (int row = 0 ; row < n; row = row+1)
  {
  //spaces
  for(int col=0; col<n-row-1; col++)
     {
       cout << " ";
     }

  // stars
  for (int col = 0; col < 2*row+1; col++)
     {
       if (col == 0)
         {
           cout << "* - ";
         }
       else if (col == 2*row)
         {
           cout << " * ";
         }
       else
         {
           cout << " - ";
         }
     }
  cout << endl;
}
```

```
for( int row=0; row<n; row=row+1)
 {
 for (int col=0; col<row ; col=col+1) -space
    {
      cout <<"   ";
    }

   //st
    for (int col=0; col<2*n-2*row+1 ; col++)
       {
```

Plipped Solid Dia





n = 4

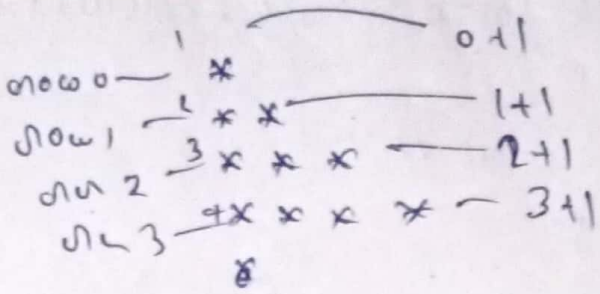
row 0 → 4 — n
row 1 → 3 — n-1 (n-row)
row 2 → 2 — n-2
row 3 → 1 — n-3


```
int n;
cin>>n;
for(int row=0; row<n ; r++)
{ //half
  for(int col=0; col<n-row; col++)
  {
    cout<<"*";
  }
//space wall
```

```
for (int col=0; col< 2*row+1;   col++)
    {
      cout << " - ";
    }

// half pyr
for ( int col= 0; col<n-row; col++)
   {
     cout << "* ";
   }
  cout << endl;
}

for (int row= 0; row<n; row=row+1)
  {
    for (int col=0; col<=col+1; col=col+1)
      {
        cout << "* ";
      }
     for (int col=0; col<2*n-2*row-1 ; col++)
      {
        cout << " ";
      }
    for (int col= 0; col<row+1 ; col++)
        {
          cout << "* ";
        }
     cout << endl;
   }
```

Fancy Pattern #2

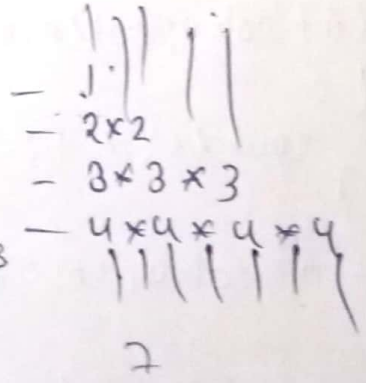
```
1
2*2
3*3*3
4*4*4*4
4*4*4*4
3*3*3
2*2
1
```



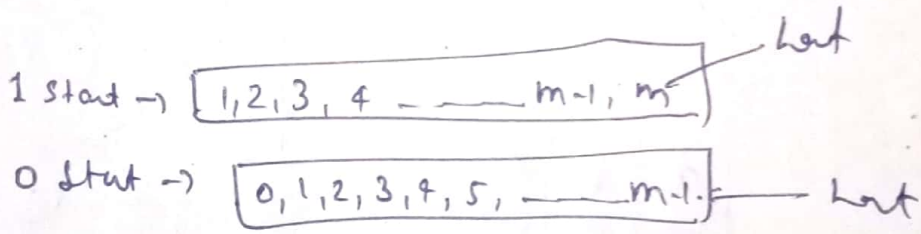

```
for (int r = 0; r<n ; r++)
{
for(int c=0;c<r+1;c++)
{
cout << "*";
}
cout << endl;
}
```

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

```
for (int row = 0; row<n; row++)
{
for( int col=0; col<row+1; col++)
{
  cout<<row+1;
  if (col!=row)
      cout << "*";
}
cout << endl;
}
```



```
for (int row=0; row<n; r++)
{
for (int col=0; col<n-row ; col=col+1)
 {
   cout << n-row;
        cout if (col!=n-row-1)
                cout << " ";
            }
   cout << endl;
}
```

A B C D C B A

A B C D E D C B A

L-5 Operators, loops & conditionals

* Bitwise Operators :-

- AND → &

a&b

a	b	O/P
0	0	0
0	1	0
1	0	0
1	1	1

- OR → |

a	b	O/P
0	0	0
0	1	1
1	0	1
1	1	1

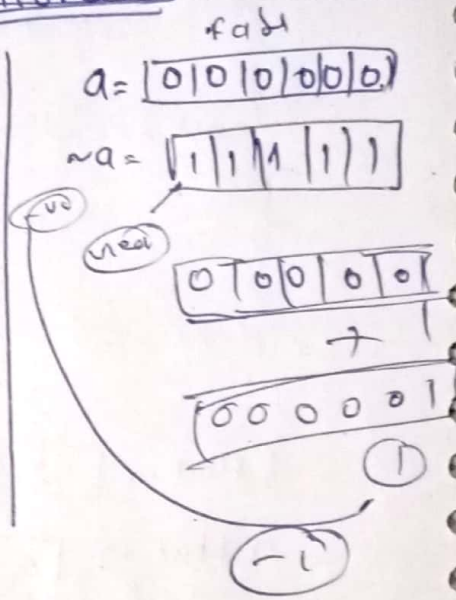
- NOT → !

a	O/P
0	1
1	0

- XOR →

a^b

a	b	O/P
0	0	0
0	1	1
1	0	1
1	1	0



2 = 10 (2 ^ 3)
3 = 11
10 ⊕ 2

5 = 101 (5 ^ 10)
10 = 1010
0101
1010
0000 → 0

(3 | 4)
3 = 11
4 = 100
011
100
111 → 7

* Left Shift (<<):

a = 5

0000 —— 101 → 5
↓ <<
0000 —— 1010 → 10
↓ <<
000 —— 10100 → 20

000 —— 001 → a = 1
<<1 ↓ ×2
000 —— 0010 → 2
<<1 ↓ ×2
000 —— 00100 → 4
<<1 ↓ ×2
000 —— 001000 → 8

>> * Right shift (>>)

a = 8 → 000 —— 1000
a >> 1
→ 000 —— 100
↓ 4

int a = 12
a = a >> 1
cout << a << endl
↓ 6

1100
110

-1
111111 —— 1011

agar kisi -ve number par hum Right shift >> karte to wo ek high value +ve no de sakta hai

0110 —— 101

← high
→ low

* Pre / Post → Increment / Decrement operators

Pre Increment

```
int a = 5
++a → a = 6
cout << ++a ; → 6
```

Post Increment

```
int a = 5
cout << a++ ; → 5
a = 6
```

Pre decrement

```
int a = 5
cout << --a ; → 4
```

Post decr

```
int a = 5
cout << a--; → 5
cout << a ; → 4
```

```
int a = 5;
cout << (++a) * (++a);
```

* Break & Continue keyword

Break

```
for ( int i = 0; i < n; i++)
{
  cout << "Babbar";
  break;
}
cout << "Love";
```

O/P → Babbar Love

Continue

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

O/P → Nothing (empty)

* for skip any iteration

```
for ( _ _ _ )
{
  ↓↓↓↓↓ if (i == 2) cont
```

i = 0
i = 1
(i = 2)

* Variable Scoping

(Local Variables)

```
int main()
{
  int a;
  int b=5;
  b=10;
  if (true) {
     cout<< b << endl;
  }
  cout << b << endl;
```

→ self modification for better understanding -

* Operator precedence

→ Use brackets for best practice -

* Switch Case

```
switch (exp)
{
 case 1: ____
         break;
 case 2: ____
         break;
         ...
         ...
 default : ____
           ____
}
```

```
int val;
cout<< "Enter the value " << endl;
cin >> val;
switch (val) {
 case 1: cout << "Love";
   break;
 case 2: cout << "Babba";
   break;
 case 3: cout << "Ramesh";
   break;
 default : cout << "Hello";
}
```

L-6 Functions & Some Problems

⇒ linked with a well-defined task.

⇒ Benifits re-use same code again & again.

⇒ Avoids lengthy or Bulky code, Readeblelity.

Syntax

Input Parameters

int, void, float, Bool etc — Return type — function name ( )

{

~~~~~ } {Body}

}

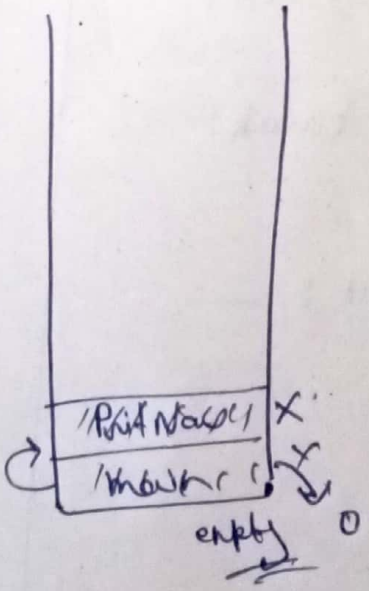
* Function call Stack

→ fuction call

→ kis fuct M dusra fuct ko call kra hai

→ fuct K kon kon se local variable h

→ fuct kya return kra hai


~~~~~

Qus write a function to add 2 numbers.

```
int main()
{
    int a;          [3]
    cin>>a;
    int b;          [4]
    cin>>b;
    int sum = add(a,b);
    cout<<"Result"<<sum<<endl;
}   return 0;       (7)
```

```
int add(int a, int b)
{
    int result = a+b;
    return result;
}
```

[3] [4] [7]

* agar hum log function ko main ke niche likh de to humko us function ko main se phele declare karna hota hai.

```
int add(int x, int y);
```

Qus Find max of 3 number.

```
int a,b,c;
cin>>a>>b>>c;
int maxnumber = FindMax(a,b,c);
cout<<maxnumber<<endl;
```

```
int FindMax(int num1,
            int num2, int num3)
{
  if (num1>num2
      && num1>num3)
  {
    return num1;
  }
  else if(
    num2>num1 && num2>num3
    {
      return num2;
    }
  else num3;
```

Q. Counting 1 to n

```
⇒ {int n;
   cout<<"Enter value"<<endl;
   cin>>n;
   PrintCounting(n);
   return 0;
  }
```

*n Actual Value

*n is a parameter

```
void printCounting(int n)
{
  for(int i=1; i<=n; i++)
  {
    cout<<i<<"-";
  }
  cout<<endl;
}
```

Q. Write a func for student Grade Problem.

```
⇒ int main()
  {
    int mark;
    cout<<"Mark"<<endl;
    cin>> marks;
    char finalGrade = getGrade(marks);
    cout<<finalGrade<<endl;
    return 0;
  }
```

```
char getGrade(int ma)
{
  if(mark >= 90)
    return 'A';
  else if(ma >= 80)
    ret "B';
  else if(m >= 70)
    re 'C';
  else if(m >= 60)
    re 'D';
  else
    rete 'E';
}
```

Q. Sum of eve no. upto N

```
int n;
cin>>n;
int ans = getsum(n);
cout<<"Sum up N"<<n<<"-"<<ans<<
return 0;
}
```

```
int getSum(int n)
{
  int sum=0;
  for(i=0; i<=n; i++)
  {
    sum = sum+i;
  }
  ret sum;
}
```

* a5 a % b = (a)
a < b = (a)

```
int getEvenSum (int n)
{
    int sum = 0;
    for (int i = 2; i <= n; i+2)
    {
        sum = sum + i;
    }
    return sum;
}
```

Homework — **Function**

1. W.A.F to display area of circle. — πr^2
2. Find Number is even or odd.
3. Find factorial of a number. — i/p → n (1-17)
4. Check Number is prime or not. — i/p → n
5. Print all prime number from 1 to N.

1 — — — — N

* **Conditional / Ternary Operator**

```
if (age >= 18)
    cout << "can Vote";
else
    cout << "Can not Vote";
```

cond ? ___ : ___

True

False

↓

~~cout << (age~~

```
String str = age >= 18 ? "Vote" : "Not";
cout << str
```

n = 234

↓

432

```
{
  int n = 234;
  while (n != 0)
  {
    int rem = n % 10;
    cout << rem << " ";
    n = n / 10;
  }
  return 0;
}
```

```
int n = 534;
for (; n != 0; n = n / 10)
{
  int rem = n % 10;
  cout << rem << "-";
}
cout << 0;
}
```

* ~~5 2 8 → 5, 2, 8~~

8	2	3	7
dig 0	dig 1	dig 2	dig 3

→ 8237

```
int digit[] = {8, 2, 3, 7};
int ans = 0;
for (int i = 0; i < 4; i++)
{
  ans = ans * 10 + digit[i];
}
cout << ans << endl;
}
```

* Count number of set bits

n = 3

000000 0000 0011

No. of set bits = 2

00000 1

& 1

1

>> 00000

0000 1

& 1

1

>> 0000

Stop

```
int n=3;
int ans=0;
while(n!=0)
{
  if(n&1)
    {
      ans++;
    }
  n=n>>1;
}
cout<<ans<<endl;
```

I/P → 3
O/P → 2

```
int km;
cin>>km;
cout<<"Value in miles is"<<(1/1.6)*km<<endl;
```

⇒ Number System - Binary & Decimal

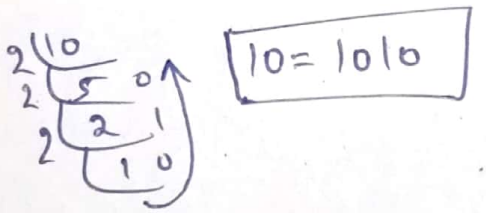
Decimal System :-

- The decimal number system has base 10.
- It uses digit from 0 to 9.
- Base. it is the number of symbols (digit) a number system uses.

Binary system :-

- Number system using base 2.
- It uses only two digits 0, and 1.

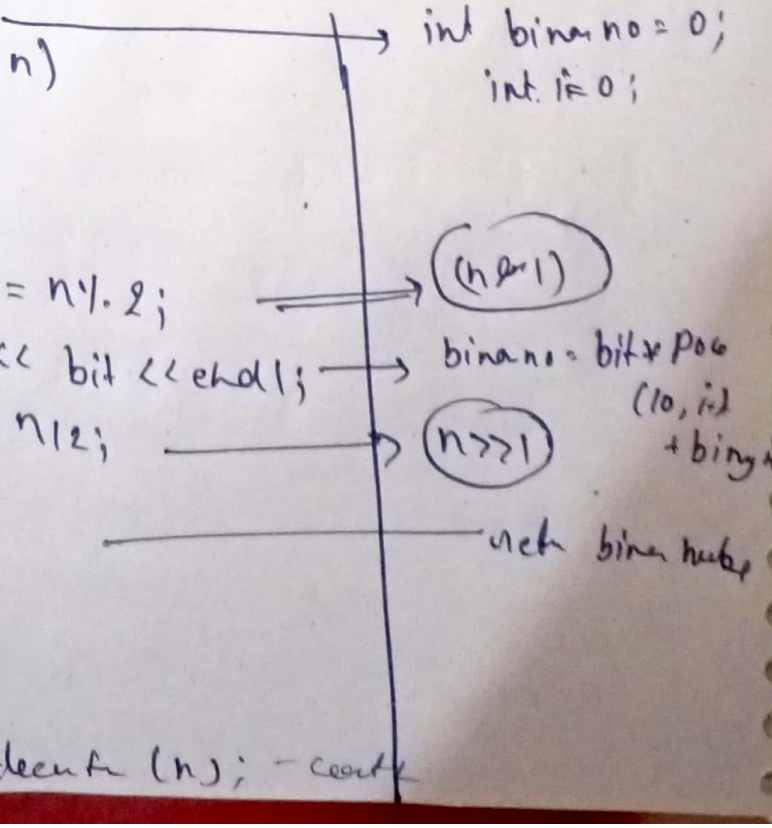
* Decimal to Binary :-



$10 \rightarrow (1010)_2$

In code —

```
int decimaltoBinary (int n)
{
  while (n>0)
  {
    int bit = n%2;
    cout<< bit <<endl;
    n = n/2;
  }
  ret 0;
}
int main()
{
  int n;
  cin>> n;
  int binary = decimal (n);
}
```



ans = (digit * 10^i) + ans

<bits/stdc++.h>

① $0 * 10^0 + 0 = 0$

② $1 * 10^1 + 0 = 10$

③ $0 * 10^2 + 10 = 10$

④ $1 * 10^3 + 10 = 1010$

alternative

n = 10

n & 1

1010

0001

0000 — ⓪ — first bit

>>

101

001

001 — ① — second bit

>>

10

01

00 — ⓪ — third bit

>>

1

1

1 — ① found bit

* Binary to Decimal :-

$10 \rightarrow (1010)_2$

123

$100 \rightarrow 1 \times 10^2$

~~0+0+0~~

~~0+0×2⁰~~

1 0 1 0

$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$8 + 0 + 2 + 0 = 10$

formula ⇒ digit * (Base)^i

In Code —

```
int binaryToDecimal (int n)
{
    int decimal = 0;
    int i = 0;
    while (n
    {
        int bit = (n % 10);
        decimal = decimal + bit * Pow(2, i++);
        n = n/10;
    }
    return decimal;
}

int main()
{
    int binaryno;
    cin >> bno;
    cout << binaryToDecimal (binaryno) << endl;
}
```

diagram 1010

dec = $0 + 0 * 2^0 = 0$

n = 101

bit = 1

dec = $0 + 1 * 2^1 = 2$

n = 10

bit = 0

de = $2 + 0 * 2^2 = 2$

n = 1

da = $2 + 1 * 2^3 = 10$

* Explore

by using bitwise

n = 10

binarynu = 0

i = 0

bit = 0

binaynu = 0 * pow(10, 0) + 0 = 0

n >> 1

bit = 1

binarn = 1 * 10^1 + 0 = 10

n >>

bit = 0

binn = 0 * 10^2 + 10 = 10

n >>

bit = 1

bin = 1 * 10^3 + 10 = 1010

2 | 10 (5
10
0

10

2) 5 (2
4
1

2

1010
01
0000 - ①

>>

101
01
001 - ①

>>

10
01
00 - ②

>>

1
1
1 - ①

1000
10

2 | 10
2 | 5
2 | 2
1 | 0

n = 1010

bit = 0

dec = 0 + 0 * 10^0 = 0

n = 101

bit = 1

dec = 0 + 2 * 10^1 = 10

n = 10

bit = 0

10) 1010 (101
10
10

dec = 0 + 0 * 2^0 = 0

n = 101

bit = 1

dec = 0 + 1 * 2^1 = 2

n = 10

bit = 0

dec = 2 + 0 * 2^2 = 2

n = 1

bit = 1

dec = 2 + 1 * 2^3 = 8 + 2 = 10