Topics covered:
→ Course Flow
→ Flow charts
→ Pseudocode

Q. How to solve a programming problem?

→ Given the problem,

① Understand the problem. → Add 2 numbers

② Check the given values. → 2 variables. Data Types?

③ Figure out an approach → a+b = my answer.
- This comes from practice and past coding experience.

④ Code!   → int ans = a+b; cout << ans << endl;

WARNING ⚠

Agar code samajh na aaye to

Aage sab kuch cover hoga.

Given some problem [P], say you 'think' of some solution, ki aise aise karenge, etc. Now write down this crude solution on paper, not necessarily in correct syntax (code ki bhasha). Now your idea is on paper. Convert this rough work, also called 'pseudocode' into a program in a programming language of your choice, say C++.

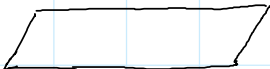Pseudocode: A very simple and high-level (upar-upar ka) form of computer language that is used in program design.

A Flowchart is a diagramatic representation of an approach. This draws out all the steps of your approach in order.
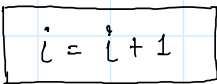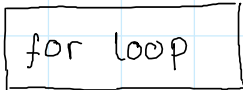
Components:
① Terminator : Specifies the start and end of a program.

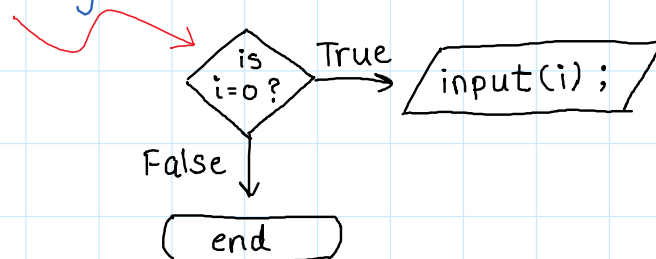$$(\text{Start/End}) \rightsquigarrow \text{Terminator}$$

② Parallelogram: For taking input or showing output.

Input / Output

③ Process : Operations and processes ke liye.

$i = i + 1$ or   for loop
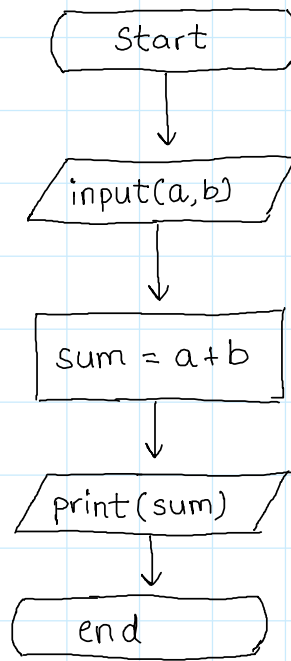
④ Decision Making : (Diamond Shape)

is i=0? → True → input(i);

False ↓

end

⑤ Circle : Connectors (To be covered when we discuss functions / methods)

⑥ Arrows: Code ka pravaah dikhane ke liye.
(Upar flowchart dekho ↑)

EXAMPLE :

Flow chart for adding 2 numbers.

```
        ( Start )
            |
            v
      /input(a,b)/
            |
            v
     | sum = a+b |
            |
            v
     /print(sum)/
            |
            v
        ( end )
```

Pseudocode for adding 2 numbers:

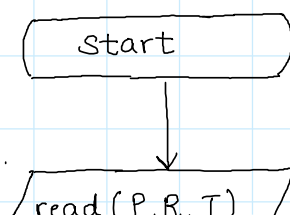→ input 2 numbers a and b.

→ Let sum = a+b

→ Print out sum

→ Read a and read b.

→ sum variable is a+b

→ sum chaapo
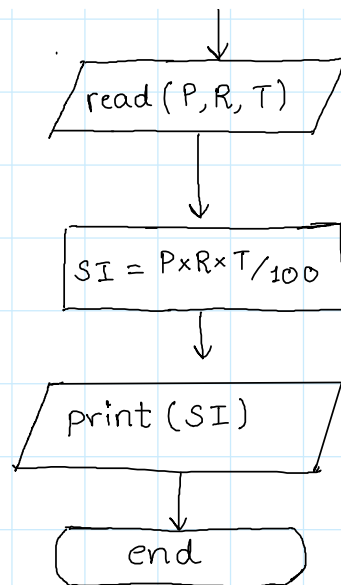
Both pseudocodes are OK. No pseudocode is wrong as long as the logic is same / similar.

EXAMPLE: Calculate simple interest.

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

$\begin{pmatrix} P - \text{Principal Amount} \\ R - \text{Rate of Interest} \\ T - \text{Time} \end{pmatrix}$

```
     ( start )
         |
         v
   /read(P,R,T)/
```

read(P, R, T)

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

print(SI)

end

Pseudocode

$\longrightarrow$ read P, R and T

$\longrightarrow$ Make $SI = P \times R \times T / 100$

$\longrightarrow$ Print SI

EXAMPLE: Determine if $a < b$

start

input(a, b)

is $a < b$?

Yes $\longrightarrow$ Print "yes"

NO

Print "No" $\longrightarrow$ End
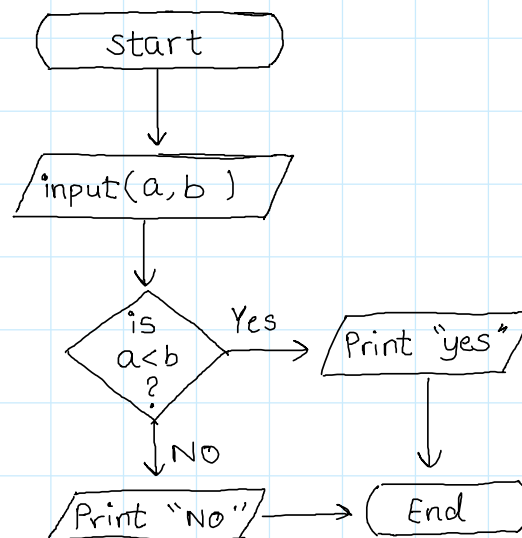
PSEUDOCODE:

$\longrightarrow$ Read a and b.

$\longrightarrow$ if $a < b$
then print Yes
else
print No

NEW CONCEPT: % (modulo) operator

Gives the remainder after division a/b.

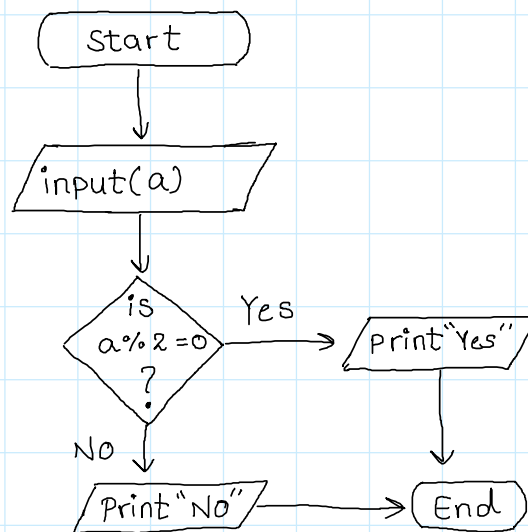$\therefore$ $a \% b$ = Remainder of a/b.
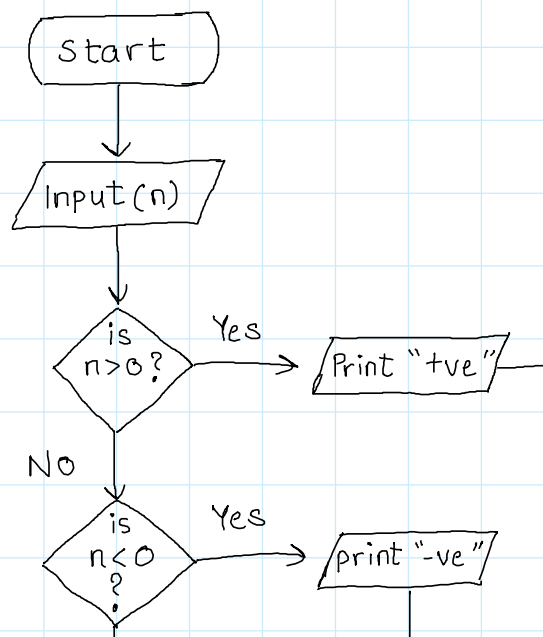
Eg: $5 \% 2 = 1$

$6 \% 4 = 2$

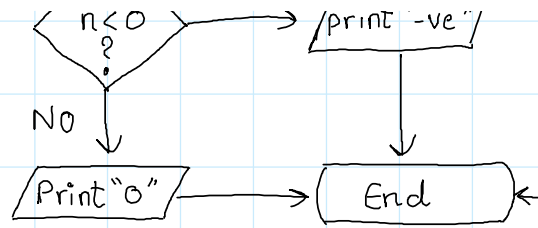$8 \% 4 = 0$

$* \ 4 \% 9 = 4$ when $a < b$, $a \% b = a$.

$\therefore$ If $n \% 2 = 0$, n is even

else n is odd.

EXAMPLE: Check if n is even or odd.

EXAMPLE: Is n positive, negative or zero.

n < 0 ? ⟶ /print "-ve"/

NO ↓

/Print "0"/ ⟶ End

Homework : Check if a given triangle is valid.

Hint - $a+b > c$, $b+c > a$ & $c+a > b$

(Start)

↓

/input(a,b,c)/

↓

is $a+b > c$? ——NO——▶ /Print "Invalid"/

Yes ↓

is $b+c > a$? ——NO——▶ /Print "Invalid"/

Yes ↓

is $c+a > b$? ——NO——▶ /Print "Invalid"/

Yes ↓

/Print "valid"/ ⟶ End
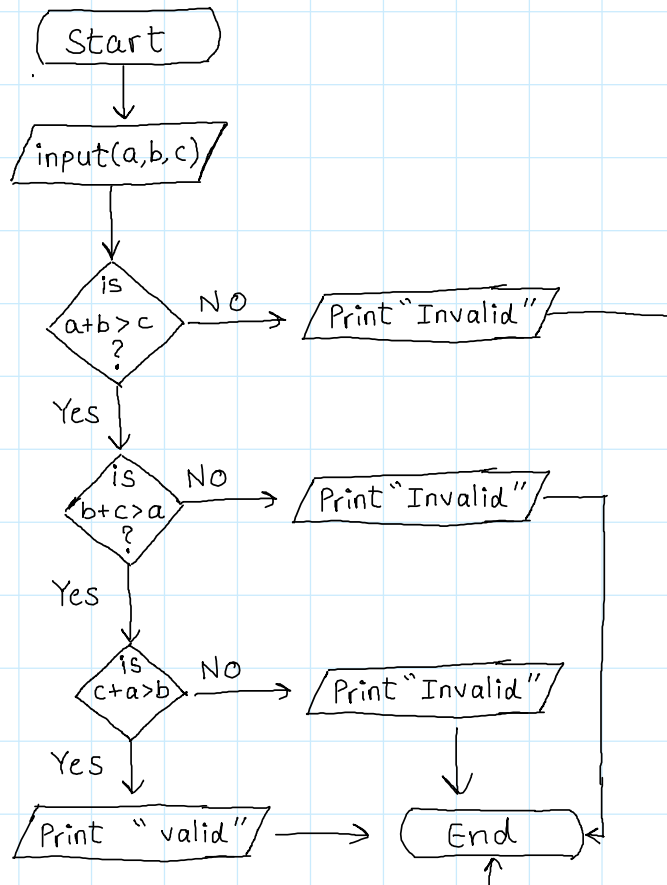
Let Variable n = 1.

Now make n go from 1 to 5.

Soln.
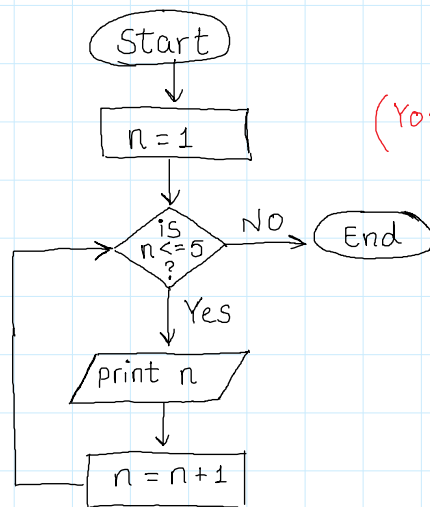
```
        ( Start )
            |
            v
      +------------+
      |   n = 1    |          (You can also give input n = 1)
      +------------+
            |
            v
         / is  \      NO
    --->(  n<=5  )--------->( End )
    |    \  ?   /
    |        | Yes
    |        v
    |   / print n /
    |        |
    |        v
    |   +----------+
    ----| n = n+1  |
        +----------+
```
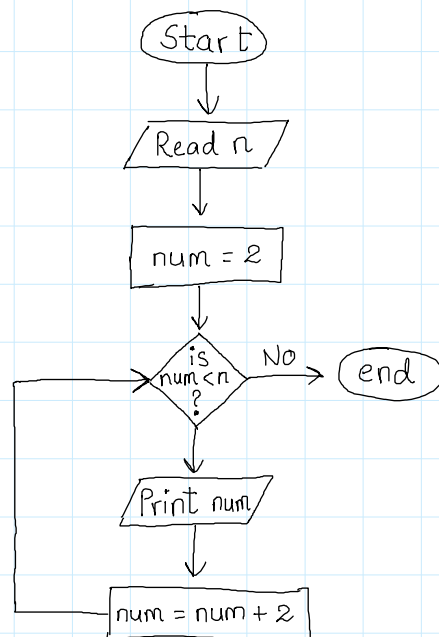
This is where we use loops to continuously perform some action while updating some value.

EXAMPLE: Print even numbers b/w 1 and n. (exclusive)
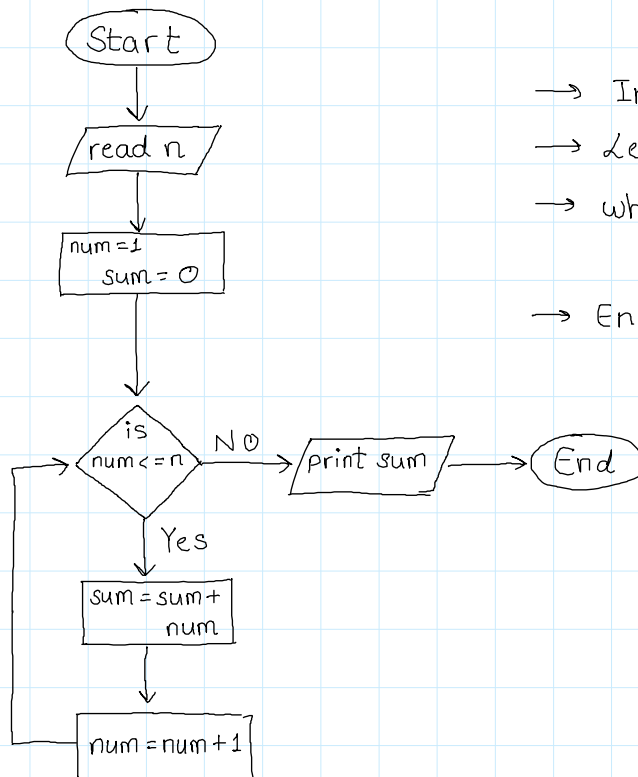Hint - Even numbers start from 2 and occur alternatively.

```
        ( Start )
            |
            v
      / Read n /
            |
            v
      +-----------+
      |  num = 2  |
      +-----------+
            |
            v
         / is  \      No
    --->( num<n )--------->( end )
    |    \  ?  /
    |        |
    |        v
    |   / Print num /
    |        |
    |        v
    |  +--------------+
    ---| num = num + 2|
       +--------------+
```

Pseudocode
→ Input(n)
→ Let num = 2
→ while num < n,
       print num
       num = num+2
→ End

EXAMPLE: Print all odd numbers from 1 to n (inclusive)
(Homework hai)

→ Input (n)

→ Let a = 1

→ while a < 1, print(a)

  a = a + 2.

→ End

EXAMPLE : Find sum from 1 to n.

Start

read n

num = 1
Sum = 0

is num <= n ——NO→ print sum → End

Yes

sum = sum + num

num = num + 1
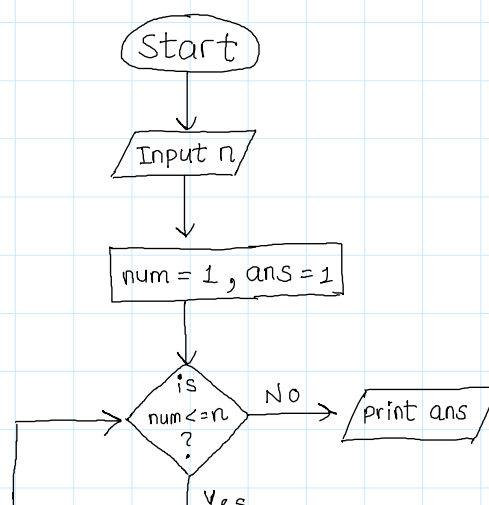
Pseudocode

→ Input (n)

→ Let num = 1 and sum = 0

→ while num <= n,

  sum = sum + num
  num = num + 1

→ End

Homework : Find n!

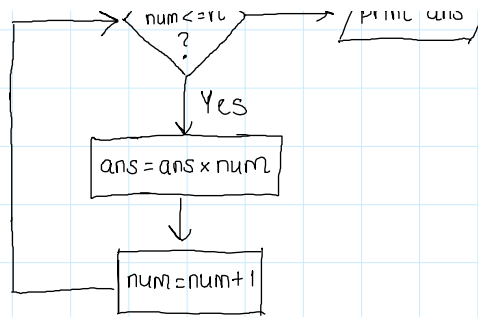Hint — n! = n × (n-1) × (n-2) × .... × 3 × 2 × 1

Start

Input n

num = 1, ans = 1

is num <= n ? ——No→ print ans

Yes

Pseudocode

→ Input (n)

→ Let num = 1 and ans = 1

→ while num <= n,

  ans = ans × num
  num = num + 1

→ End

num <= n ? → print ans

Yes

ans = ans × num

num = num + 1

Example : Check if n is prime.

Input : 5   Input : 9
Output : Yes   Output : No

Prime n :   n % (any number from 2 to n-1) ≠ 0 → Written as
                                                      != in C++
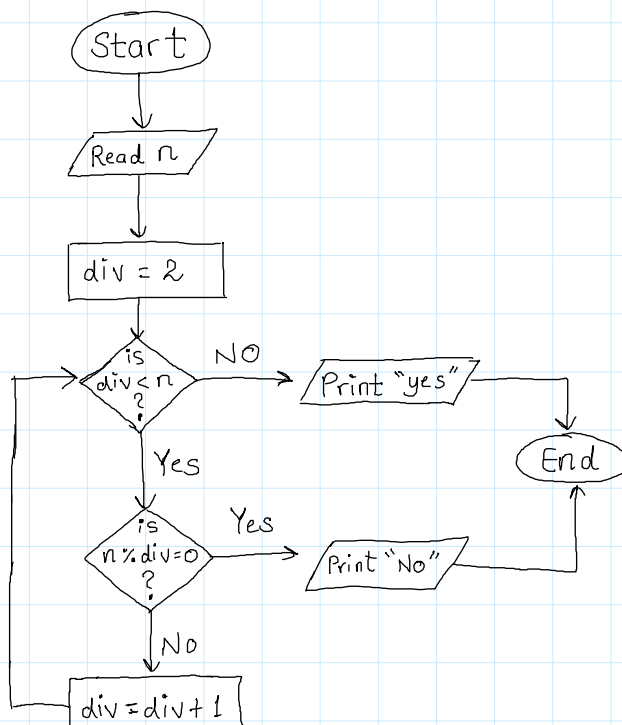
Start

Read n

div = 2

is div < n ? — NO → Print "yes"

Yes

is n % div = 0 ? — Yes → Print "No"

No

div = div + 1

End

What is a Programming Language ?

→ Jaise Khaana khaane ke liye mummy/papa ko bolna padhta hai, vaise hi we must instruct our computer to perform some task for us.

→ More formally, a programming language is a way to communicate with a computer. It is a formal language which consists of sets of strings that produce various kinds of machine output.

Eg: C, C++, Java, Python, R, GO, etc.

```cpp
#include<iostream>

using namespace std;

// main function -
// where the execution of program begins
int main()
{
    // prints hello world
    cout<<"Hello World";

    return 0;
}
```

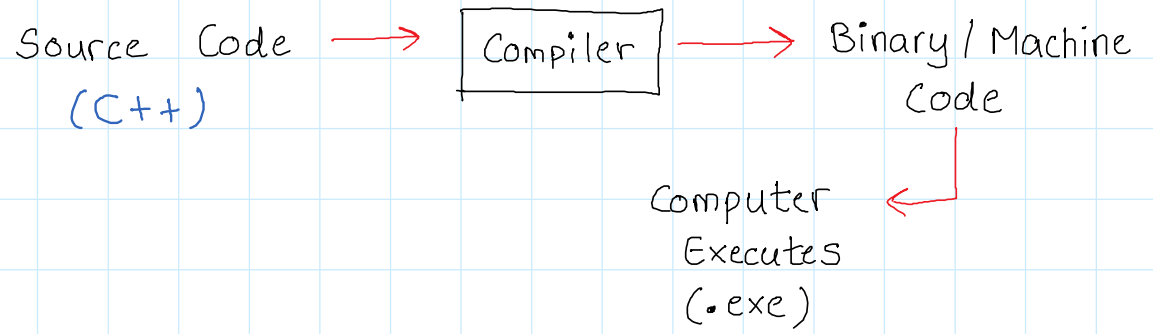⇒ This code in C++ instructs our computer to print "Hello World" on our screen.

```
Output:

    Hello World
```

Source: https://www.geeksforgeeks.org/writing-first-c-program-hello-world-example/

→ Every language must be written following some rules called syntax of that language.

→ A computer essentially only understands binary codes of 0s and 1s. A compiler processes the statements of a programming language into Machine Code (Binary).

Source Code ──→ [Compiler] ──→ Binary / Machine

Source Code $\longrightarrow$ | Compiler | $\longrightarrow$ Binary / Machine
(C++) Code

Computer $\longleftarrow$
Executes
(.exe)

Read more about Compilers here: https://www.geeksforgeeks.org/introduction-to-compilers/