

Lecture 1: Intro to Programming & Flowcharts:

(14-01-2023)

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 → $a+b = \text{my answer}$
 - This comes from practice and past coding experience.

④ code?

→ `int ans = a+b;`

`cout << ans << endl;`

Given some problem [P], say you 'think' of some solution, ki aise aise Kareenge, 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 (uppar-upar ka) form of computer language that is used in program design.

A Flowchart is a diagrammatic representation of an approach. This draws out all the steps of your approach in order.

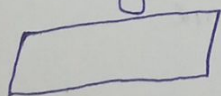
Components:

① Terminator: Specifies the start and end of a program

`Start / End`

→ Terminator

② Parallelogram: For taking input or showing output.



Input / Output

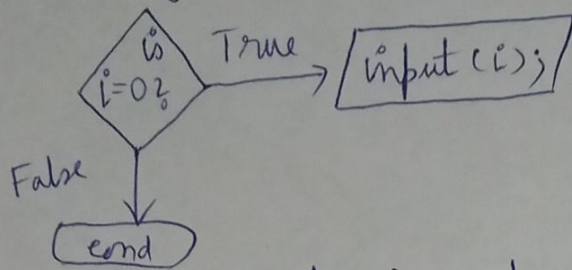
③ Process: Operations and processes ki liye

`i = i + 1`

or

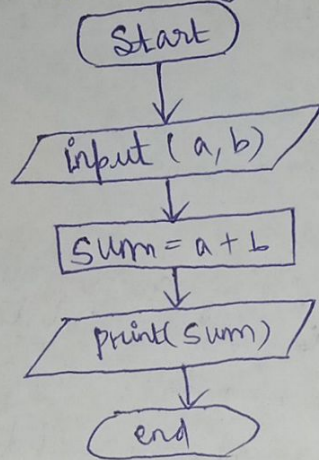
`for loop`

④ Decision Making: (Diamond Shape)



- ⑤ Circle: connectors (To be Covered when we discuss functions/methods)
- ⑥ Arrows: Code Ka pravaah dikhane ke liye.
(upar flowchart dekho ↑)

Example: Flowchart for adding 2 numbers



Pseudocode for adding 2 numbers:

- input 2 numbers a and b
- let sum = a + b
- Print out sum

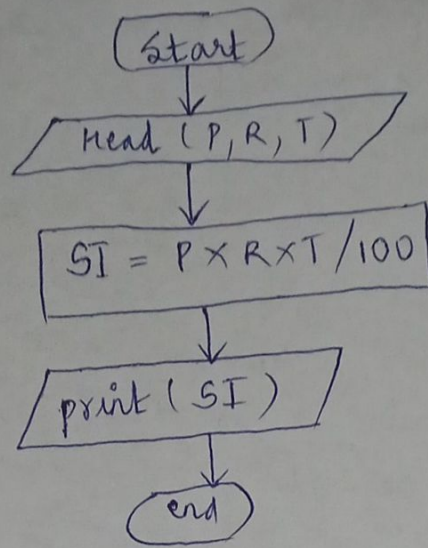
- Read a and Read b
- sum variable is a + b
- sum chaspo.

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

Example: Calculate simple interest.

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

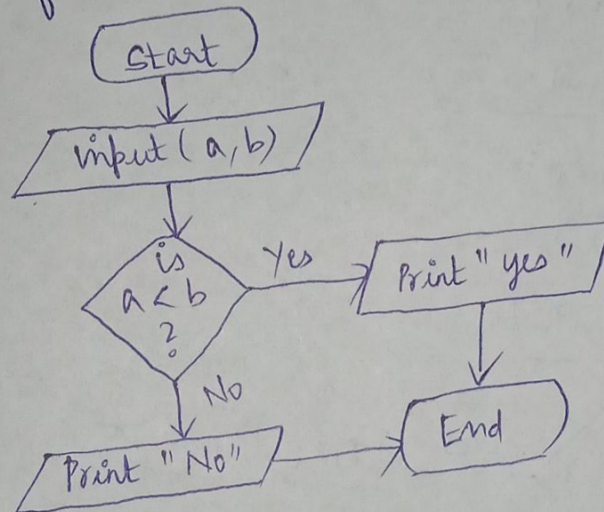
(P - Principal Amount
R - Rate of Interest
T - Time)



Pseudocode

- Read P, R and T
- Make $SI = P \times R \times T / 100$
- Print SI

Example: Determine if $a < b$



Pseudocode:

- Read a and b
- if $a < b$
 then print Yes
else
 print No

New Concept : % (modulo) operator

Gives the remainder after division a/b

$\therefore a \% b = \text{Remainder of } a/b$

Eg: $5 \% 2 = 1$

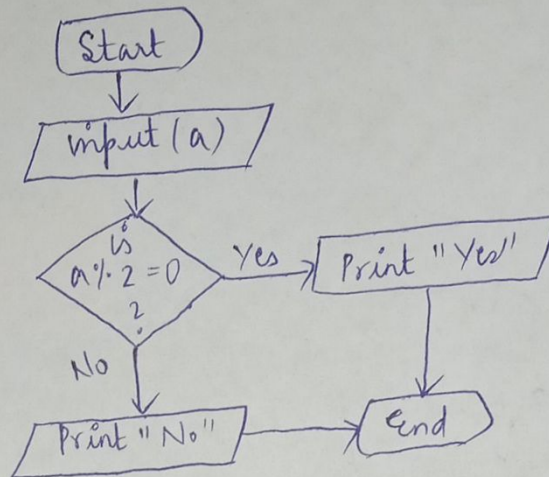
$6 \% 4 = 2$

$8 \% 4 = 0$

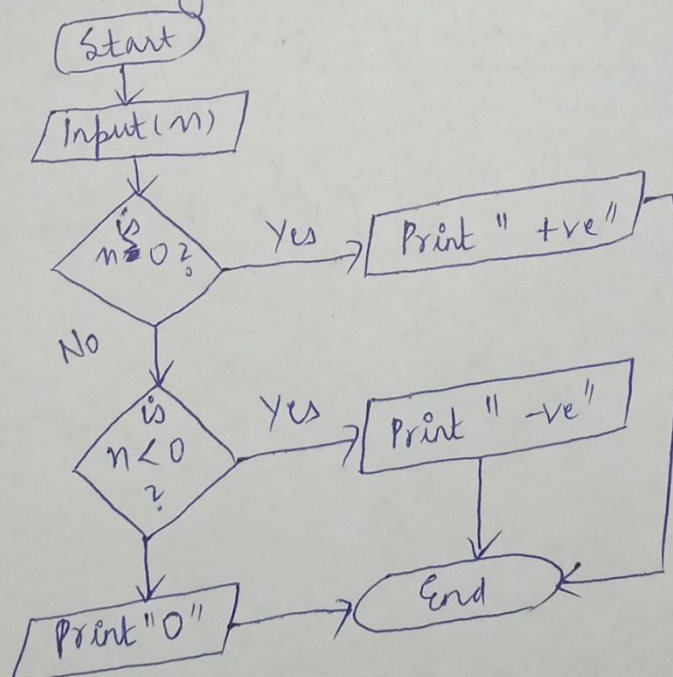
* $4 \% 9 = 4$ when $a > b$, $a \% b = a$

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

Example: Check if n is even or odd



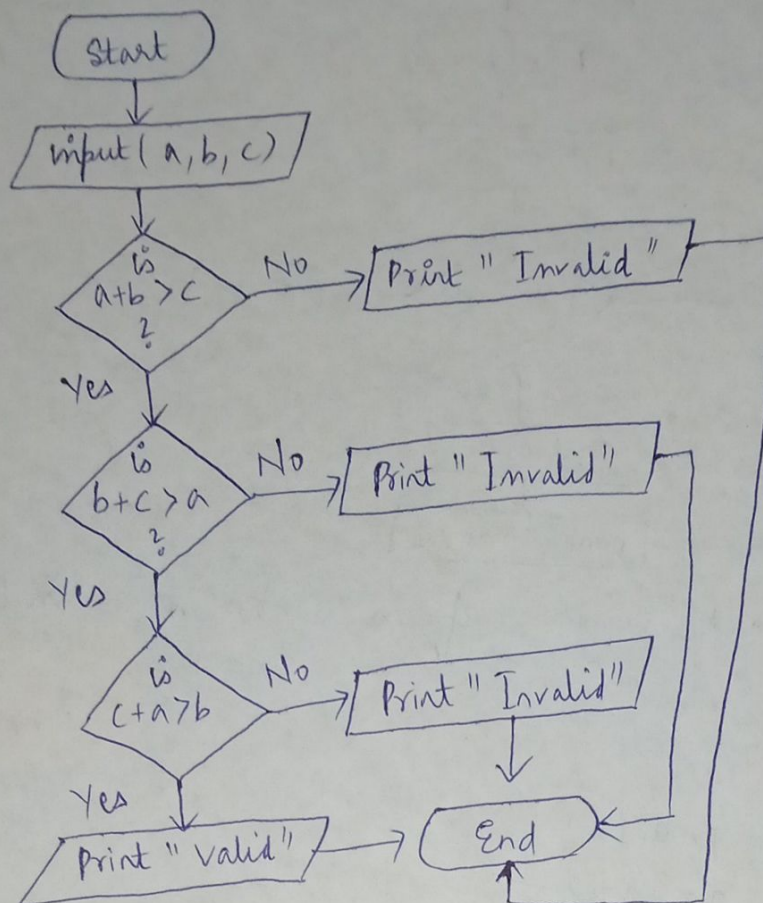
Example: Is n positive, negative or zero



Homework: Check if a given triangle is valid

(3)

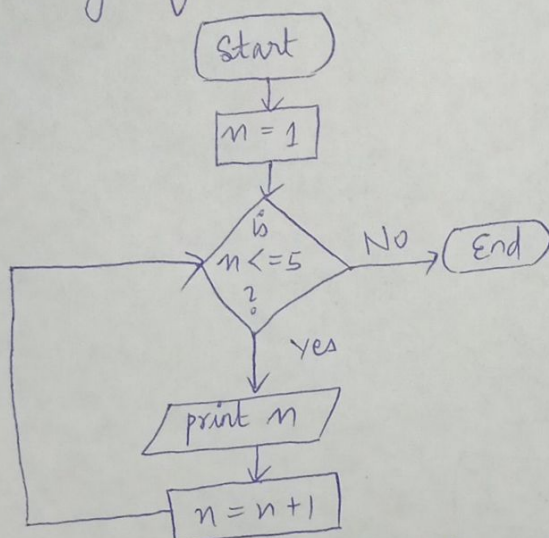
Hint - $a+b > c$, $b+c > a$ & $c+a > b$



Loops:

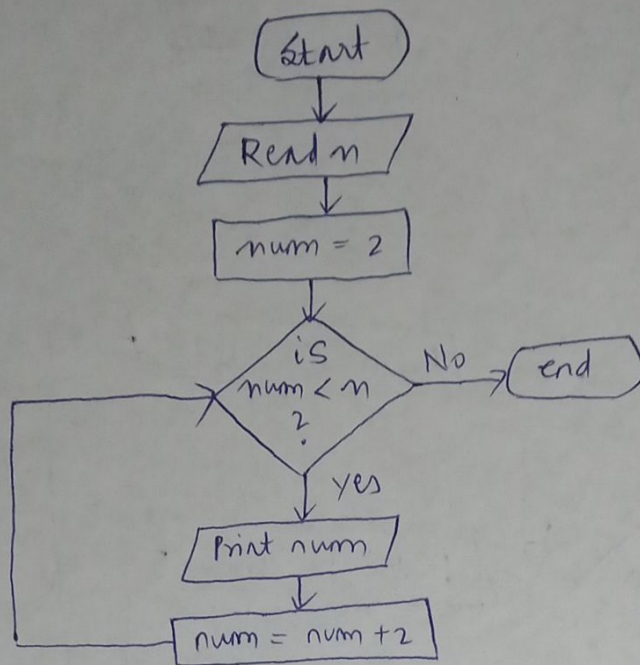
Let variable $n = 1$

Now make n go from 1 to 5



This is where we use loops to continuously perform some action while updating some value.

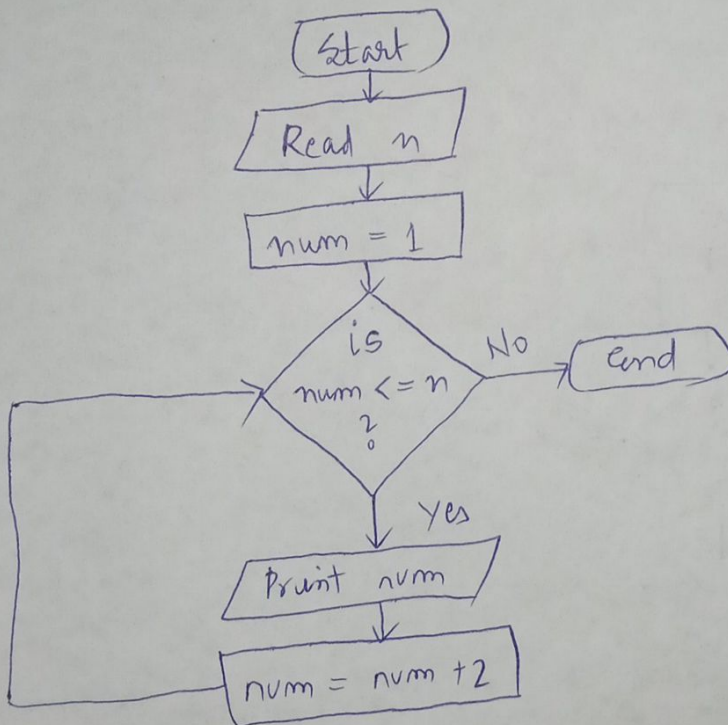
Example: Print even numbers b/w 1 and n (exclusive)
Hint → Even numbers start from 2 and occur alternatively



Example: Print all odd numbers from 1 to n (inclusive)

Pseudocode

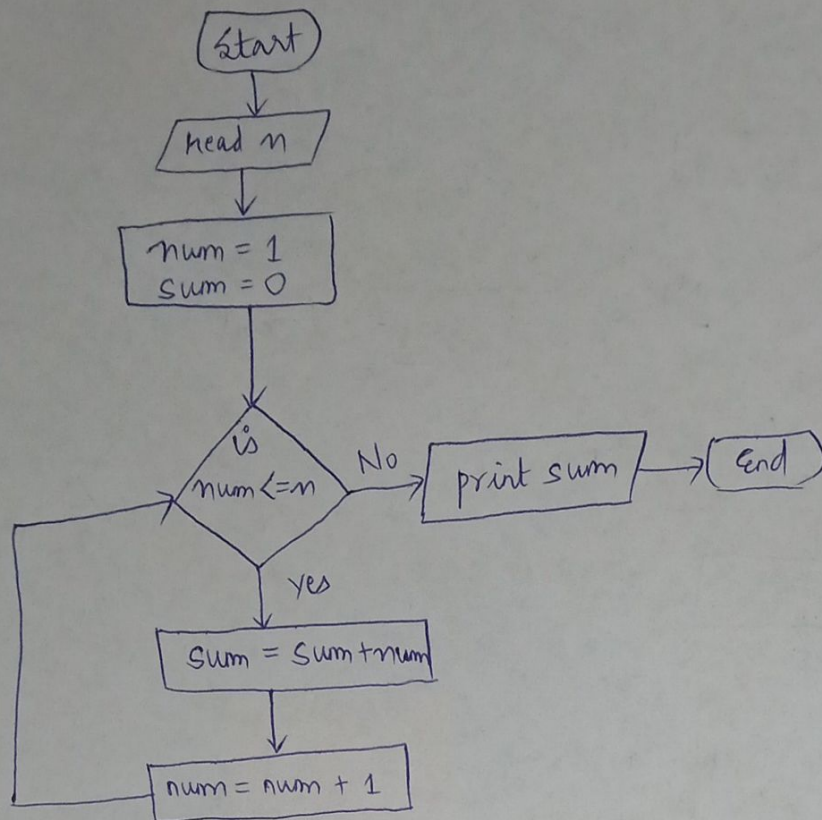
→ Input (n)
 → let $a = 1$
 → while $a < 1$, print (a)
 $a = a + 2$
 → end



Example: Find sum from 1 to n .

Example: Find sum from 1 to n

(4)



Pseudocode

- Input (n)
- let num = 1 and sum = 0
- while num ≤ n, sum = sum + num
num = num + 1
- End

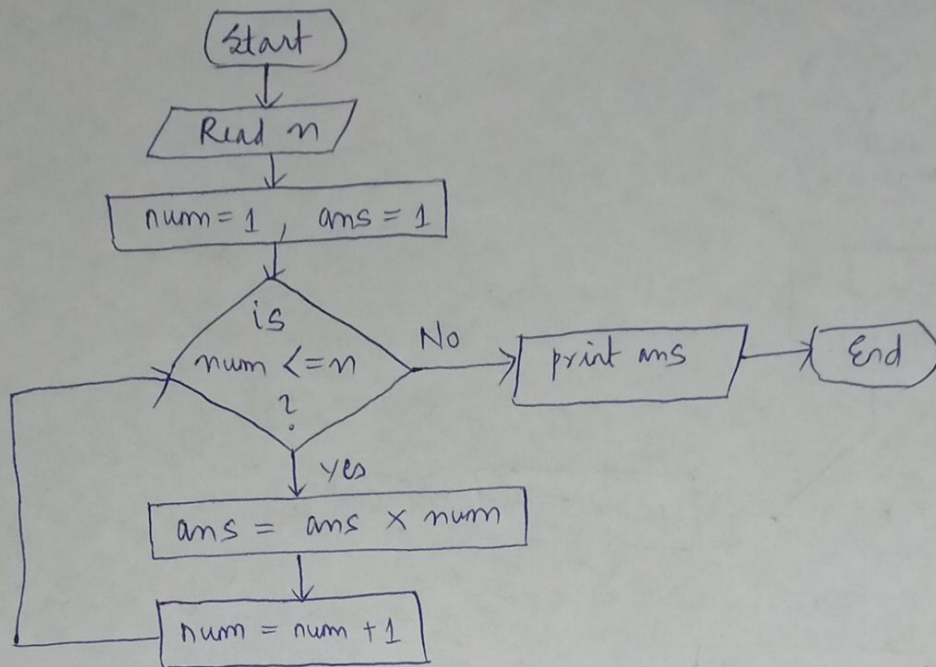
Homework Find n!

Pseudocode

- Input (n)
- let num = 1 and ans = 1
- while num ≤ n
ans = ans × num
num = num + 1

→ End

Hint - $n! \Rightarrow n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$



Example: check if n is prime

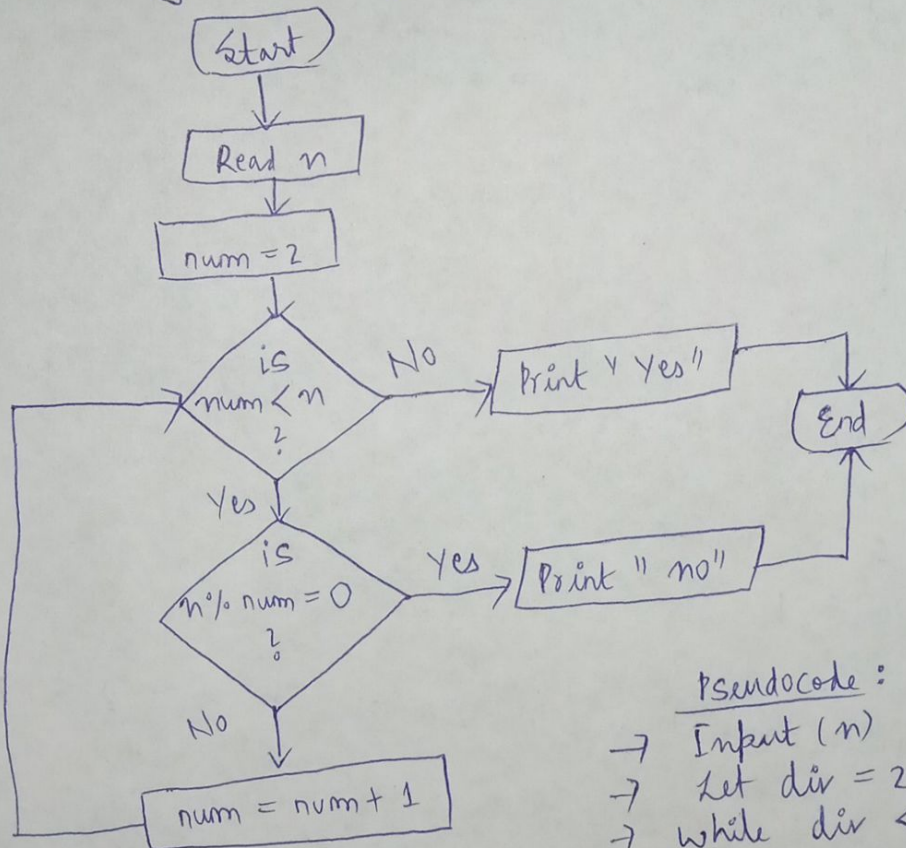
Input: 5

Output: Yes

Input: 9

Output: No

Print n : $n \%$ (any number from 2 to $n-1$) $\neq 0$ written as $!=$ in C++



Pseudocode:

→ Input (n)
 → Let $div = 2$
 → while $div < n$
 if $n \% div = 0$
 Print "no"
 Exit

else

$div = div + 1$

→ Print "Yes"

Programming Language

(5)

What is a Programming Language?

- Jaise Khana Khane ke liye mummy/papa ko bolna padhta hai, vaise hi kisi 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.

```
#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

- 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)

