1:- Intro. to Programming & Flowcharts

Topic 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 << end ;

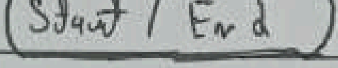
Given some problem $\boxed{P}$, say you 'think' of some solution, ki aise aise karenge, etc. Now write down this crude solution on paper, not necessarily in correct syntax (code ki bhasha). Now your idea is on paper. Convert this rough work, also called 'pseudocode' into a program in a programming language 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 diagrammatic representation of an approach. This draws out all the steps of your approach in order

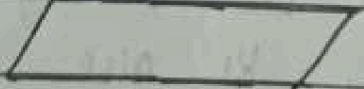
Components :-

① Terminator :- Specifies the start and end of a program.



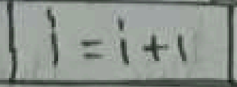
-> Terminator

② Parallelogram :- For taking input or showing output

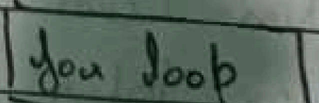


Input / Output

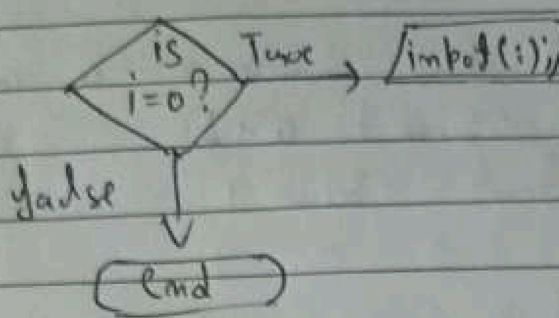
③ Process :- Operations and processes ke liye



or



(4) Decision Making ÷ (Diamond Shape)

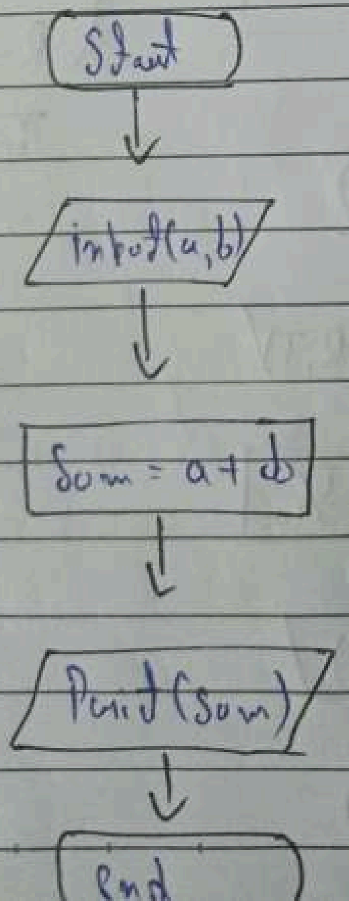


(5) Circle ÷ Connectors (To be covered when we discuss function / methods)

(6) Arrows ÷ Code ka Pravah dikhane ke liye (upar flowchart dekho) ↑

Example ÷

Flowchart for adding two numbers



Pseudocode for adding 2 Numbers :

Ye bhi theek hai

```
→ input two numbers a and b.
→ let Sum = a+b
→ Printout Sum
```

Ye bhi theek hai.

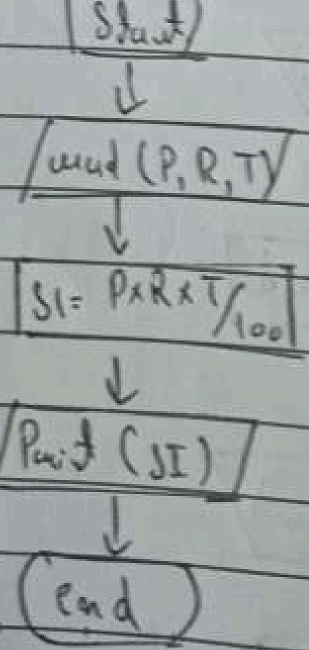
```
→ Read a and read b
→ Sum variable is a+b
→ Sum Chhapoo
```

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

Example : Calculate Simple Interest

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

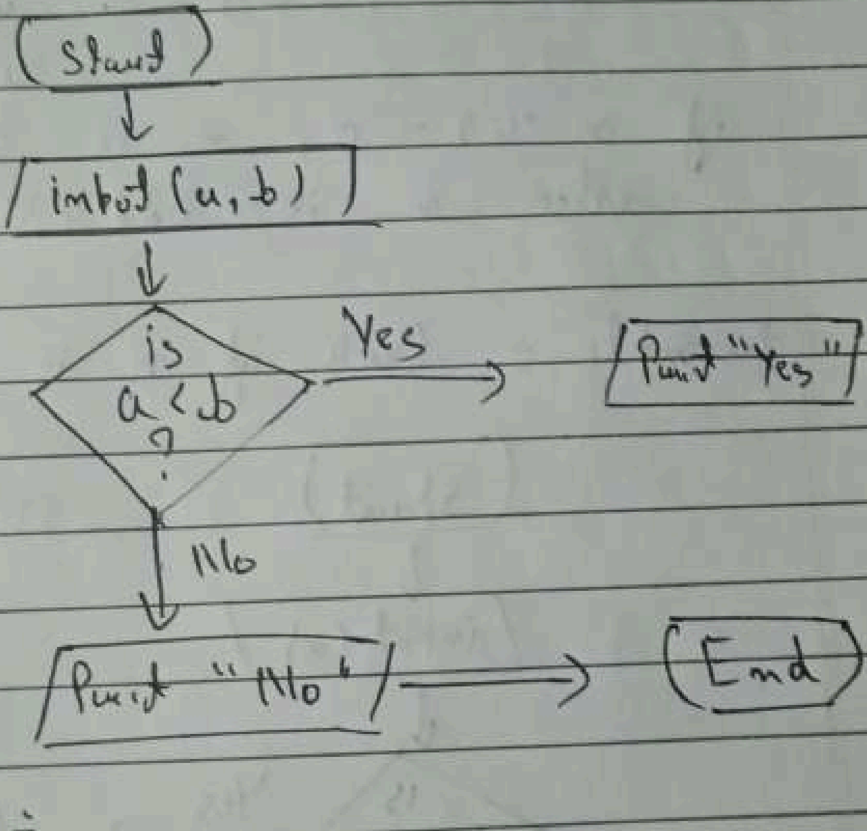
(P- Principal Amount
R- Rate of Interest
T- time



Pseudocode

```
-> read P, R and T
-> make SI = P × R × T / 100
-> Print SI
```

Example : Determine if a < b



Pseudocode :

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

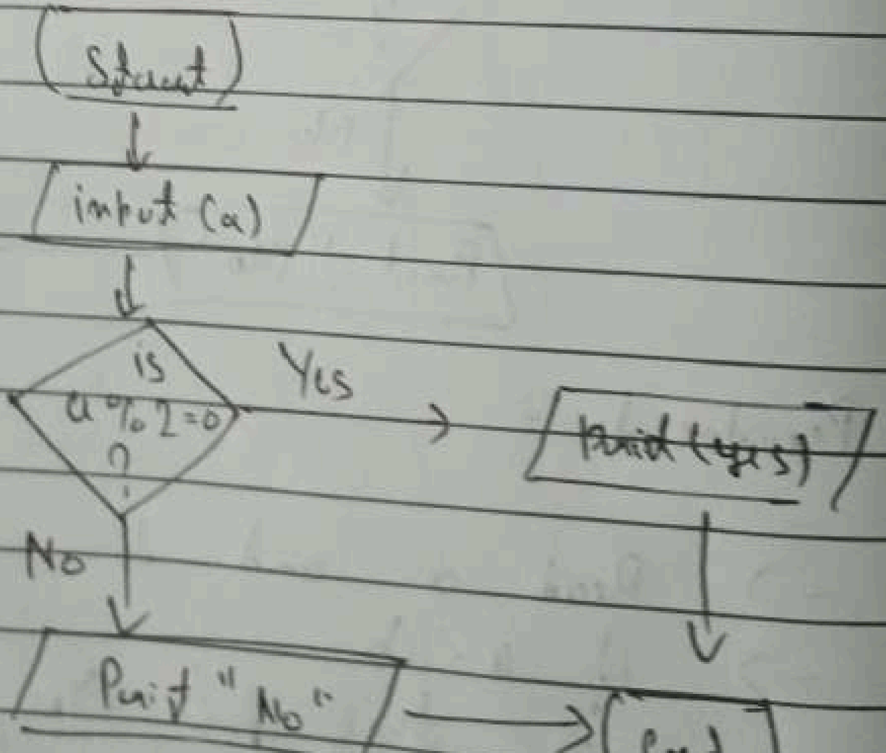
New Concept ÷ % (modulo) operator
Gives the remainder after division

a/b.
∴ a % b = Remainder of a/b

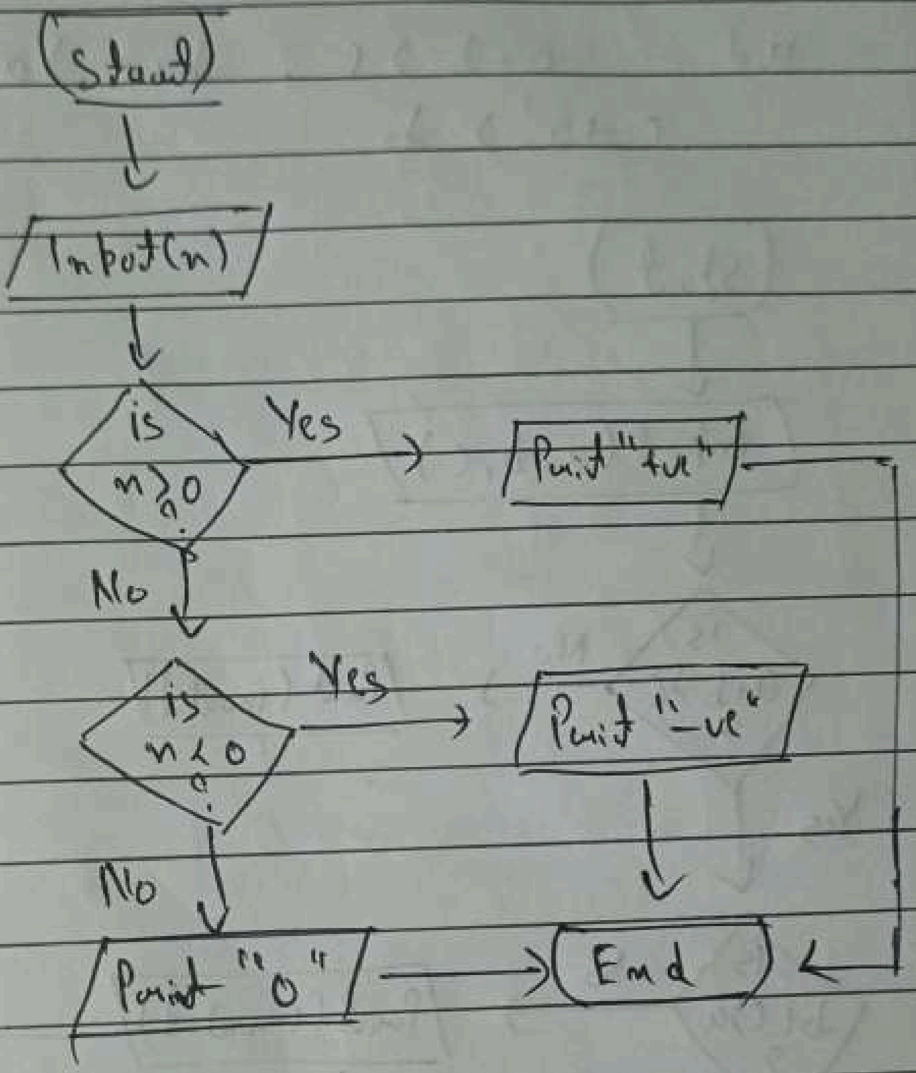
Eg: 5 % 2 = 1
6 % 4 = 2
8 % 4 = 0
* 4 % 9 = 4 when a > b,
a % b = a

if n % 2 = 0, n is even
else n is odd.

Example ÷ Check if n is even or odd

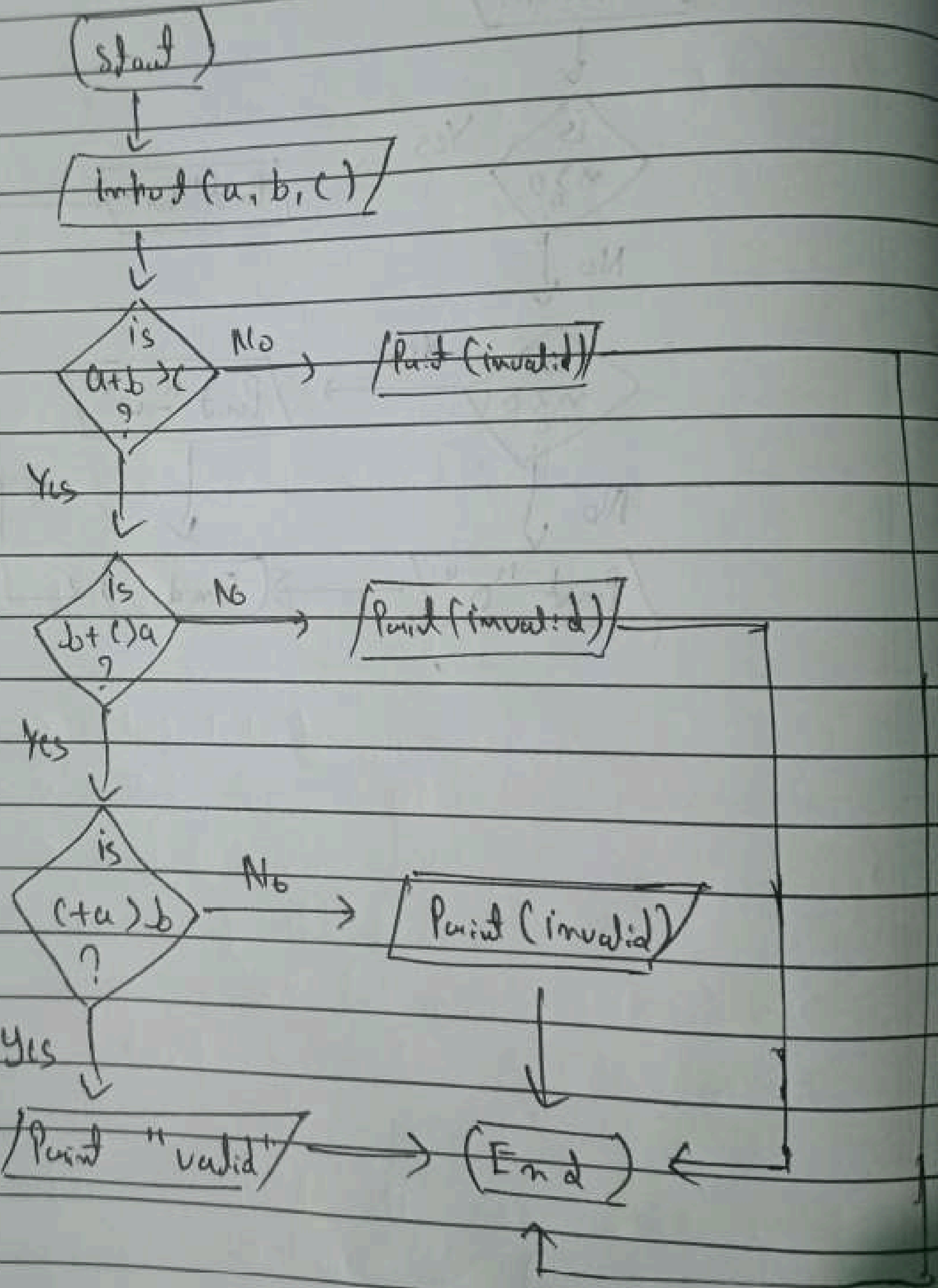


Example ÷ Is n positive, negative or zero.



Homework : Check if a given triangle is valid.

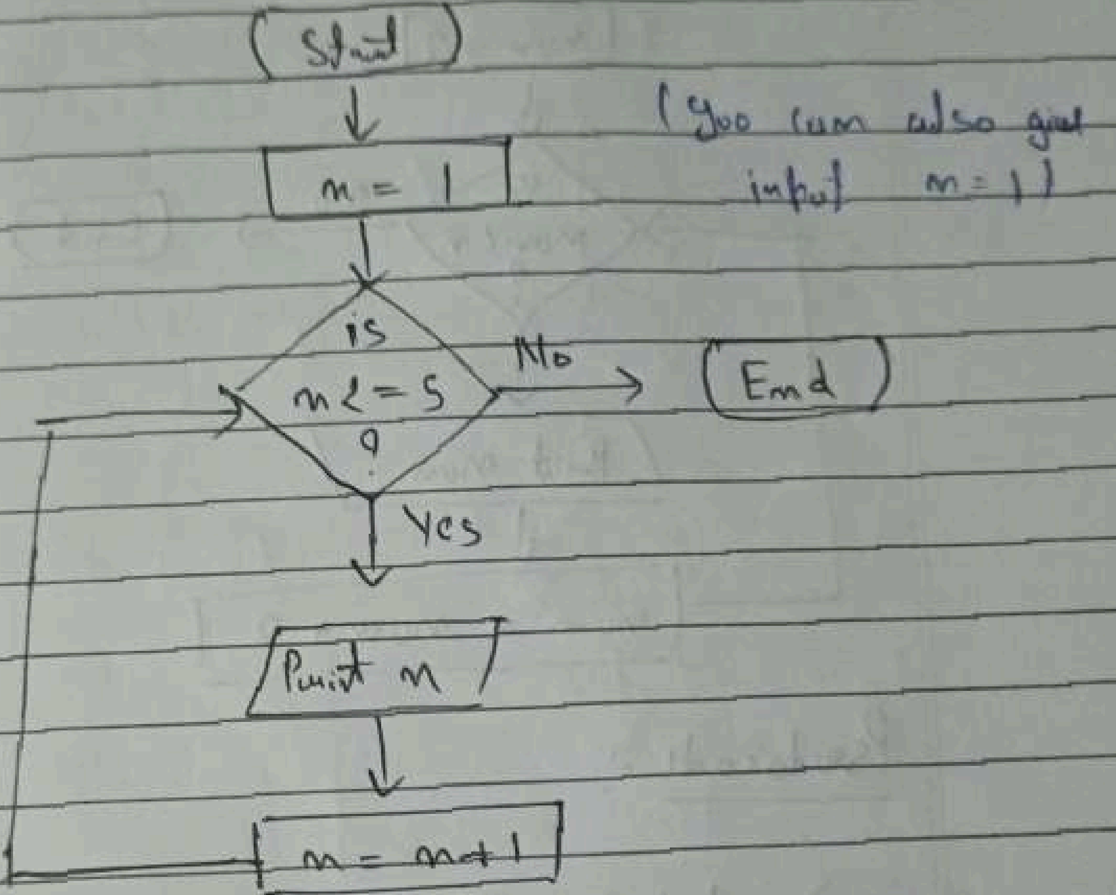
Hint - a+b > c , b+c > a &
c+a > b



Loops

Let variable n = 1
now make n go from 1 to 5.

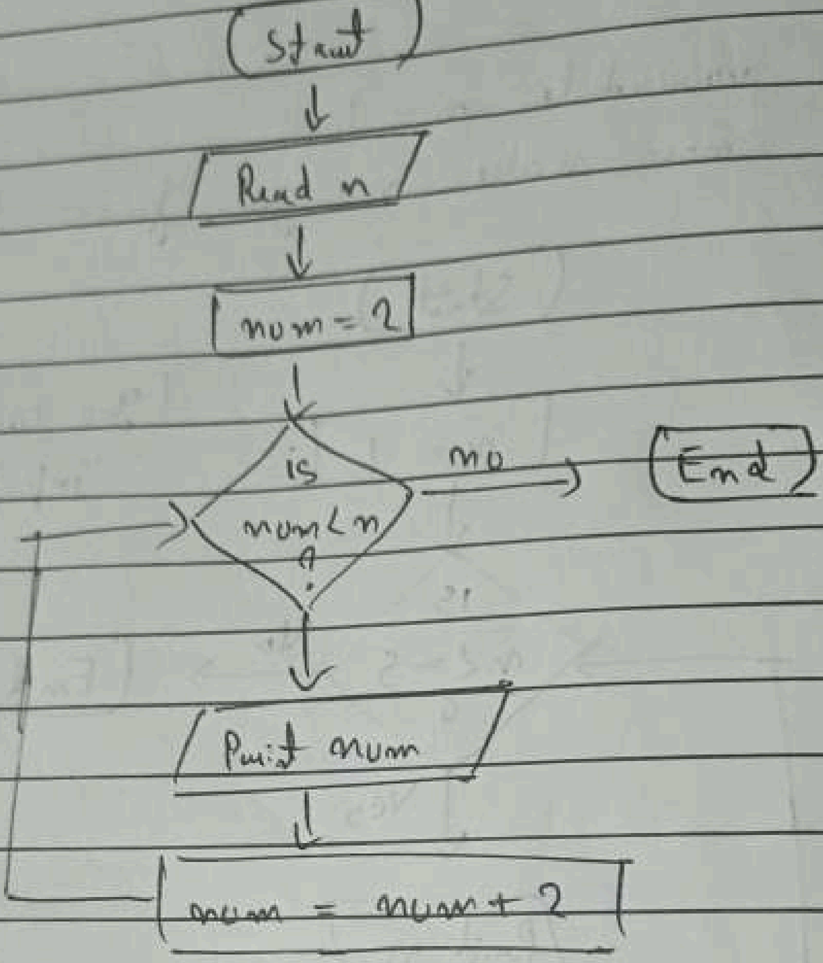
Soln.



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

Example ÷ Print even numbers b/w 1 and n. (enclusive)

Hint ÷ Even numbers start from 2 and occur alternatively.



Pseudocode ÷

```
-> Input (n)
-> let num = 2
-> while num < n,
       print num
       num = num + 2
-> End
```

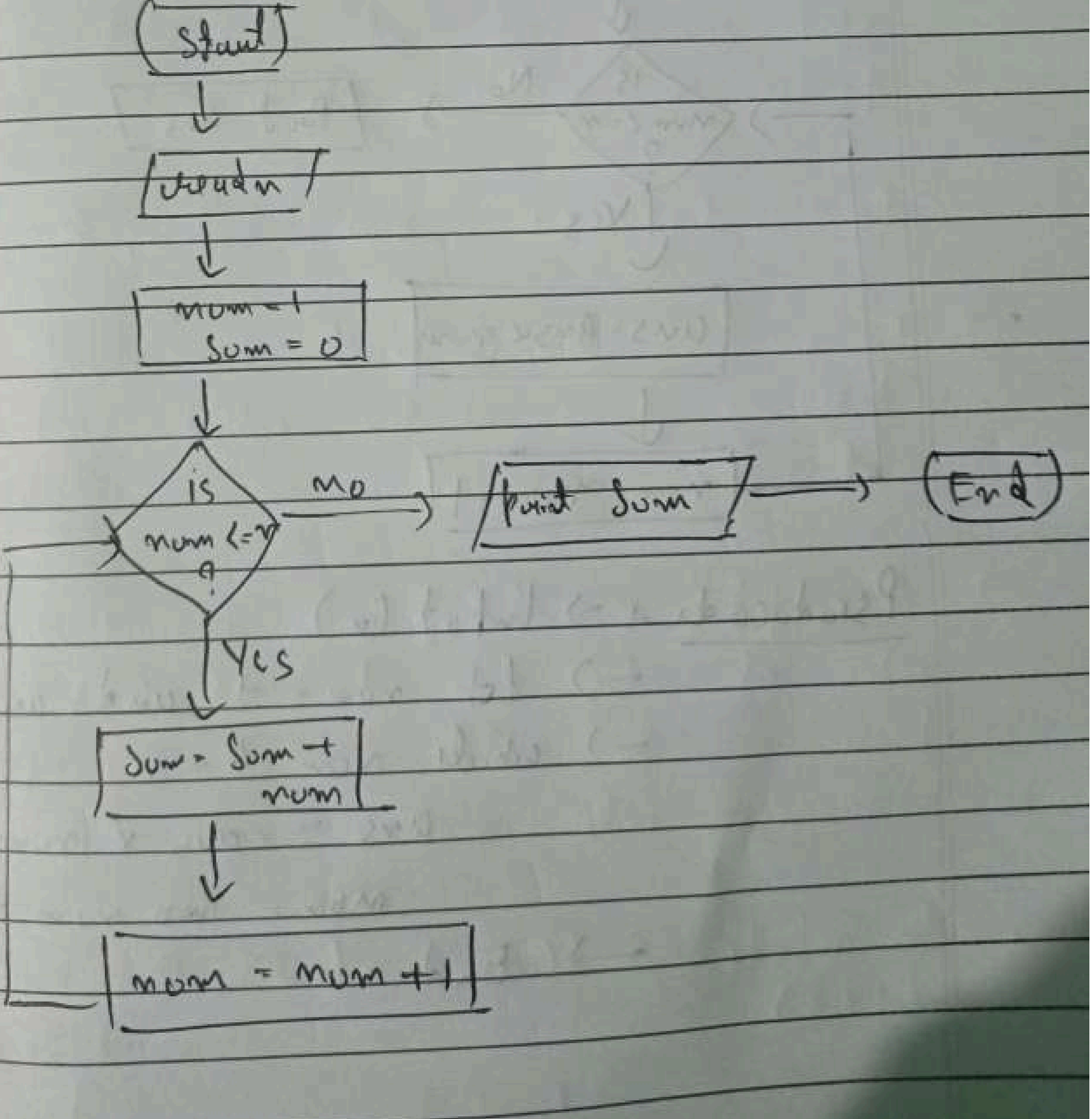
Example ÷ Print all odd numbers from 1 to n (inclusive)

(Homework)

Pseudocode :

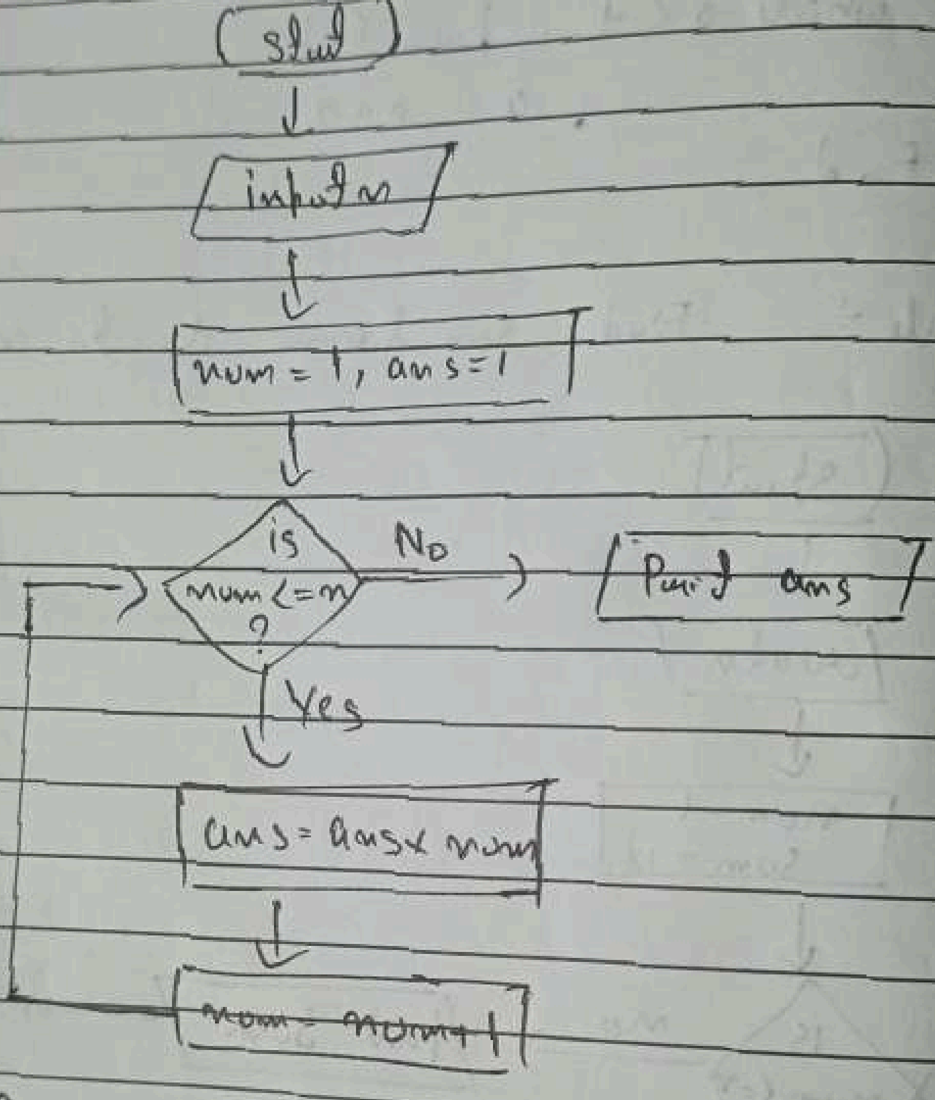
```
-> input (n)
-> let a = 1
-> while a < 1 , print (a)
                 a = a+2
-> End
```

Example : Find sum from 1 to n.



Homework :- find n!

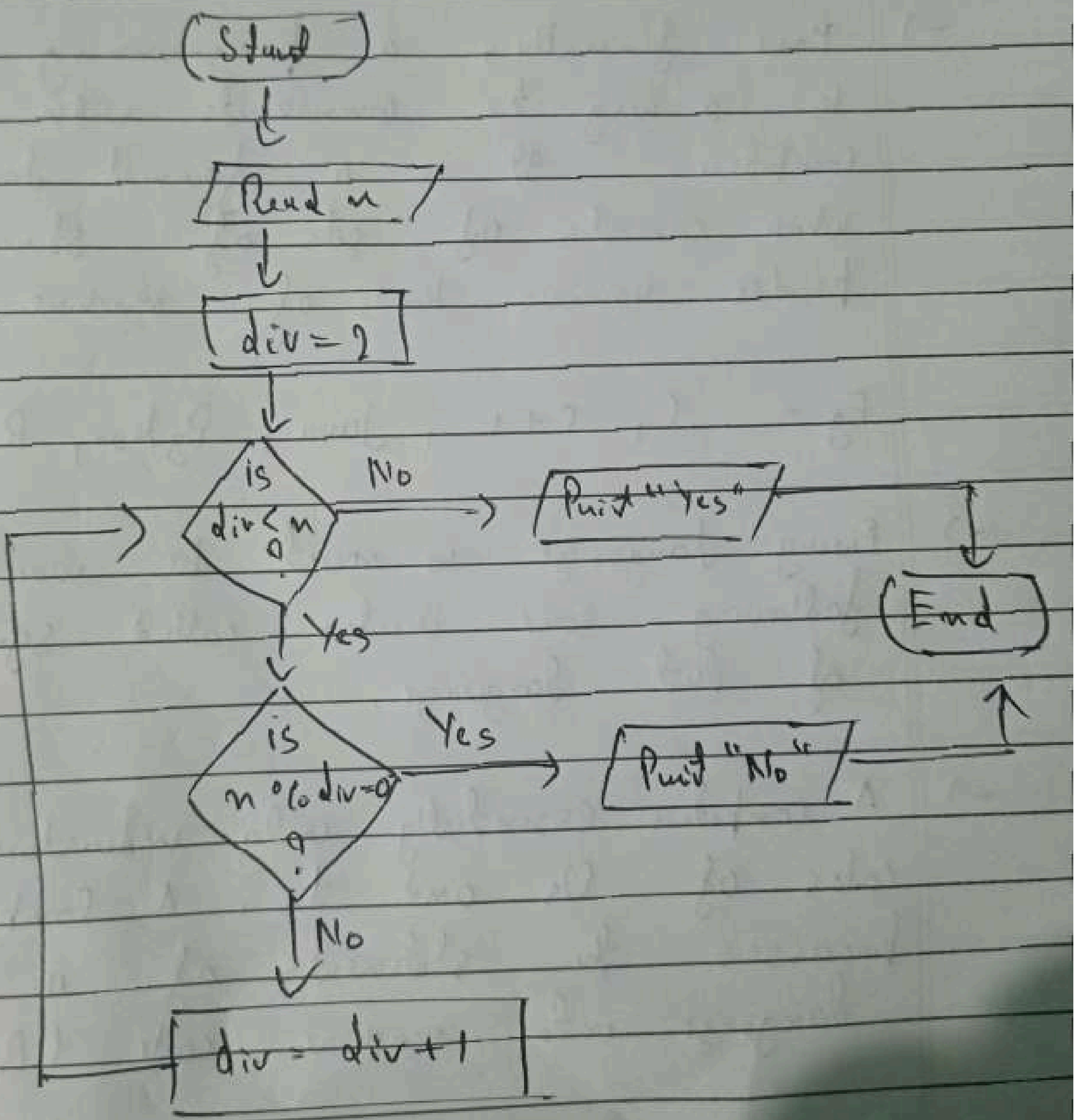
Hint - $n! = n \times (n-1) \times (n-2) \times \ldots \times 3 \times 2 \times 1$



```
Pseudocode :- -> Input (n)
              -> let num = 1 and ans = 1
              -> while num <= n
                     ans = ans x num
                     num = ~~num~~ num + 1
              -> End
```

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



Programming Language

What is a Programming Language?

-> Jaise khana khane ke lie mommy Papa ko bolna Padta 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, C#, ...

-> Every language is 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 (C++) -> [Compiler] -> Binary / Machine code

Computer Executes (.exe) <-