

DSA (Data Structure and Algorithm)

Intro to Programming and Flowcharts!

Topics covered:

→ Course flow

→ Flowcharts

→ 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 type?

③ Figure out an approach → $a+b = \text{my answer}$

④ This comes from practise and past coding

experience

⑤ Code! → `int ans = a+b; cout << ans << endl;`

Given some problem \boxed{P} . Say you 'think' of some solution solution, Je ebabe ebabe konte hobe, etc. Now write down this solution on paper, not necessarily in correct syntax (code in language) now your idea is on paper. convert this rough work, also called ^{Welt} ^{etwa Welt} into a program in a programming language of your choice say C++.

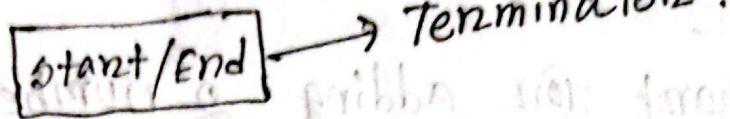
[Note taken on ~~one page~~ one page].

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

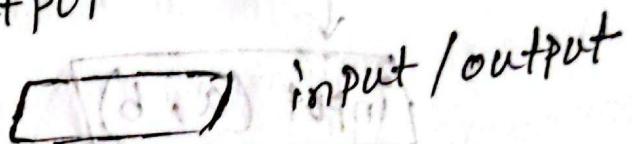
A Flowchart is a diagrammatic representation of an approach. This draws out all the steps of your approach in order.

Components:

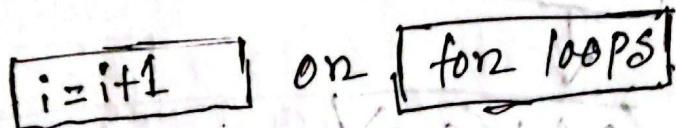
④ Terminator: specifies the start and end of a program.



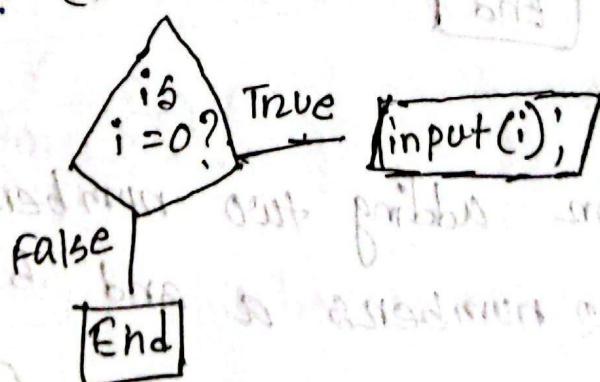
⑤ Parallelogram: for taking input or showing output



⑥ Process: operations and process on Jorro.



⑦ Decision Making: (Diamond Shape)

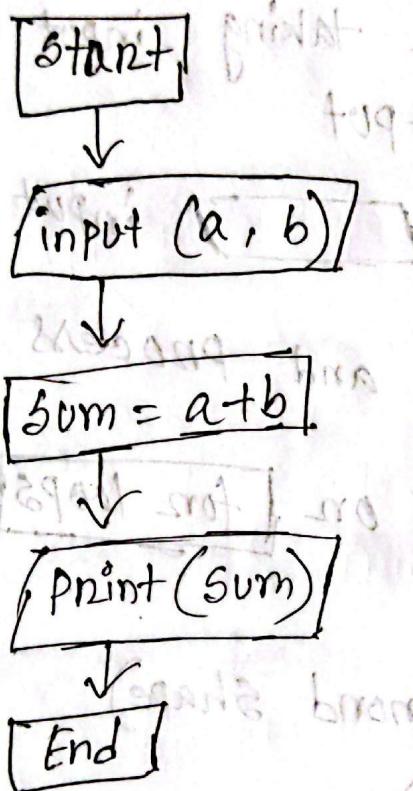


⑧ Circle: connectors (to be covered when we discuss function/methods)

⑥ Arrows: Code en provaahlo Dekanon Tonno.
(upone flowchart dekon)

Example!

④ Flowchart for adding 2 numbers



④ Pseudo code for adding two numbers!

→ input 2 numbers a and b

→ Let sum = a+b

→ print out sum

fik
etao fik!

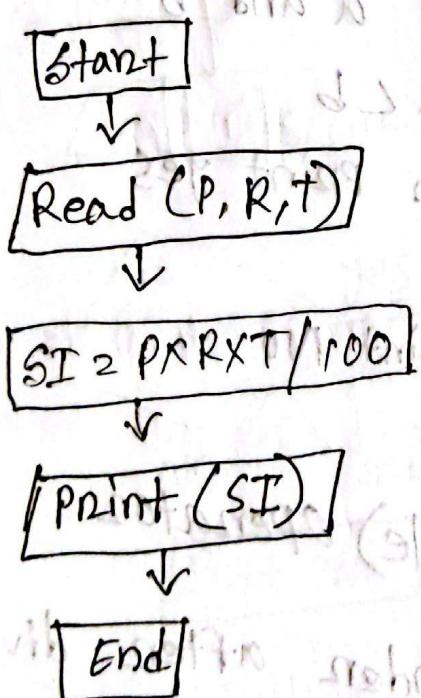
- Read a and b
- sum variable is $a+b$
- sum chaapo

⇒ Both pseudocode are ok. NO pseudocode is wrong as long as logic is same / similar.

Example: calculate simple interest

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

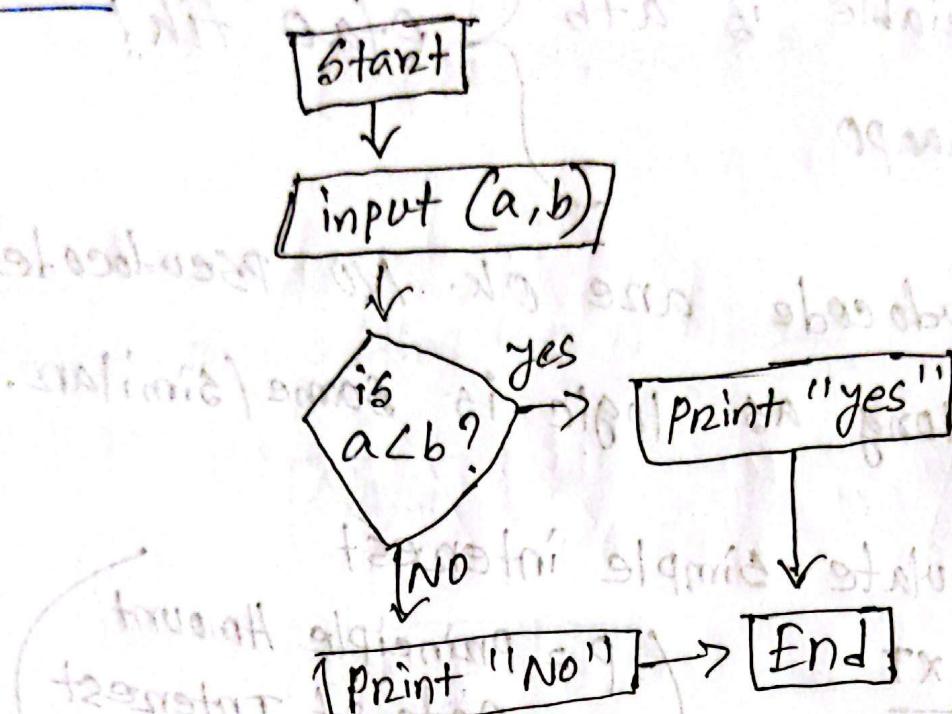
(P = Principle Amount
R = Rate of Interest
T = Time)



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! % (module) operator

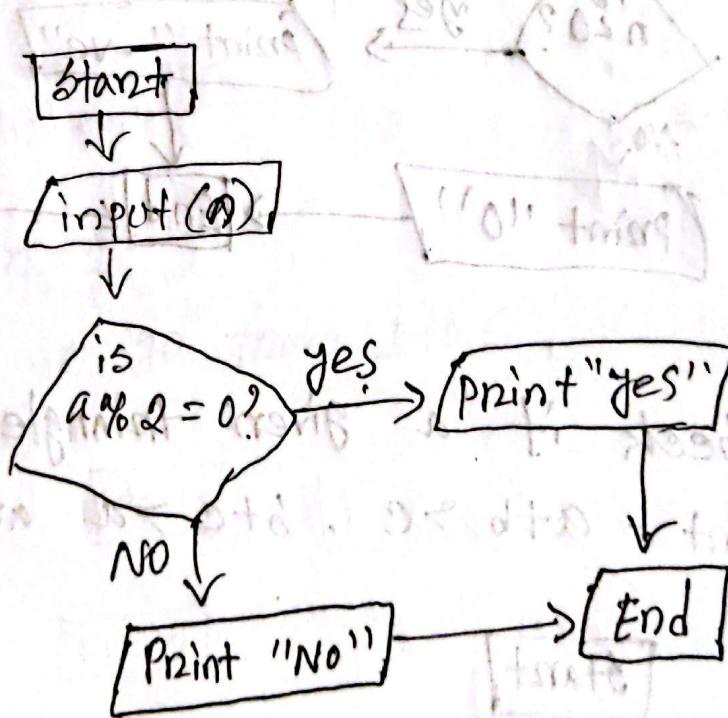
Gives the remainder after division a/b.

i.e. $a \% b = \text{Remainder of } a/b$.

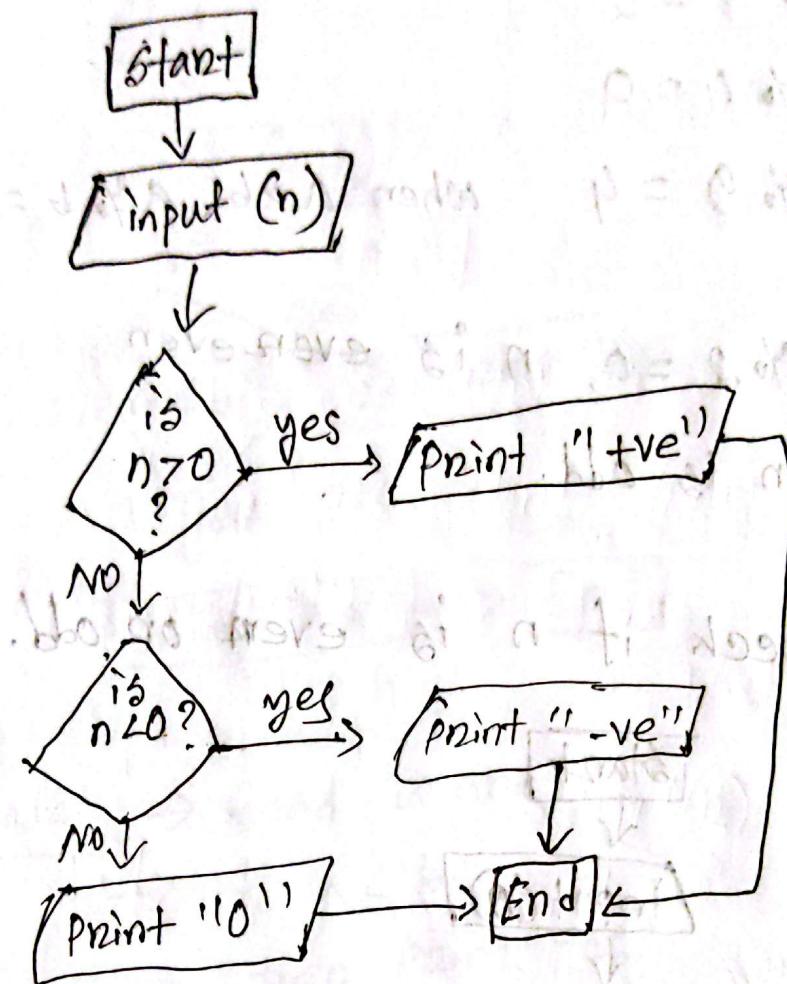
Eg! $5 \% 2 = 1$ ~~written, easiest~~ a bit hard
 $6 \% 4 = 2$
 $8 \% 4 = 0$
 $4 \% 9 = 4$ when $a > b$, $a \% b = a$

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

Example! Check if n is even or odd.

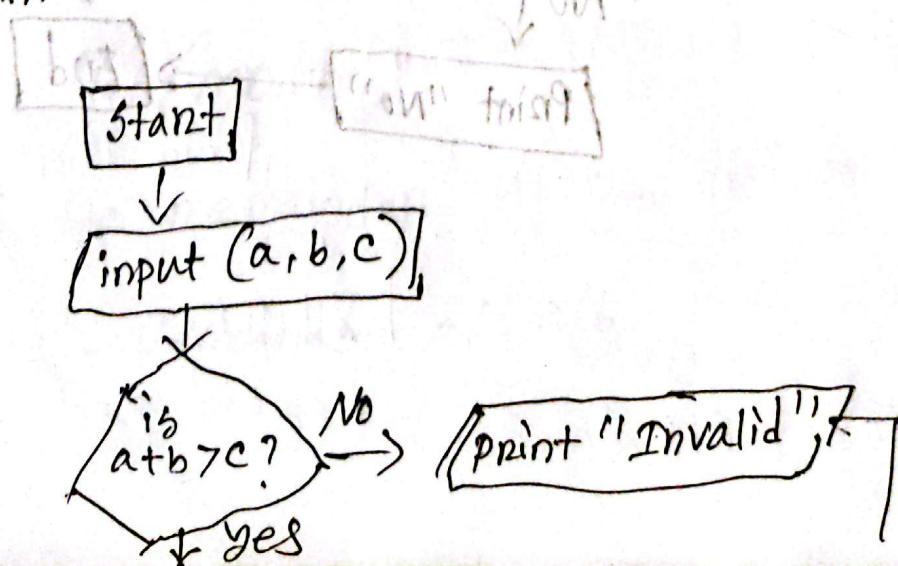


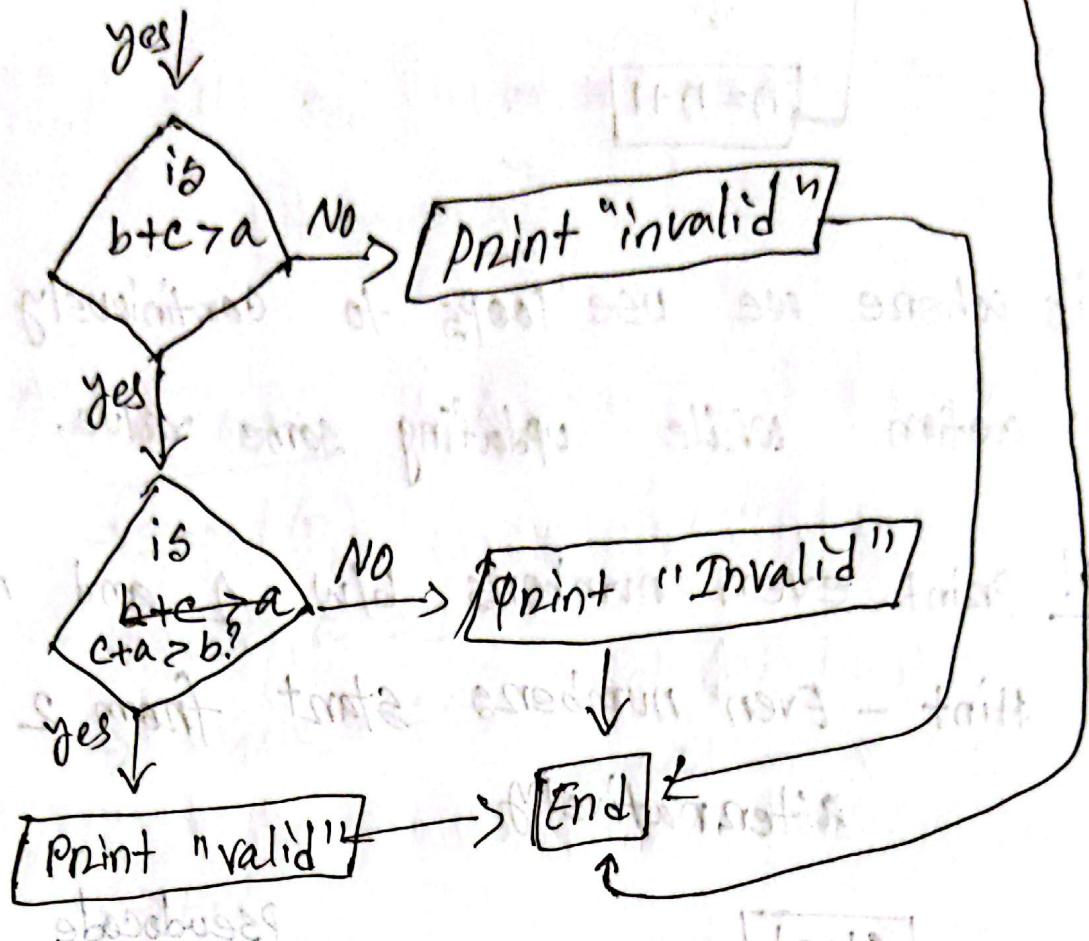
Example! Is n positive, negative or zero



Homework!

check if a given triangle is valid
Hint - $a+b > c$, $b+c > a$ and $c+a > b$



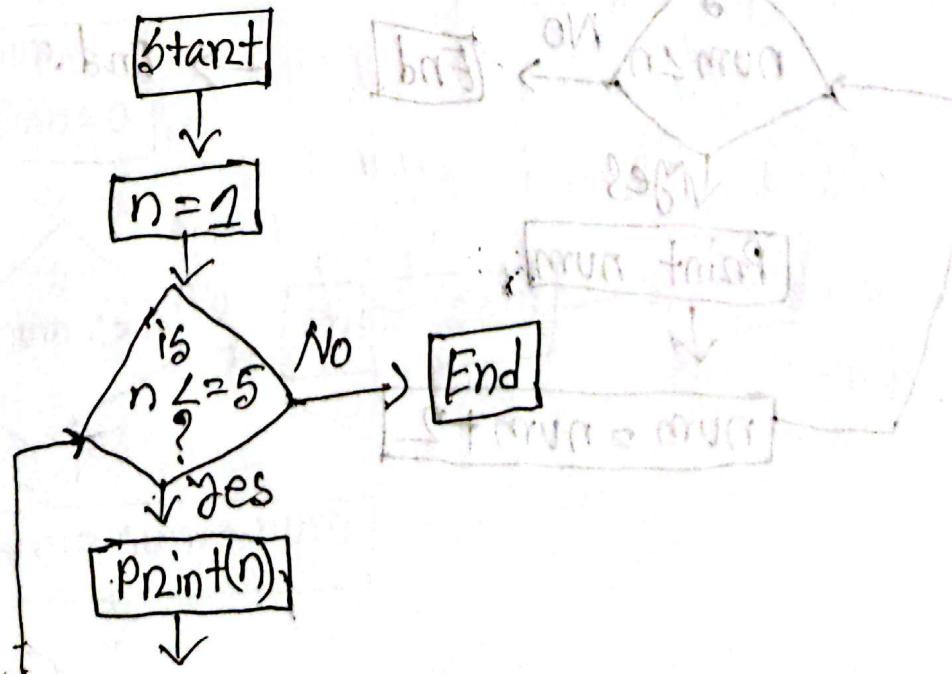


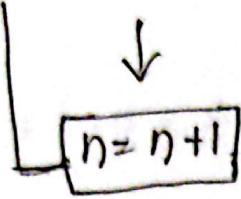
Loops

Let variable $n = 1$

Now make n go from 1 to 5

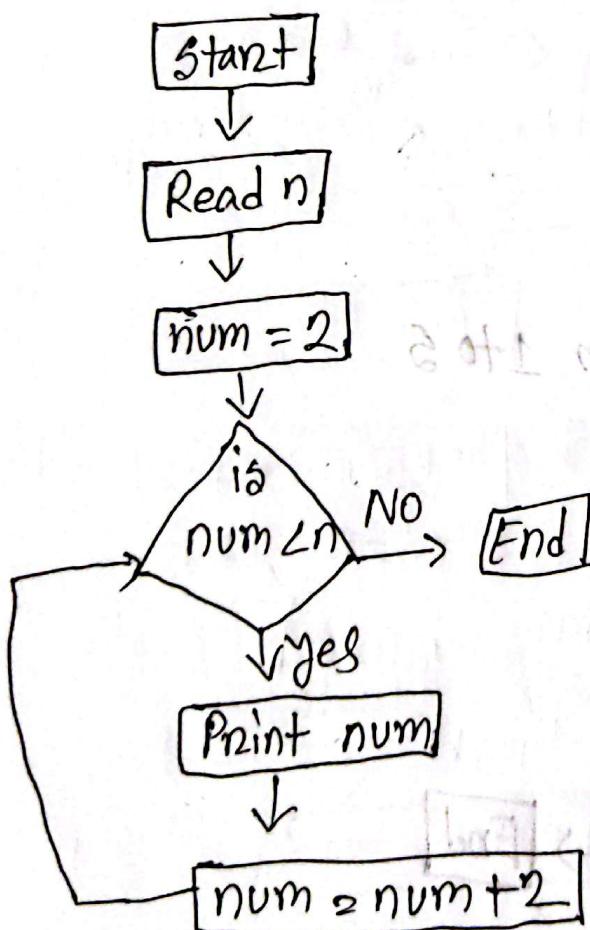
solution!





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



Pseudocode

```

→ Input(n)
→ let num = 2
→ while num < n, do
    → print num
    → num = num + 2
→ End.
  
```

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

[Homework]

pseudocode

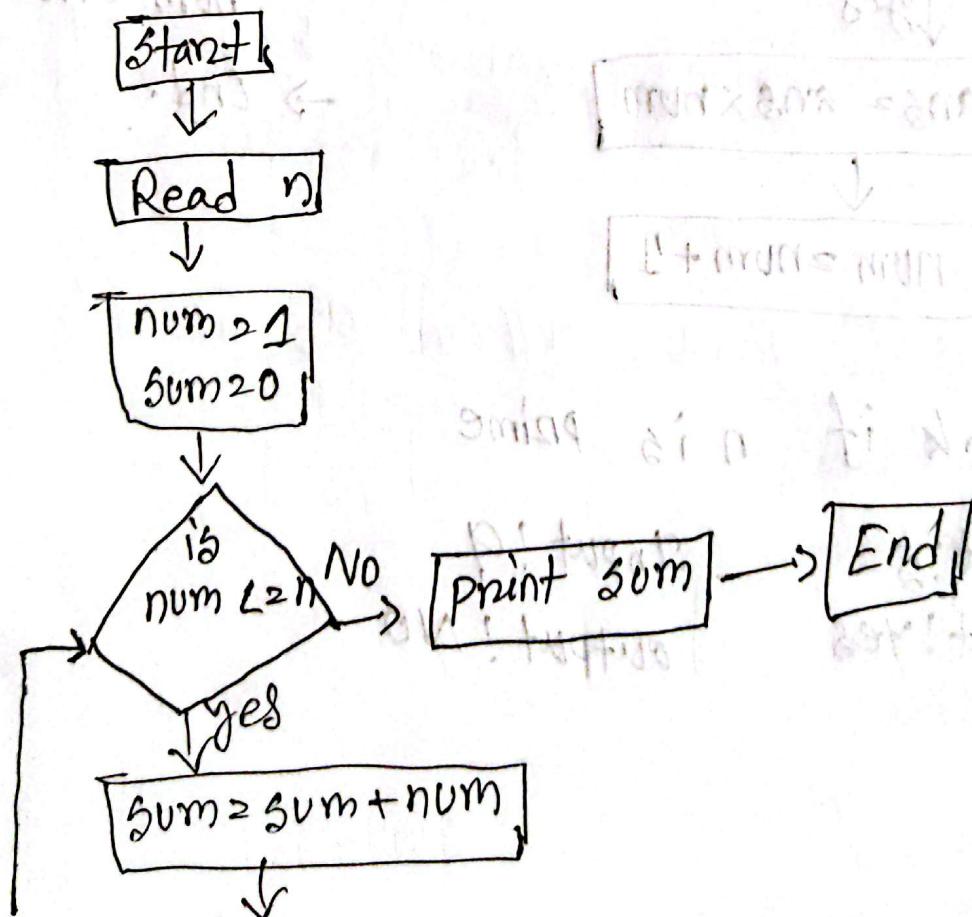
→ Input(n)

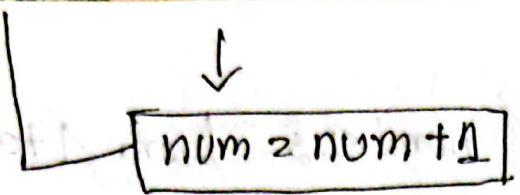
→ Let a = 1

→ while a <= n, print(a)

→ a = a + 2
→ End.

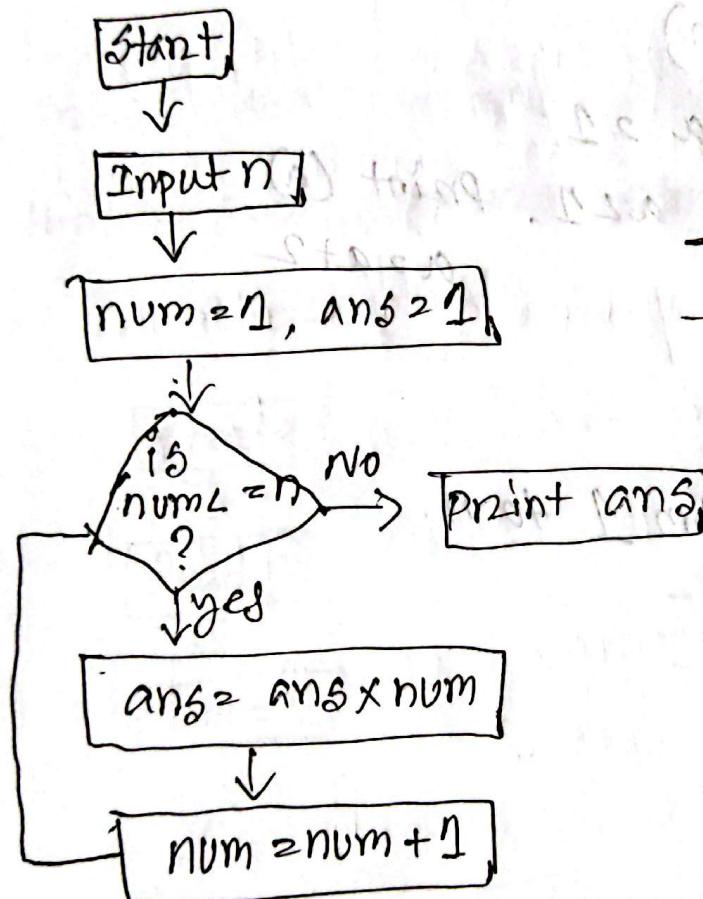
Example! Find sum from 1 to n.





Homeworks! find $n!$

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



Pseudocode

```

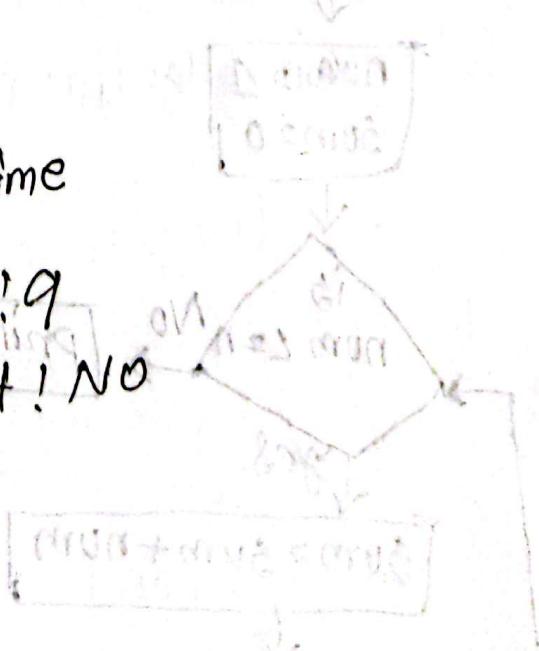
→ Input(n)
→ Let num = 1 and
  ans = 1
→ while num <= n
  ans = ans * num
  num = num + 1
→ End.

```

Example! check if n is prime

Input: 5
output: yes

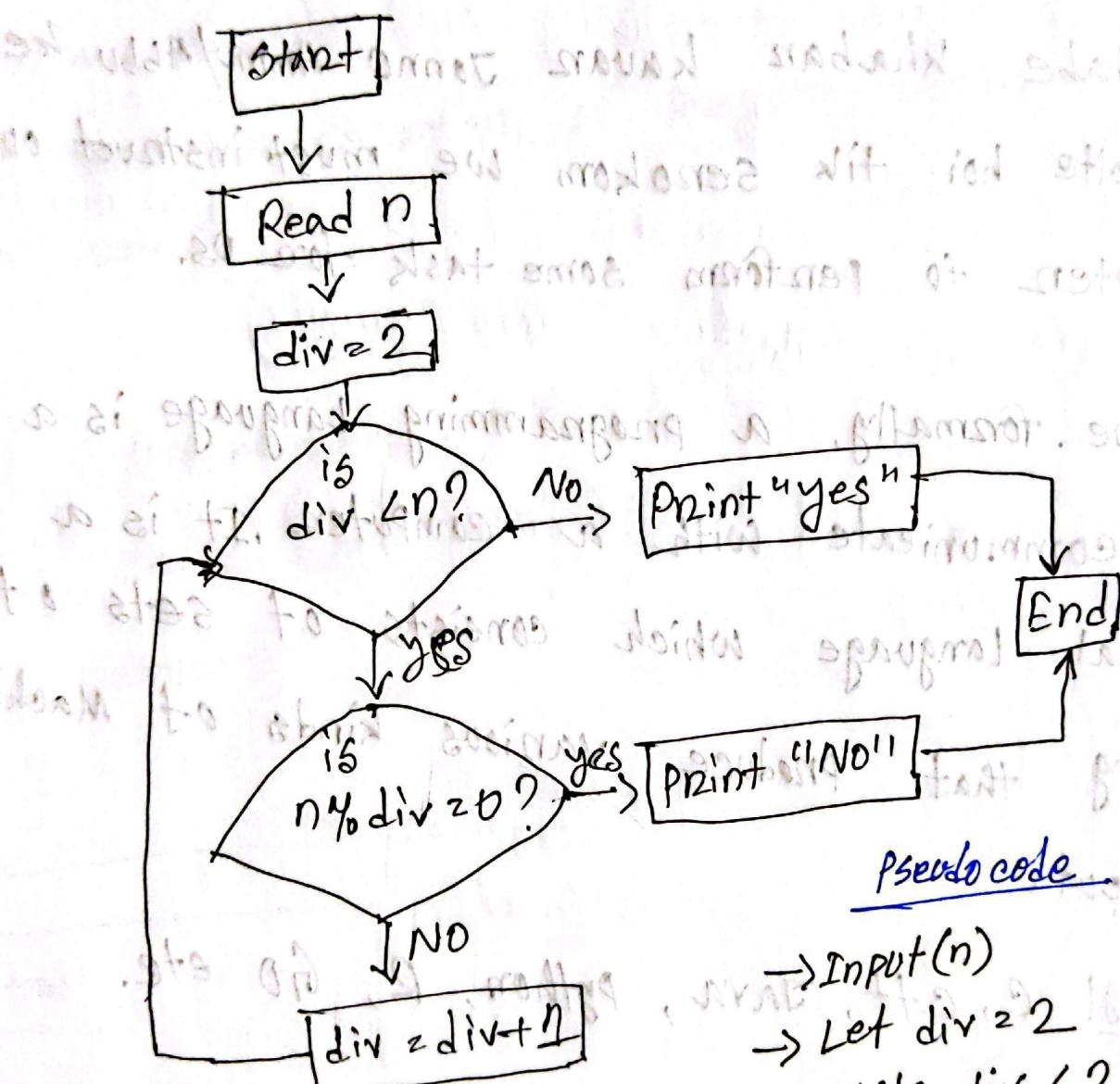
Input: 9
output: NO



Prime n: $n \% \text{ (any number from 2 to } n-1)$ $\neq 0$

written as \leftarrow

$! = \text{in C++}$



Pseudo code

```
→ Input(n)  
→ Let div = 2  
→ while div < 2  
  if n % div == 0  
    print "NO"  
    exit  
  else  
    div = div + 1  
→ Print "yes"
```