29-11-2021 DSA - Lovebabbar C++

Day 1:-

Problem Solving

P → understand
→ given
→ approach
→ Program

P → sol^n

Flowchart / Pseudocode → Program (source code)

Rough sol^n

High level language

converting

Machine understandable

→ computer understands 0's & 1's.

Ex:-
Program:- taking Ilps a & b
Printing it's sum

=▷ Flowchart:-

Diagramatic representation of approach.

① components:-

① start/end → terminator, tells start & end

② Read a → used for Ilp & Olp. Print "Hello"

③ i = i + 1 → Process [Ex:- calculation]

④ Print i!=0 — no — is i=0? — yes — end

→ Decision making [if i=0]

↓

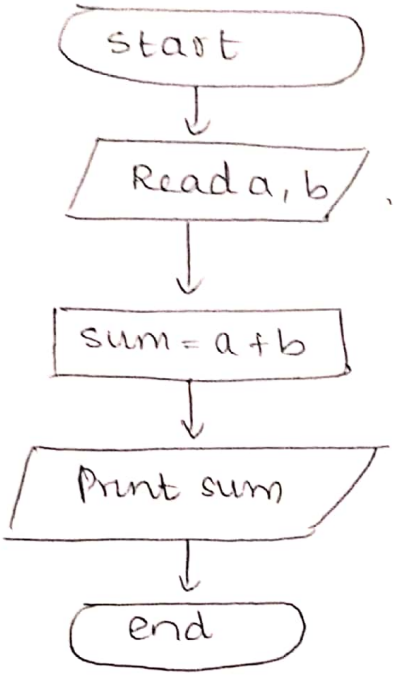
has only 2 Olps

⑤ ↓ ↑ → arrows [flow ko batane]

⑥ (A) → connection [used in f^ns]

① Flowchart of sum of 2 no's:-

→ Given a, b where a = 5 and b = 10 & sum = 15



memory k andar 2 blocks bangaye

5	10.
a	b

sum = a + b

Sum = 15

| 15 |
|---|
| sum |

Pseudocode:-
(nakli)

→ It is way of representing a logic → which is general.

→ remains same for every language.

Pseudocode for sum of 2 no's:-

→ Read 2 no's, a & b
→ Sum = a + b
→ Print sum

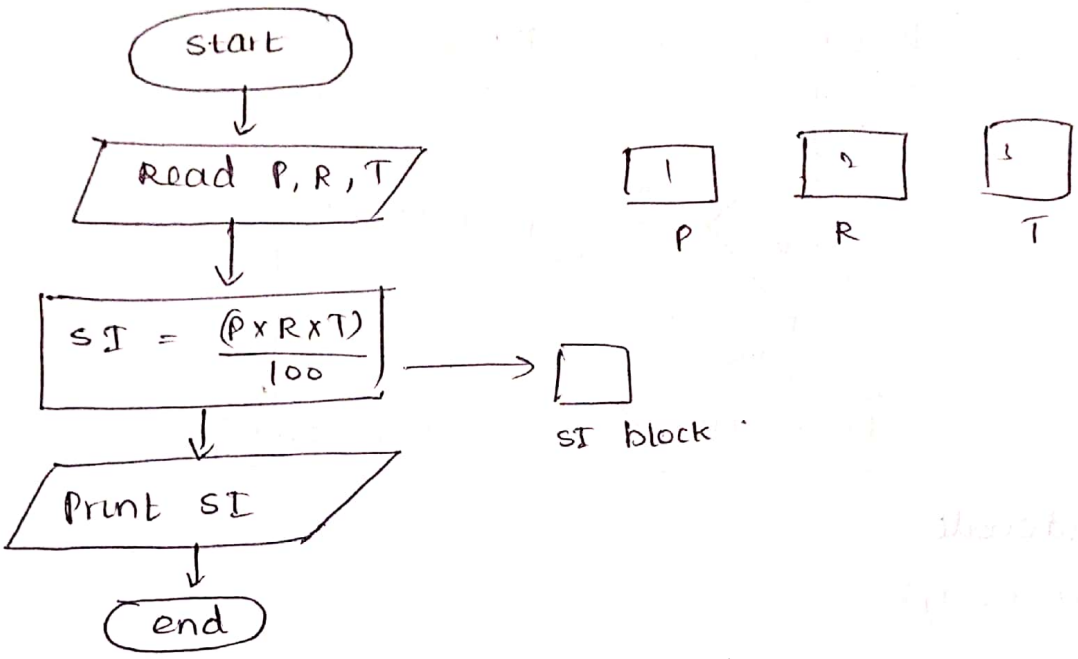
→ read a
→ read b
→ let sum = 0
→ sum = a + b
→ print sum

→ Pseudocode is right when it leads to same code

Sadiyall

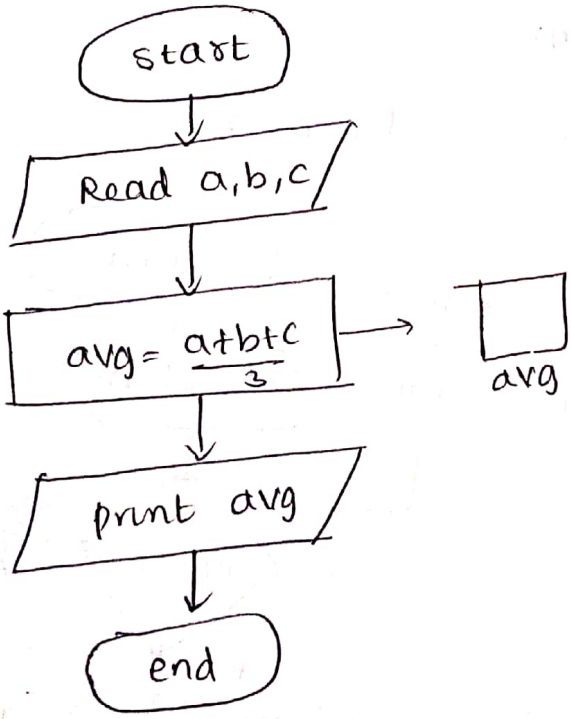
② Flowchart for calculating simple interest:-

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



③ Flowchart for average of 3 no's:-

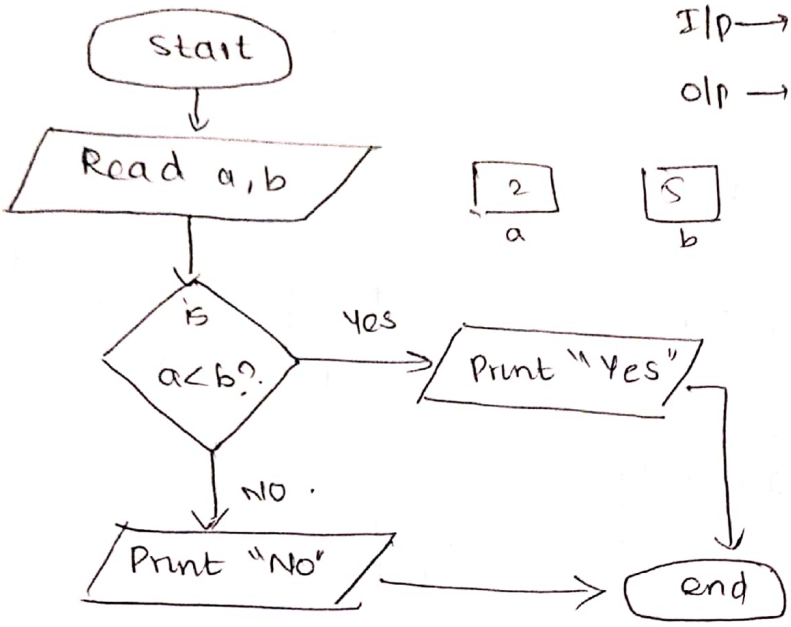
$$avg = \frac{(a+b+c)}{3}$$



④ Flowchart to know If a<b → Print 'Yes' or 'No':-

I/p → a, b

o/p → 'Yes' or 'No'



Pseudocode:-

→ Read a, b

→ If a < b

Print Yes.

→ else

Print No

⑤ Flowchart for odd or Even:-

% operator → gives rem

Ex: 5 % 3 → 2

6 % 2 → 0

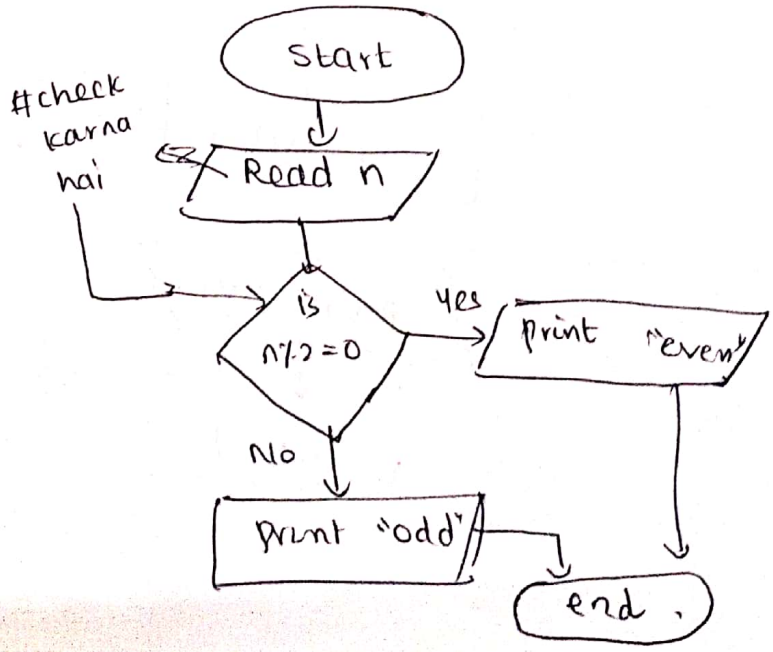
n % 2 = 0 → even

n % 2 = 1 → odd

Ex:→ 6 % 2 = 0

→ 7 % 2 = 1

Ex: n = 3



Pseudocode:-

→ Read a,s n

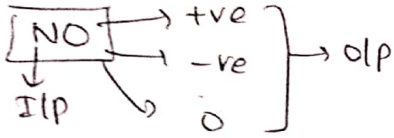
→ if n%.2 ==0 :

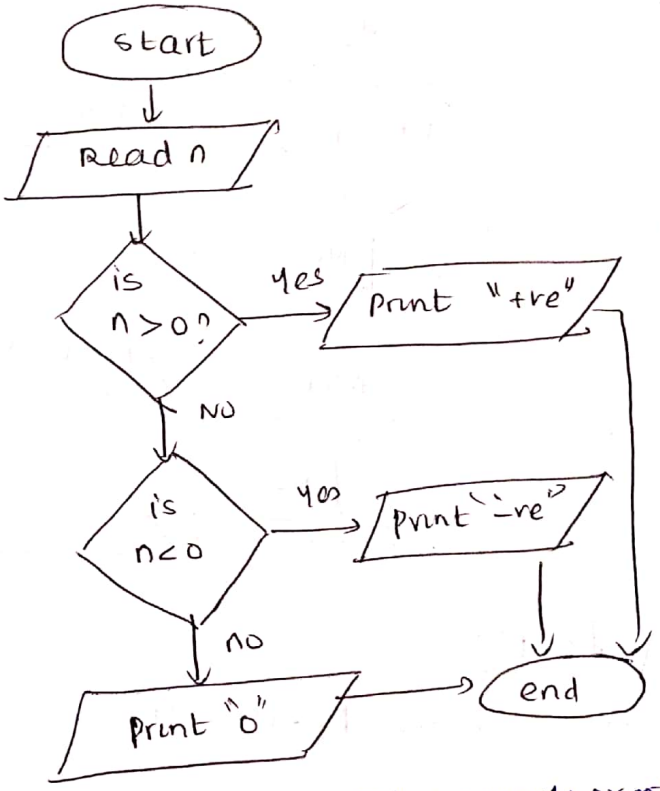
print even

→ else

print odd

⑥ Flowchart for determining whether given number is +ve, -ve or 0 :-



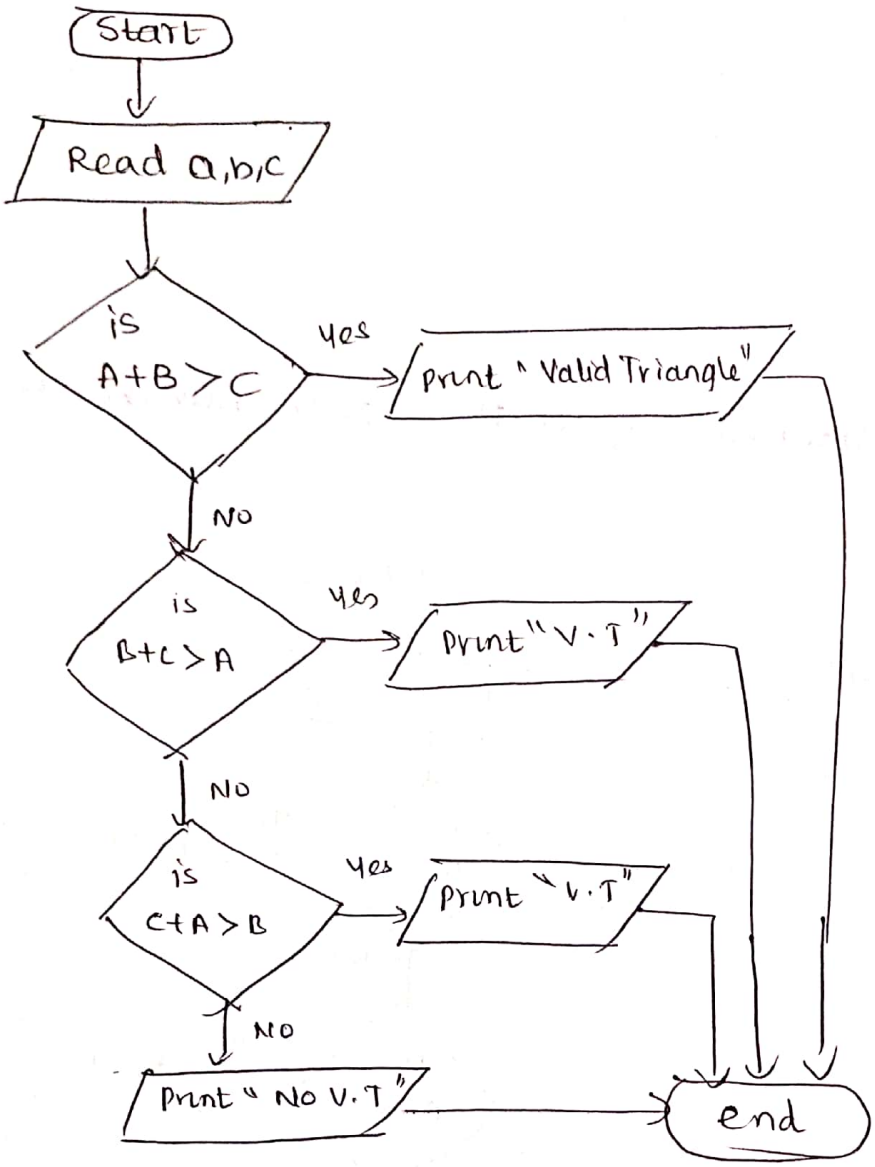


⑦ Flowchart for determining given triangle is valid triangle or not

Hw A + B > C

B + C > A

C + A > B



Loops:-

(8)

→ Given no, N → as I/p

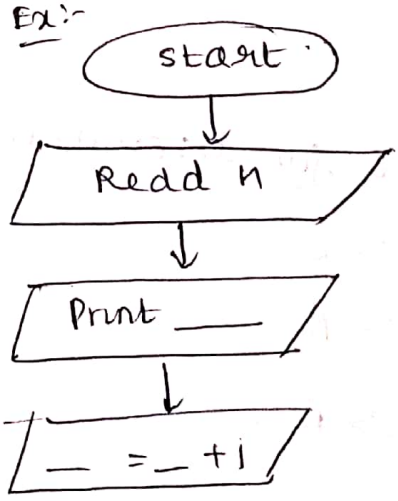
→ asked to print 1 to N

ex:- I/p → n = 5

o/p → 1, 2, 3, 4, 5

basic

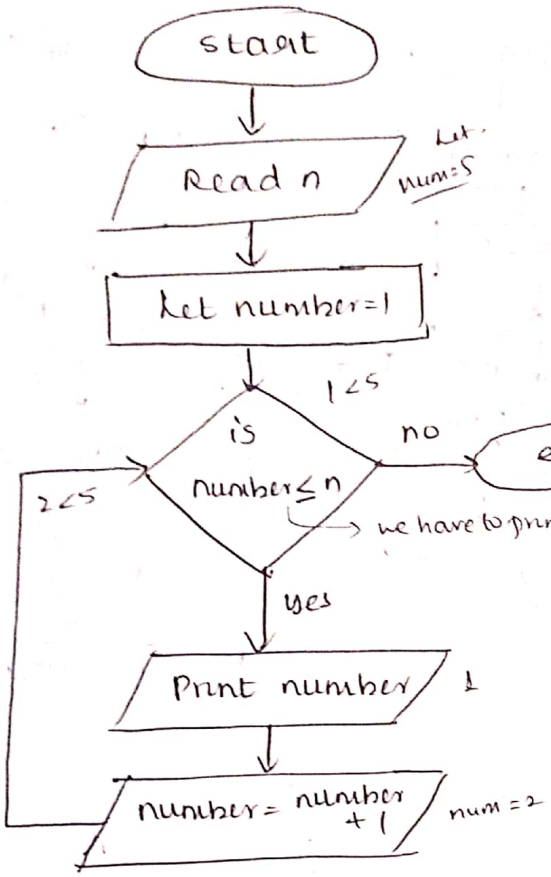
Ex:-



O/P:- 1, 2, 3, 4, 5

Ek number = 1 , lekhe.

5 ku stop karna hai



⑨ Print all even numbers from 1 to N :-

1 to n → even no's print

Let's say : I/p → n → 5

o/p → 1, 2, 3, 4, 5

Print 2, 4

Logic

number = 2

2 < n.

2 ⟶ ✓

2 + 2 = 4

4 < n ✓

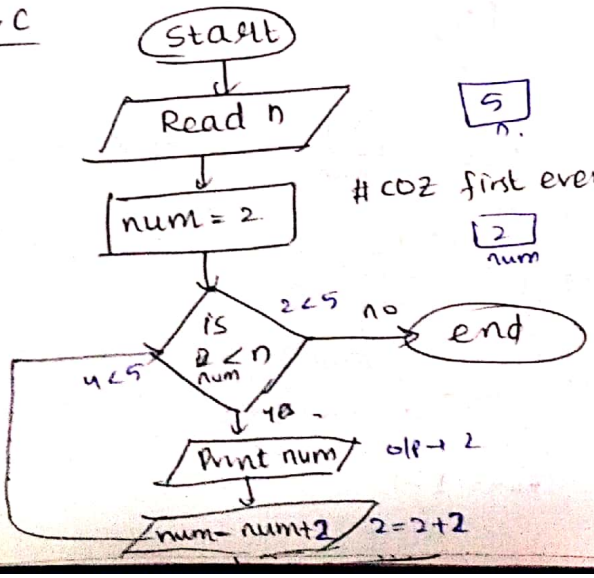
4 + 2 = 6

6 < n ✓

6 + 2 = 8

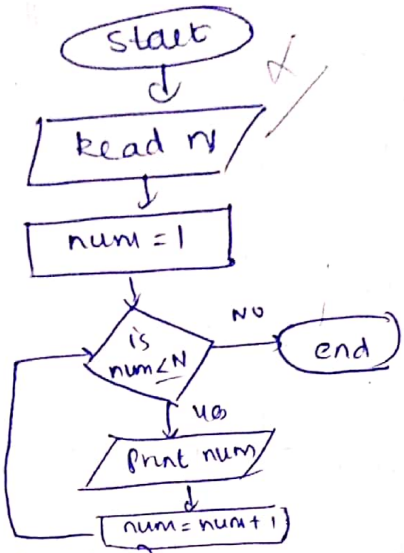
8 < n ✓

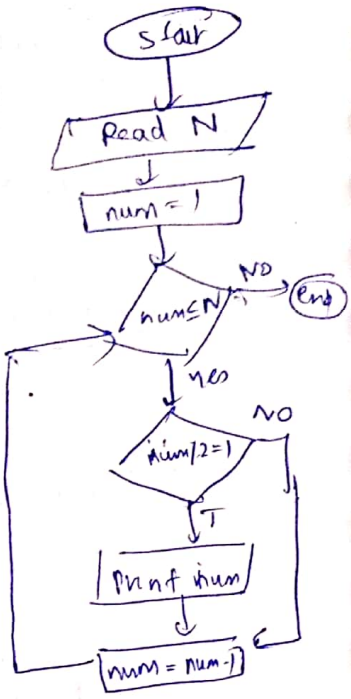
F.C



H/w

F.C:- print all odd numbers from 1 to N where (1,N) are inclusive.





(10) Find sum 1 to N (inclusive).

Let, n = 5

O/P → sum → 1+2+3+4+5 = 15

num = 1 ↳ sukhna hai → n. tak

⇒ n = 5

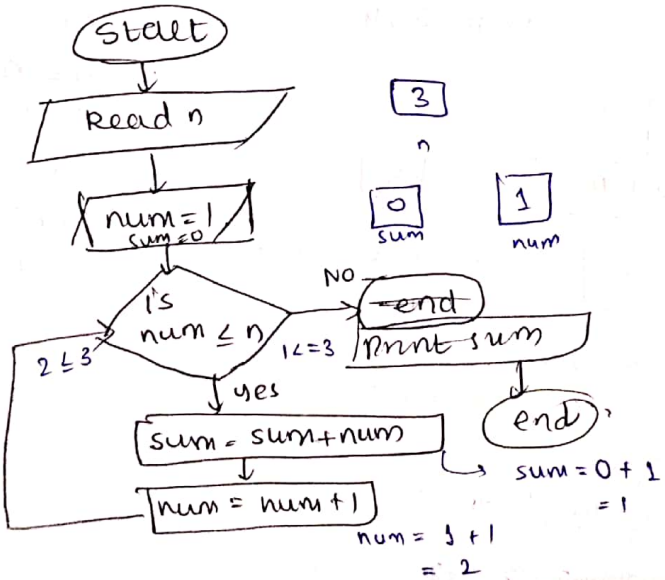
number 1 | sum = 0+1 = 1 | sum = 1+2

sum = 0 | number = 2 | number = 3.

⇒ n = 5

sum = 0, 1, 3, 6, 10, → 15

number = 1, 2, 3, 4, 5 → 6 > 5 ✗ → #Day run



5 < 5 ✗

5 <= 5 ✓

(11) Homework:-

Find factorial of n

I/p n=5

$5! = 5 \times 4 \times 3 \times 2 \times 1$

or

$= 1 \times 2 \times 3 \times 4 \times 5$

o/p → 120

(12) check prime or not

I/p → n

o/p → prime or not

Ex:- I/p → 7

Prime or not

Prime → 2, 3, 5, 7, 11, 13, 17.

⇒ n → Prime or Not

1 ---- N

shouldn't be in any table

Rem = 0 ✗

For n to be a prime number:-

2 ---- (n-1) → %. ! = 0

ex:- n = 5

2, 3, 4

⇒ 5%2 = 1

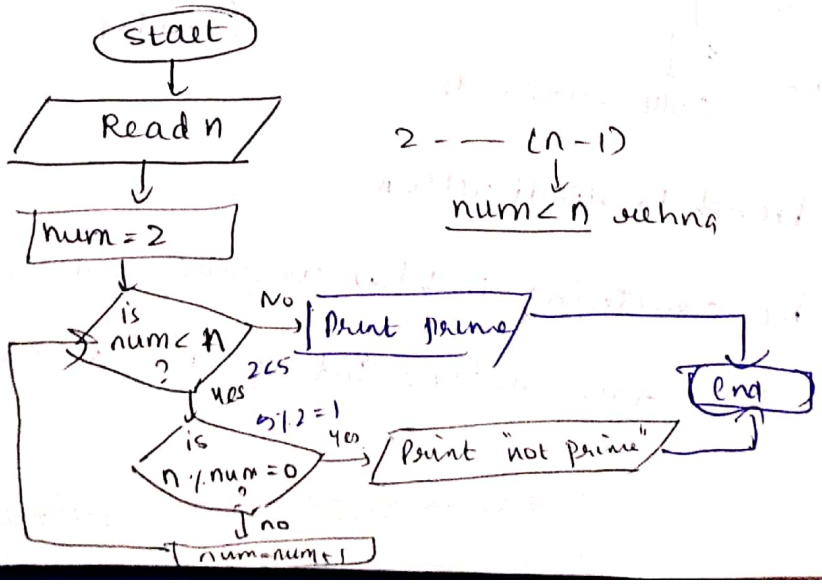
5%3 = 2

5%4 = 1

→ 5 is prime as rem ≠ 0

→ generalized logic:-

n → (2 ---- (n-1)) → divide → rem → shouldn't be zero

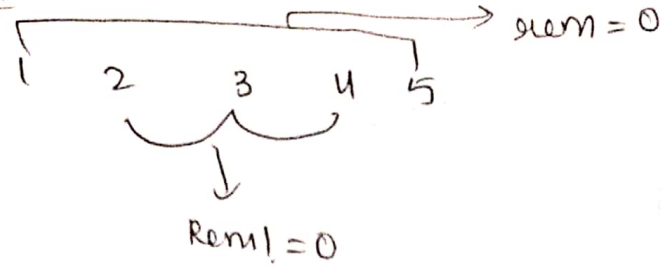


2 -- (n-1)

num < n using

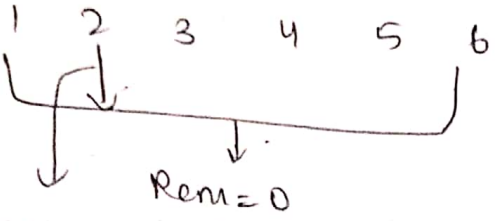
For n=5

Ex:-



Hence 5 is prime

Ex:-



but 2 kelive bhi Rem=0

so, 6 is not prime

~~HOMEWORK~~ RECAP:-

- FLOWCHART. → I/P → WHEN TO STOP
- COND^N → O/P → HOW TO INITIALIZE NUM
- LOOPING → PROCESSING → PSUEDOCODE

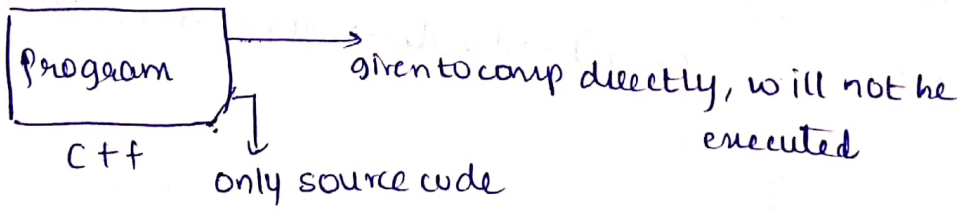
Programming Languages:-

What? computer se kaam karvana,

Why?

Every Prog L → have semantics, or rules, syntax to be followed

Lets say:-



Ex:- Talking French to Hindi person

→ There should be something in b/w. computer & program to translate i.e, translator or interpreter or magic box

→ Computer ko o/s aati hai → called as binary language bolthe

SRC Code → [Compiler] → Binary Language
↓
file called as machine executable file (format)
↳ can be given to run → computer

Learnt:- FC, Pseudocode.

H/W:- remaining FC.

Lec 2:- First program:-

ex:-
a = 1
b = 2
c = a + b;
→ If given to computer } samajnai aata.

⇒ compiler → Translation
↳ find error [RTE, CTE]

IDE → Integrated Development Environment
↳ codeBlocks, VS code etc.

→ Replit → C++ file → run. [on Google]

Flowchart:-

① → (Start) → int main() {
}

// start of program

↳ kaundar ka int main ka hai