> Basics of programming - Level 1.

Date
Page 01

Introduction to programming.

→ Algorithm : Sequence of steps to solve a problem statement.

eg: P.S chai

↳ (A) water garam
(B) chai patti
(C) ai - adrak ilaichi
(D) Doodh
(E) Sugar
:

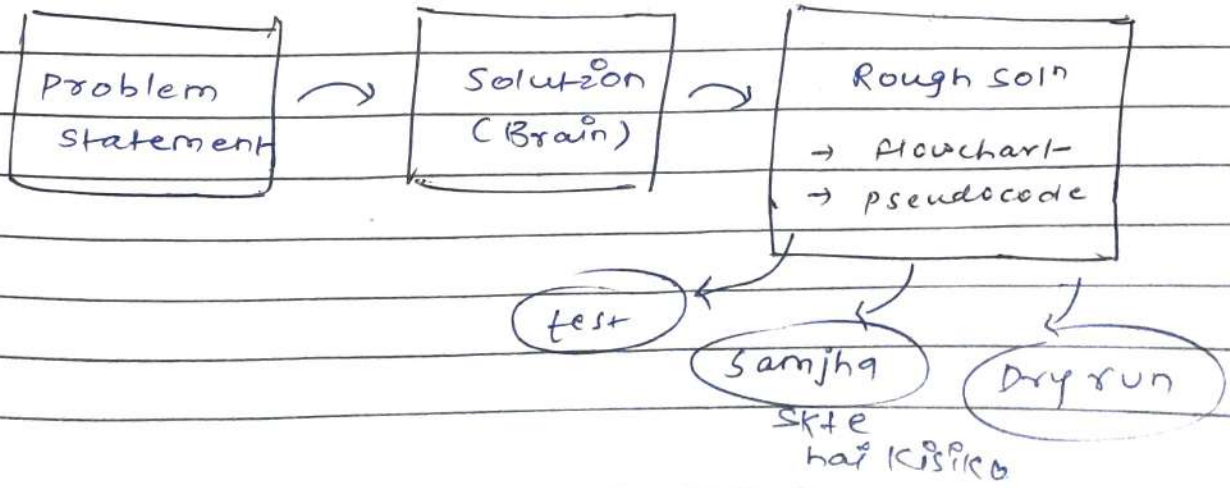
Algorithm to create chai.

→ How to Approach a problem ?
Thought process :

① Understand the problem. → eg: sum of two nos

② Input values (write down) → I/P a = 20, b = 30

③ Logic create karo / Algorithm → c = a + b



→ Pseudocode :- (Natli code)

→ Some steps of a solution of a problem statement in roughly English lines.

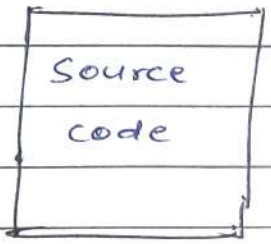
eg: add 2 nos

↳ take 2 input a & b.

↳ sum = a+b

↳ print sum.

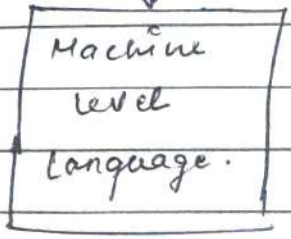
When now convert the logic into source code (because computer knows only 0s and 1s)

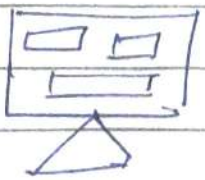


→ In High level Language
eg: C++

- User-friendly language.

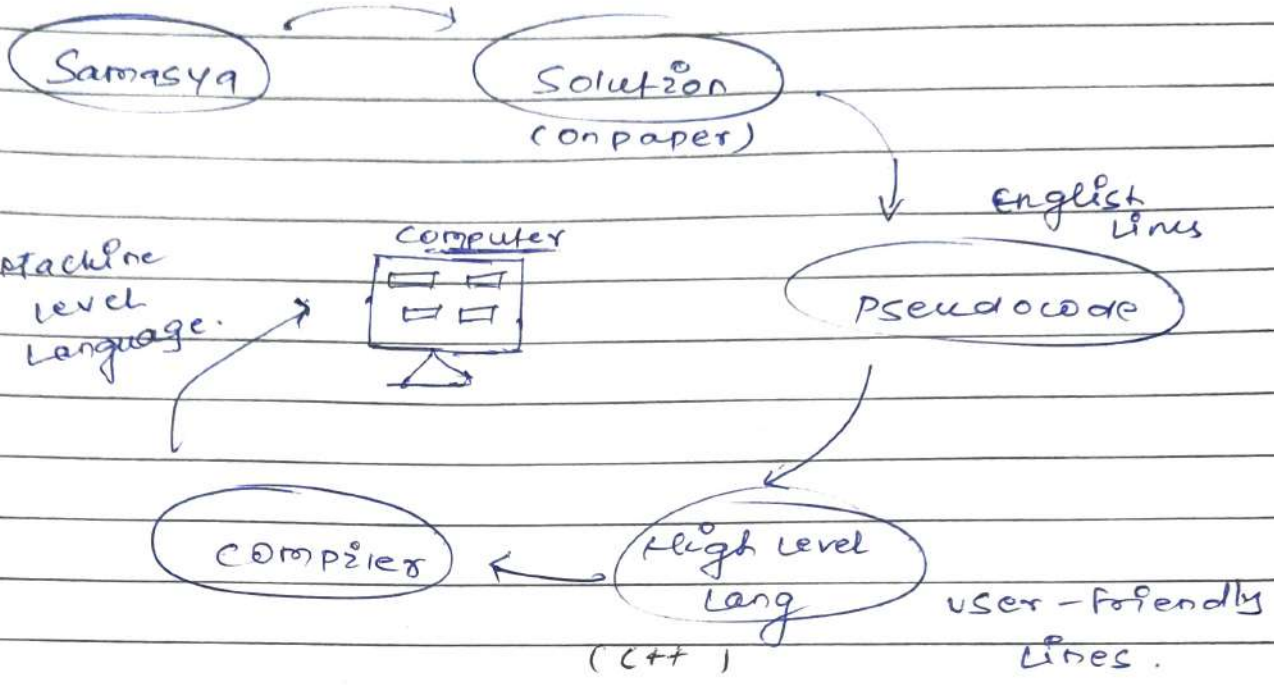
→ compiler (tool)
(which converts high level language into machine level language (0s & 1s)



→ 

↓
It can understandable by computer.

→ flow of a solving approach :



H/W

↳ Low level Language
↳ Mid level Language
↳ High level Language.

* Low level Language :- are the closest to the hardware.
:- They are machine oriented - means they are directly computer understandable.
:- This makes him very fast and efficient. But also very difficult to read & write.

eg :- Assembly Language

Mid level language :- Sit in the middle of the spectrum; but they are close to hardware
:- But they provide some abstraction by using keywords that are easier for humans to understand.

eg :- C Language.

High level Language :- are the furthest away from the hardware.
:- They are human oriented, meaning that, they use keywords & syntax that are similar Language.
:- It makes them user friendly.
:- But also makes low-efficient.

eg :- Python, Java.

:- The main difference b/w low, mid and high level Lang is the level of abstraction they provide.

:- Abstraction refers to the degree to which a Language hides the underlying hardware detail from the programmer.

flowcharts

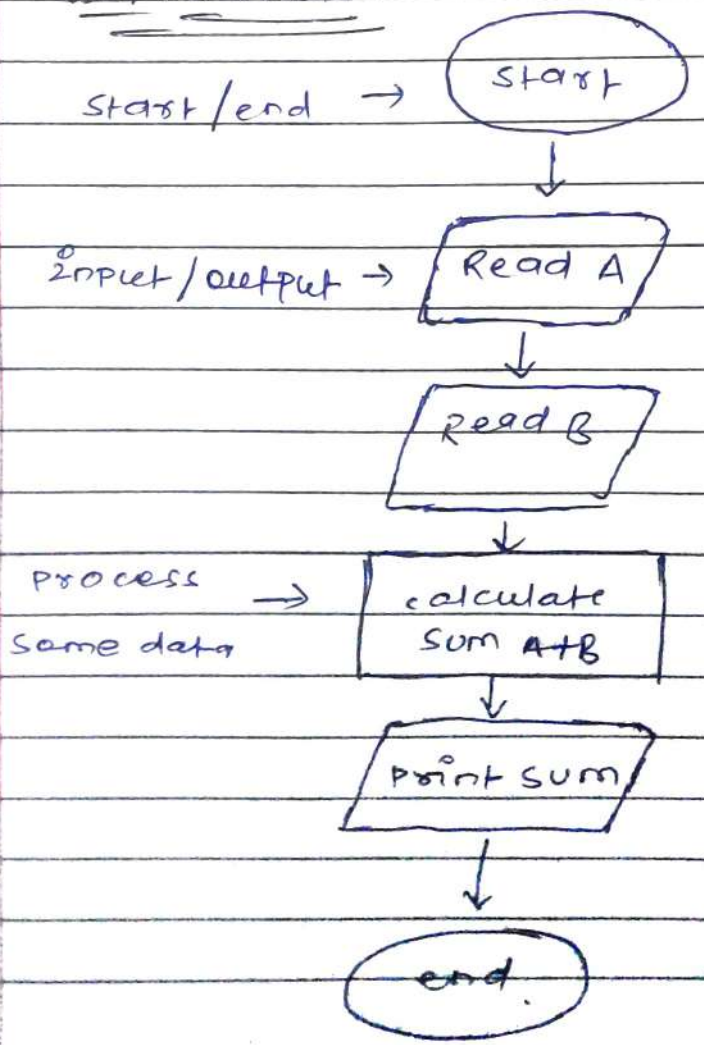
:- A flowchart is a type of diagram that represents an algorithm, workflow or process.

:- The diagramatic representation which shows steps as boxes of various kind, and their order by connecting the boxes with arrows illustrates a solution model to given problem.

:- used in analyzing, designing.

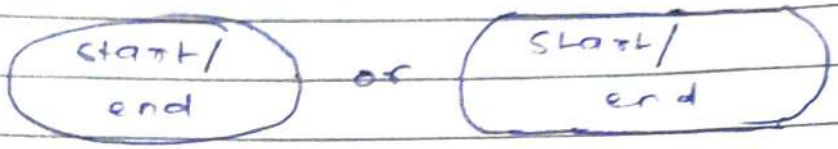
eg: Find sum of 529 and 256.

used to denote.



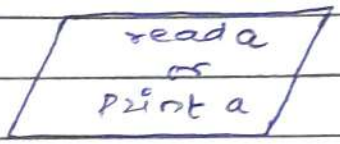
flowchart components :

① Terminator



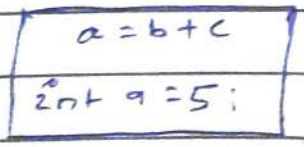
→ used to start or end to any flow of program.

② input / output block.



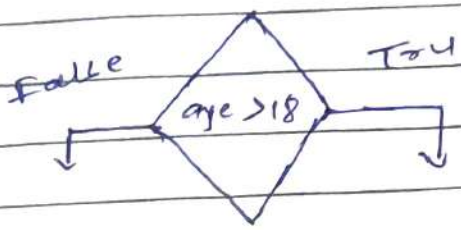
- used to take input and give output of any flow of program.

③ process block.



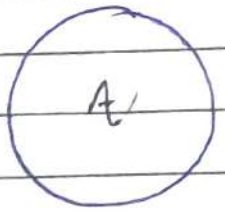
-used to calculate / initialize or declaration of variable and processing.

(4) Decision - making block.



- used to take decision on the respective inputs of programs.

(5) connector



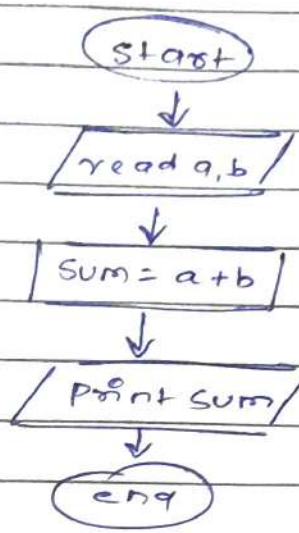
A is function

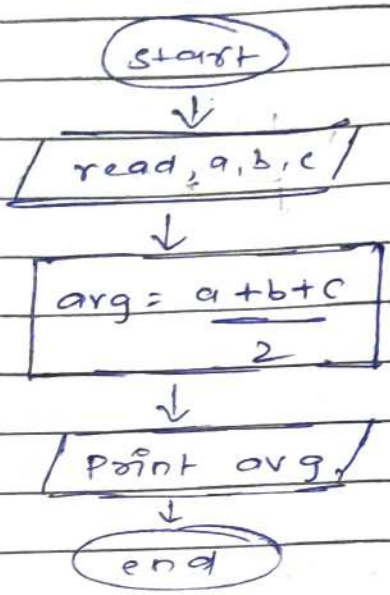
Used to connect different flowchart (Specially used in functions)

(6) Arrow.

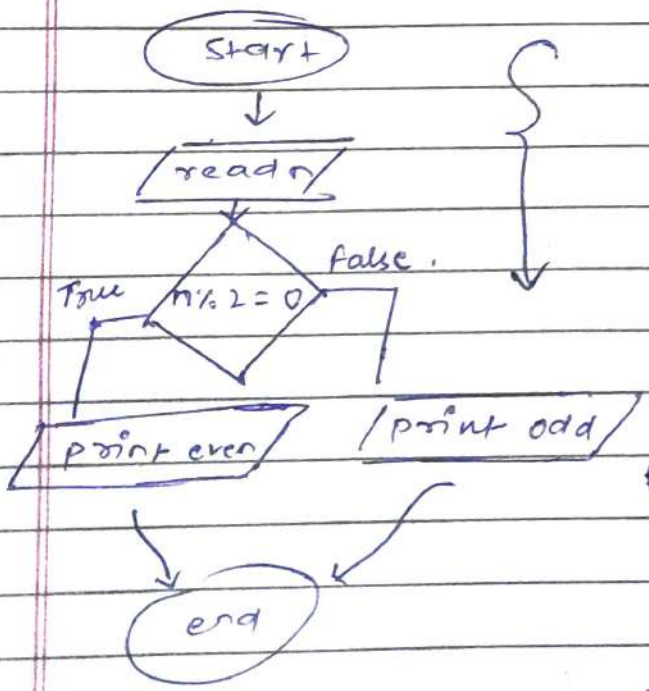
↓ → Shows the flow of Program.

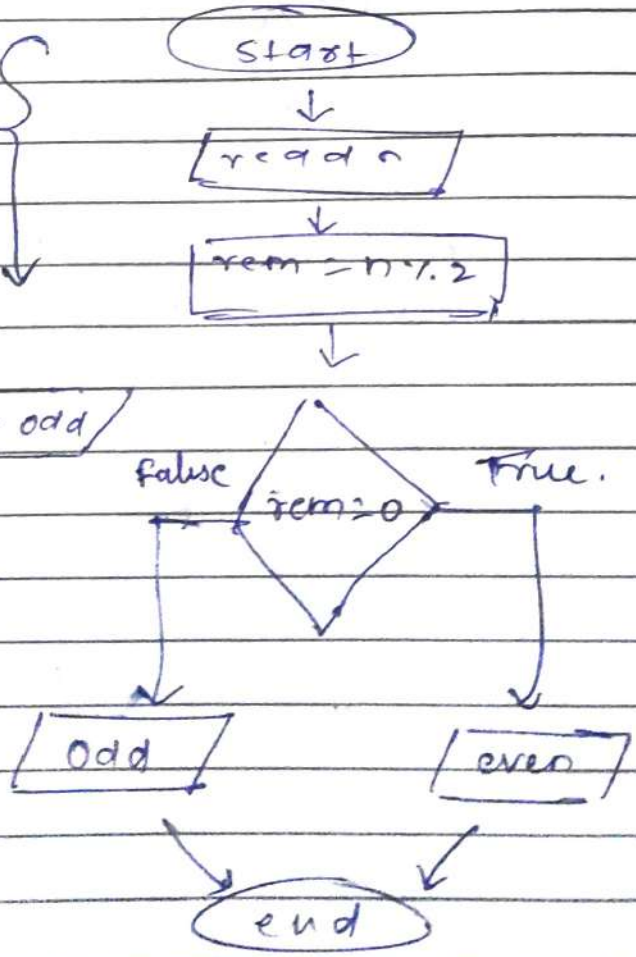
① print sum of a & b ② Avg of a, b, c.



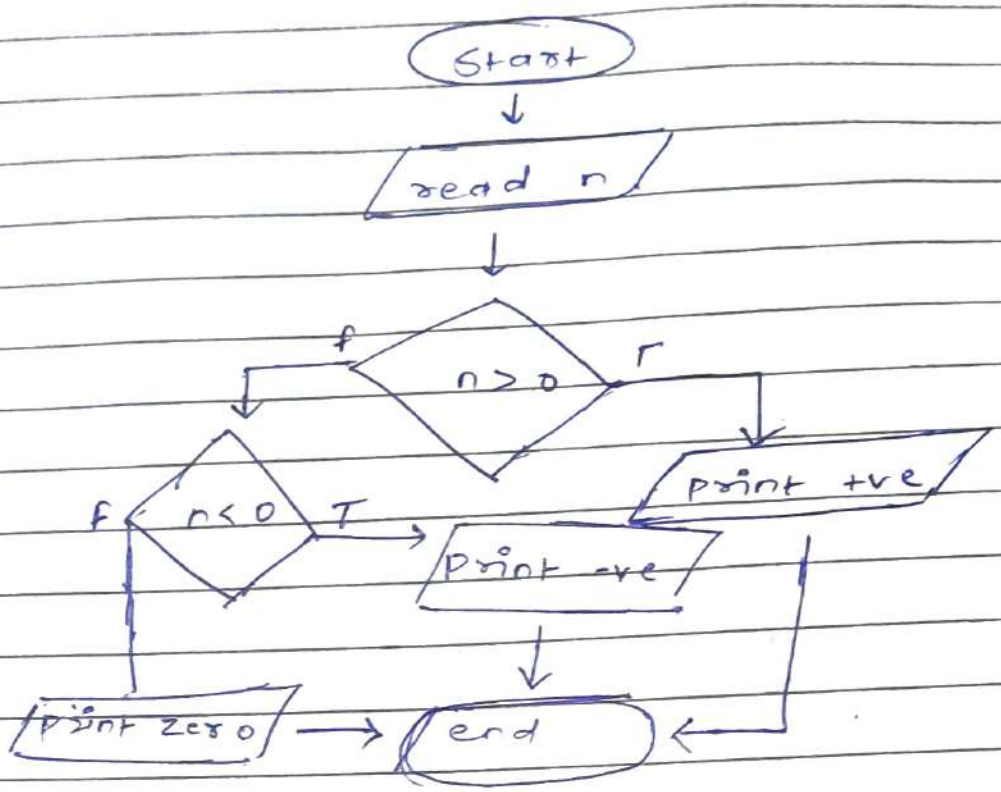


③ check num is Even or Odd.

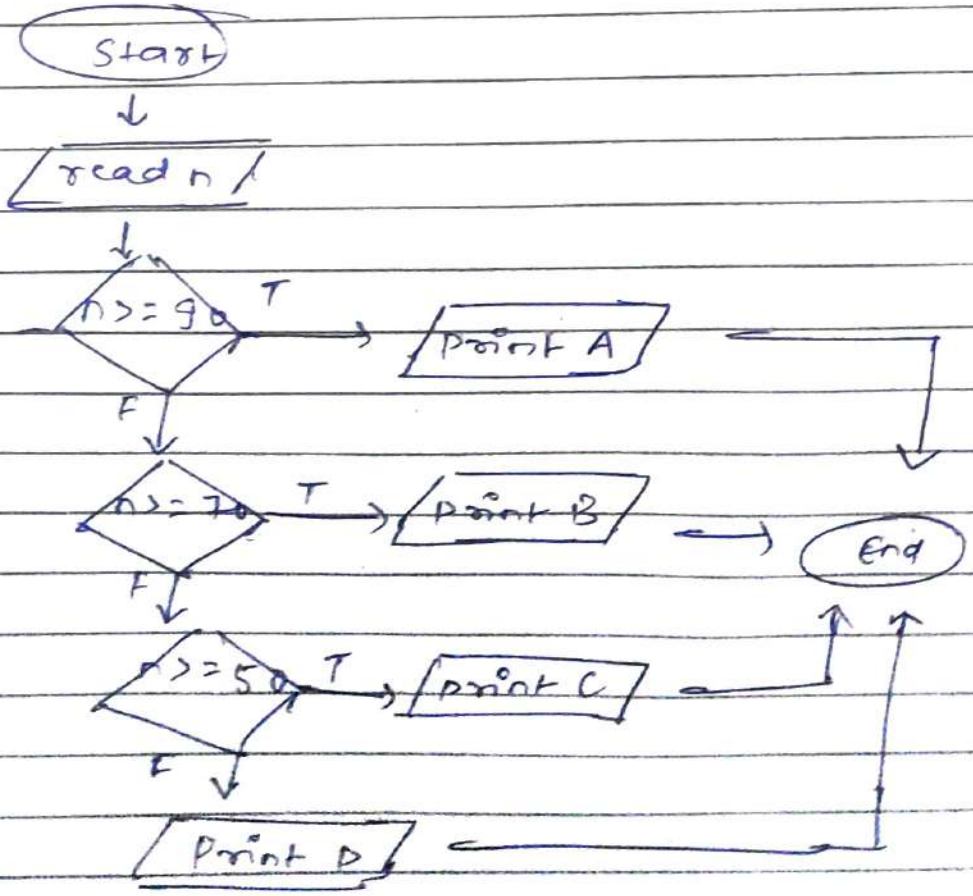




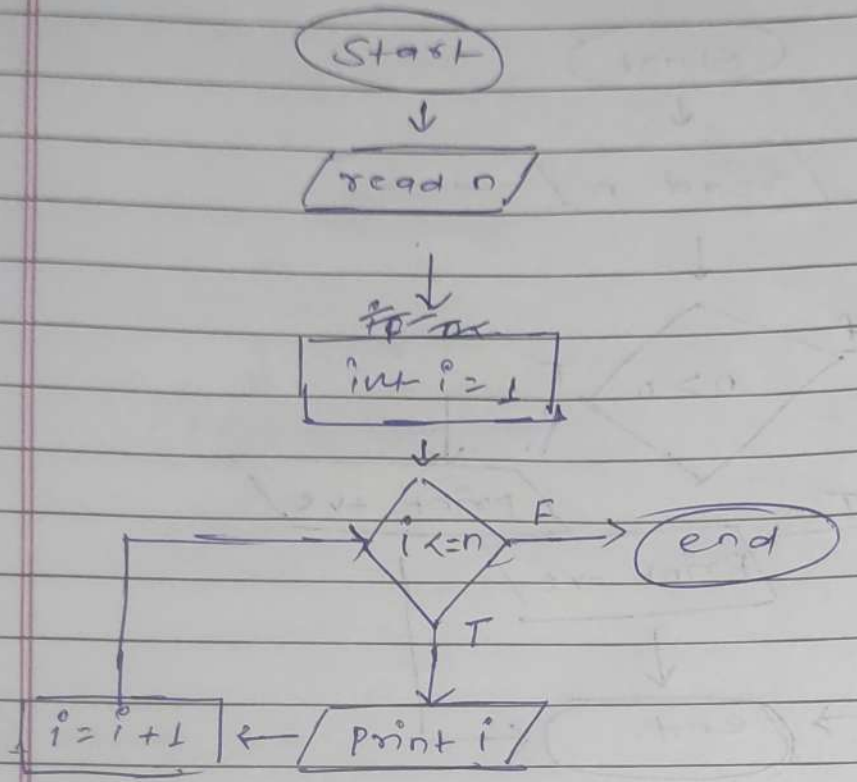
4. check +ve, -ve or 0.



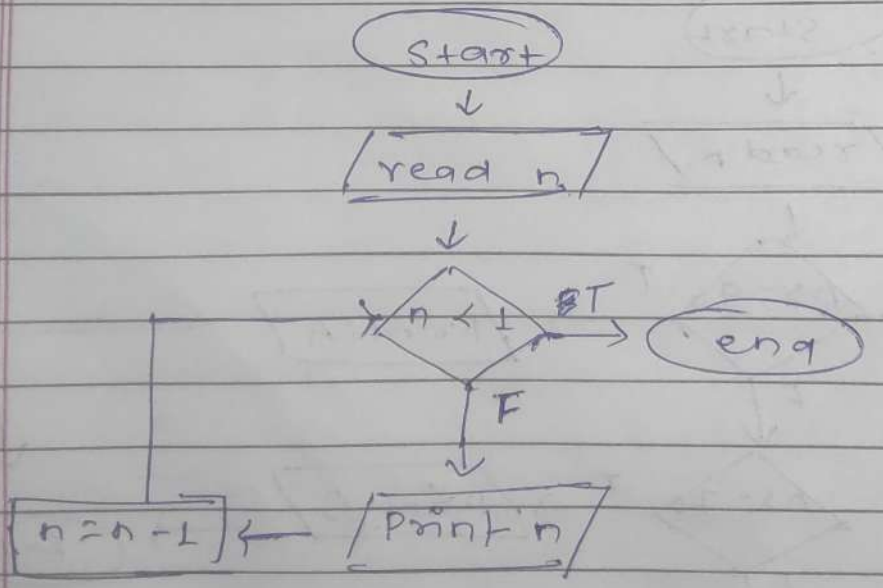
5. Student and Grade flowchart.



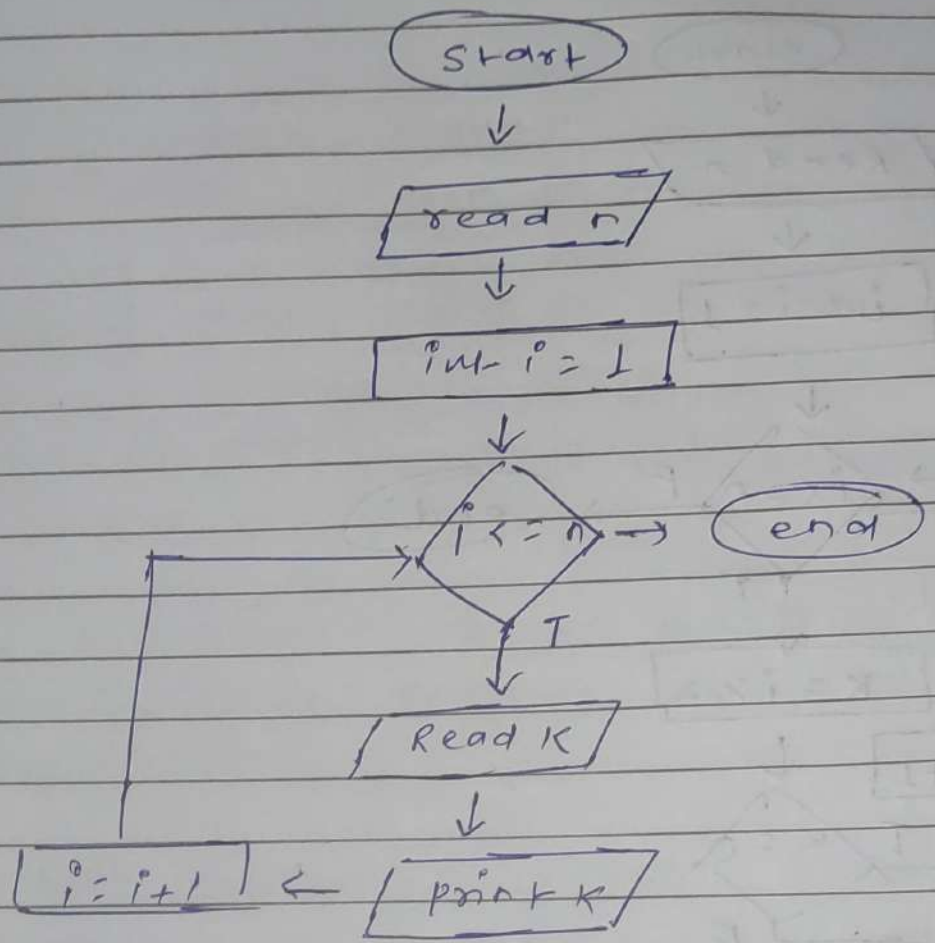
(6) print counting from 1 to N.



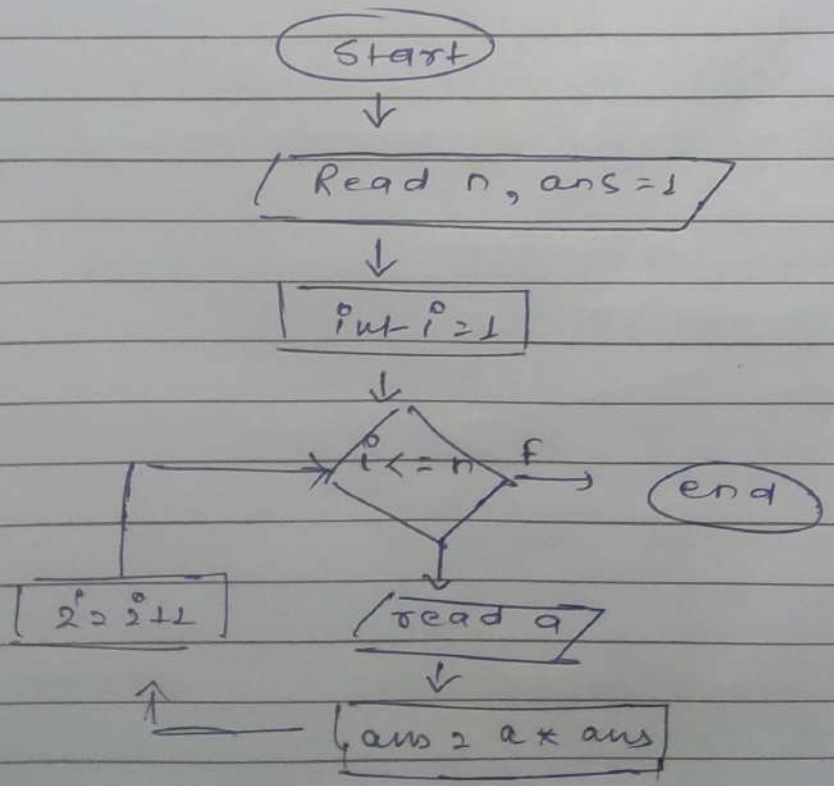
(7) print counting from N → 1.



(8) Multiply N numbers from user.



(9) Multiply N numbers from user.



(10) print 1 to N, but only Even numbers.