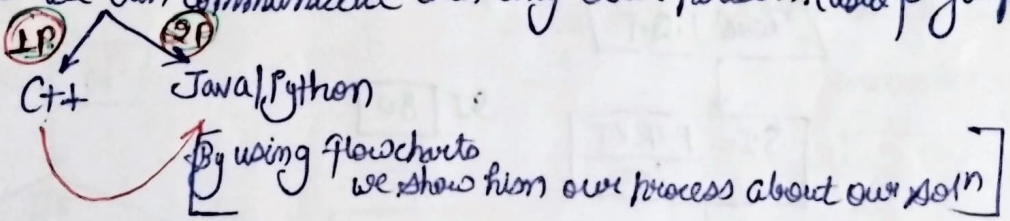
● Flowcharts : for problem solving :

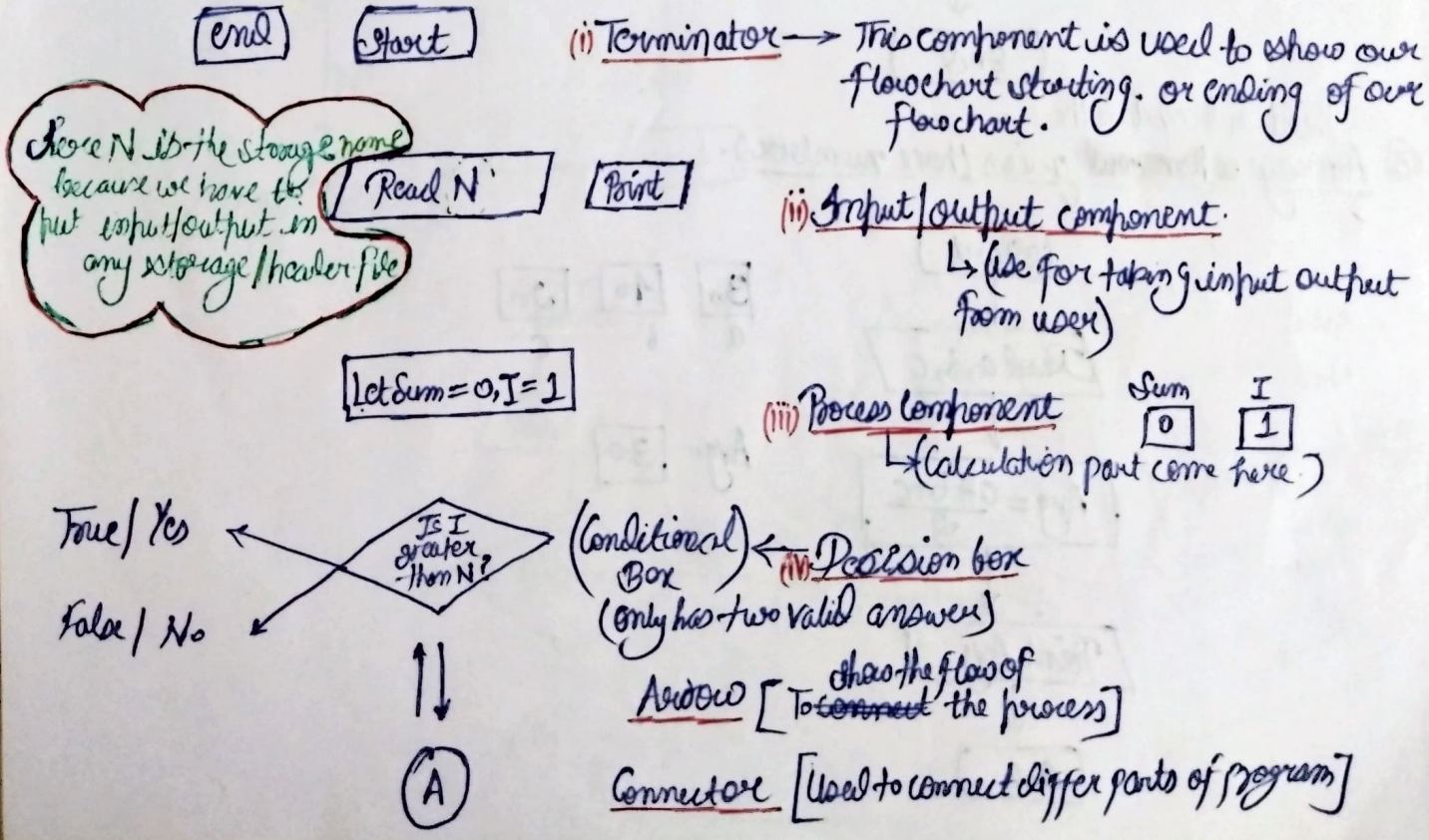
① To analyse your problem (so that we can know about input or outputt)
② Sometimes given problem is quite complex so we have to break down into sub-smaller parts.
③ To reduce the errors, we should write down soln on paper with required steps
④ Verif the soln (to verify the errors) [dry run]
(for verifing we need to take 2 or more example and see their outup
If coming right then we move ahead to code)
⑤ Write code

What is flowchart ? :

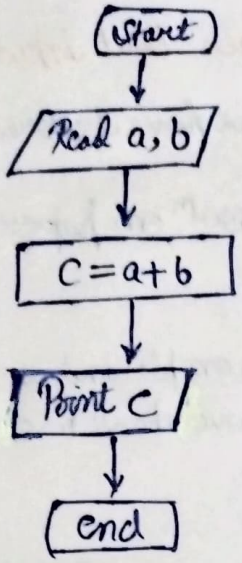
- Diagrammatic representation illustrating a soln to a given problem.
- Allows you to break down any process into simple-smaller parts and display them in a visually pleasing way.
- By flowcharts we can communicate with any other person. (used for group projects)



Flowchart components :-



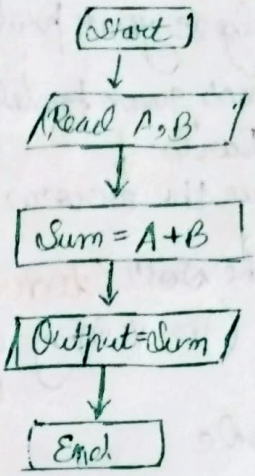
e.g. ① Add two numbers



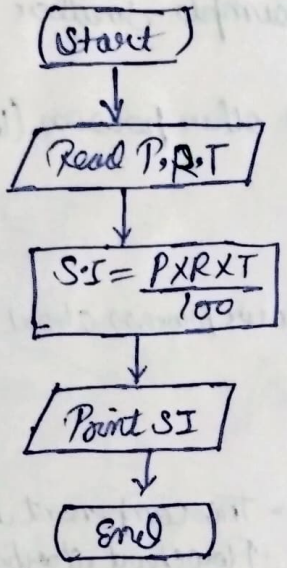
(Print use read to use output component)

5 (a) 4 (b)

9 (c)



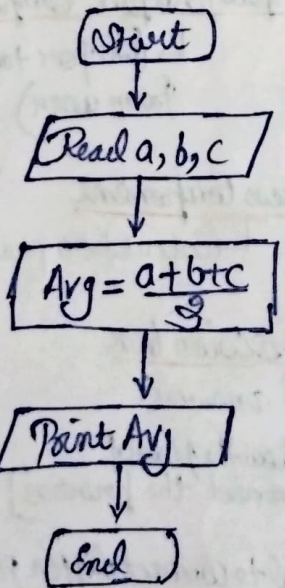
② Read Principal, Rate and Time or Print SI.



1000 (P) 4 (R) 2 (T)

SI 80

③ Average, You are given three numbers. Print and calculate the

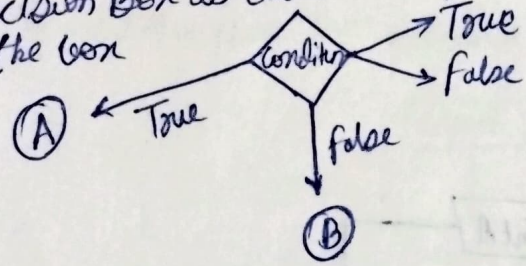


30 (a) 40 (b) 20 (c)

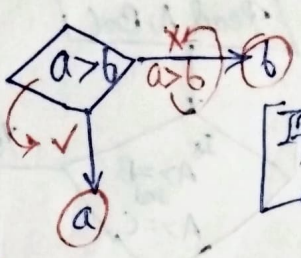
Avg = 30

- **Decision Making:**

Decision Box is diamond shaped box and we need to write the condition inside the box



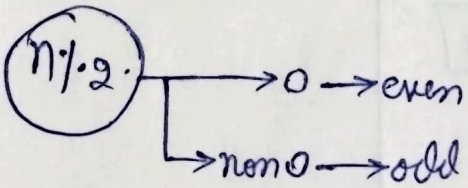
only one answer possible.



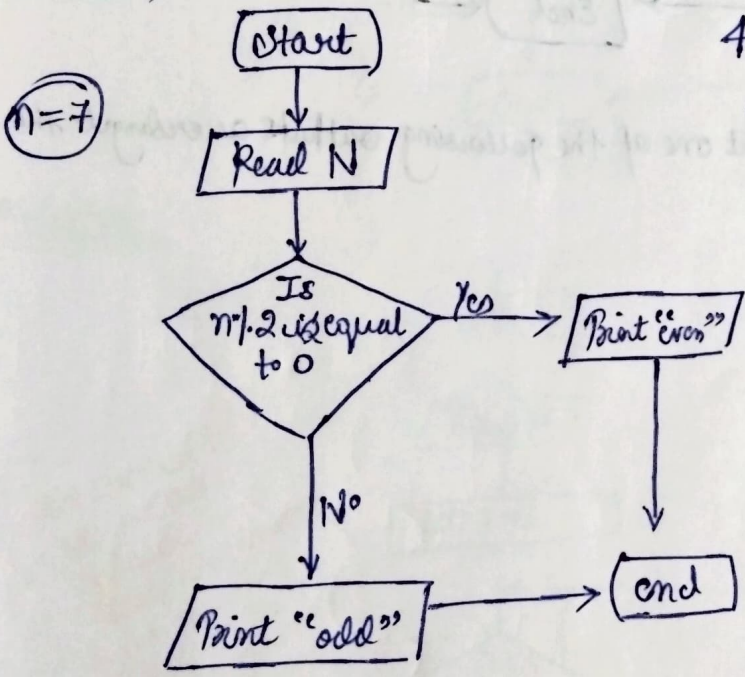
If a greater than print a or otherwise print b

e.g. Check whether number is odd or Even

n = 7



Flowchart ⇒



/ → (divide operator)

5/2 = 2.5 → 2 (ignoring decimal parts)

4/3 = 1

10/2 = 5

(we get quotient here but need of remainder)

for Remainder ①

4%.2 = 0

2 | 4 | 2
4
0

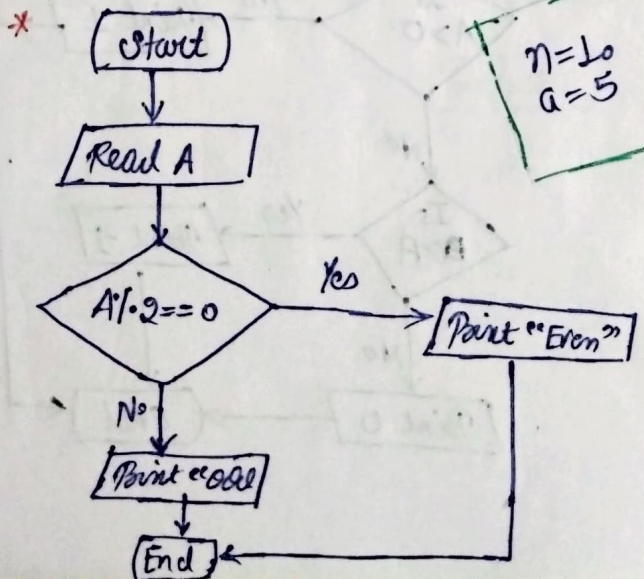
% → modulus operator or Remainder operator

5%.3 = 2

3 | 5 | 1
3
2

3%.3 = 0

3 | 3 | 1
3
0



n = 10
a = 5

10 (n) 5 (a)

n == a → Yes / No

equality check operator

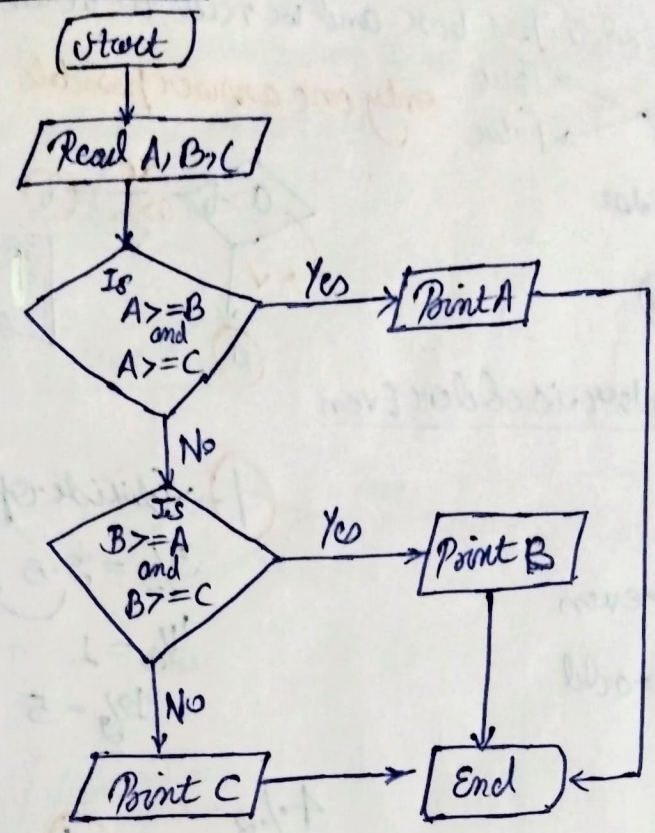
5 == 5 → Yes

5 == 2 → No

n = 5, a = 4

n == a → No

Eg. Find largest of three numbers.



a=4
b=2
c=1

a=4
b=8
c=5

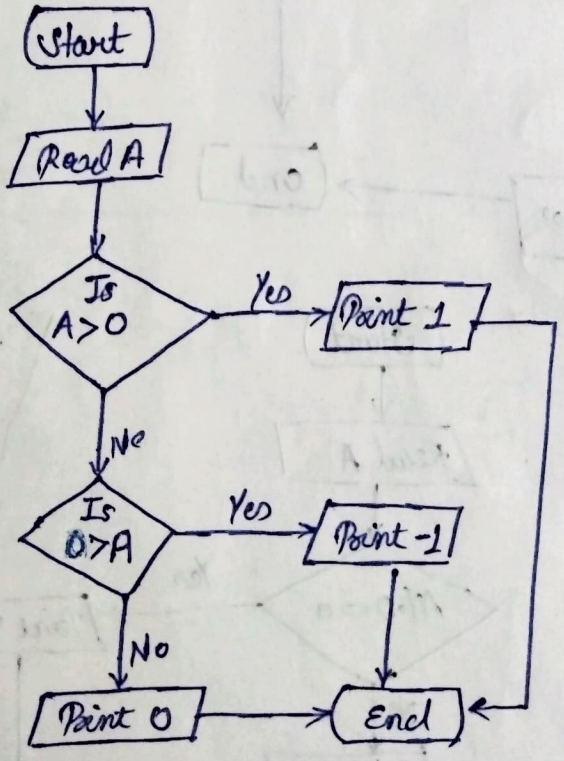
a>b
No
b>a
No

a=5
b=5
c=4

So we have to use equal sign also with this

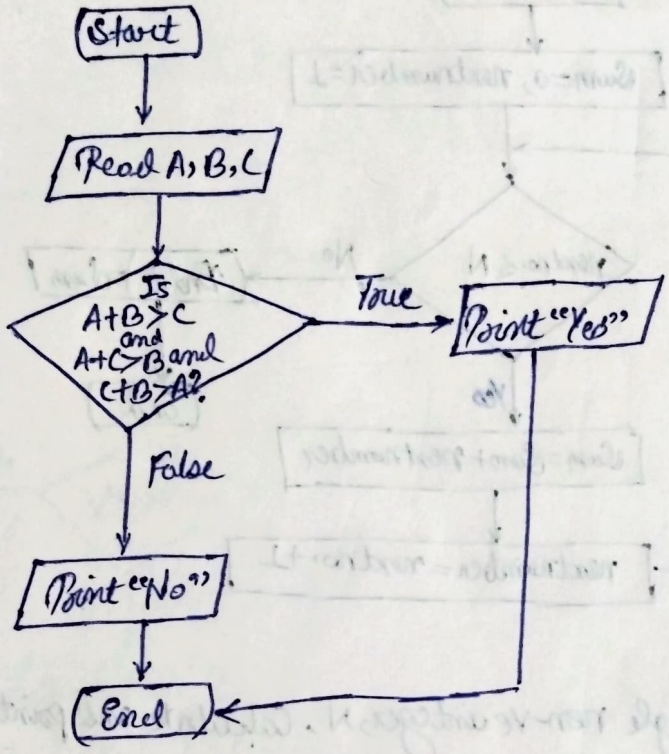
Q(1) You are given a number. You need to print one of the following outputs according to the number's nature.

Print 1, if the number is positive
Print -1, if the number is -ve
Print 0, if it's equal to zero.



Q(2) You are given a three numbers. Each numbers represent the length of a line. You need to figure out whether these lines can form a valid triangle.

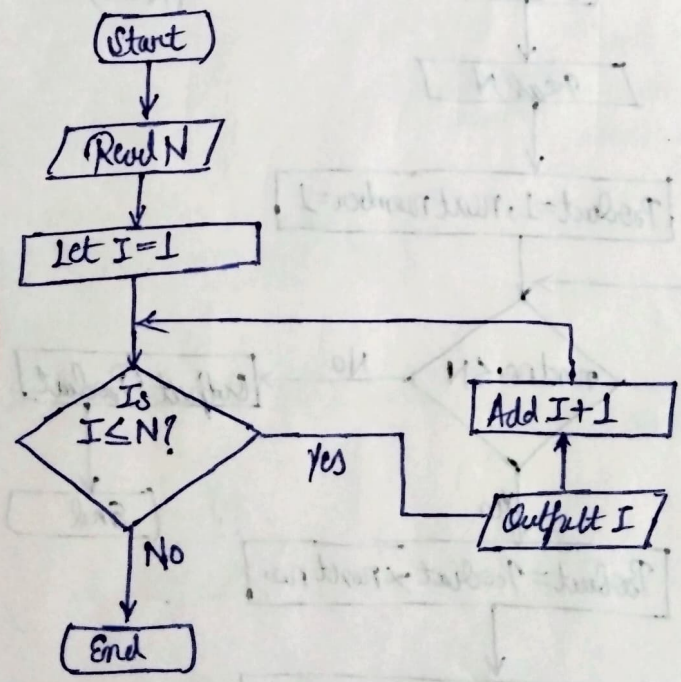
- If valid then print "Yes" otherwise Print "No"



- **Loops:**

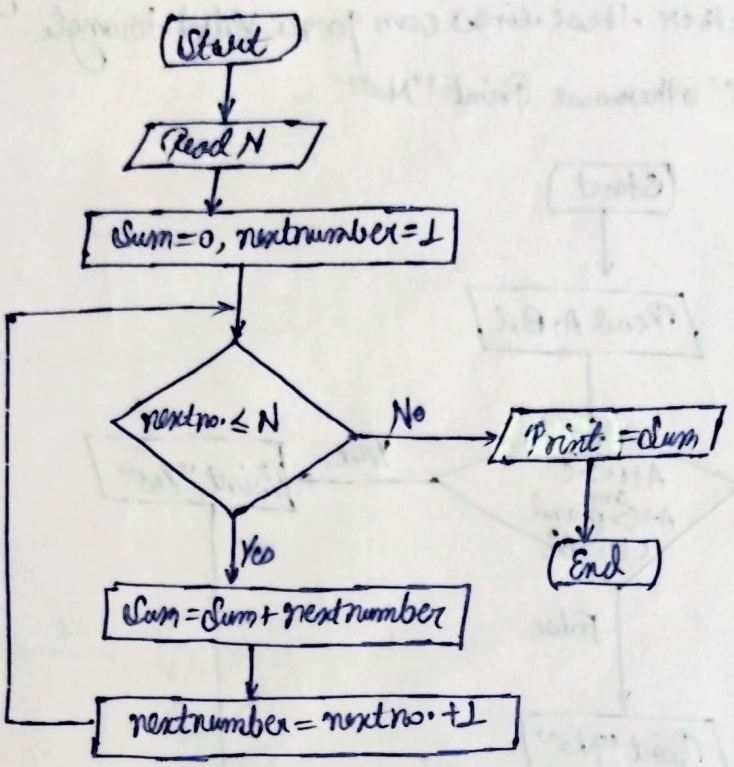
N=5

Here, we need to execute a set of instructions multiple times. So we add a loop. So that we don't need to do/write the thing again & again



1
2
3
4
5

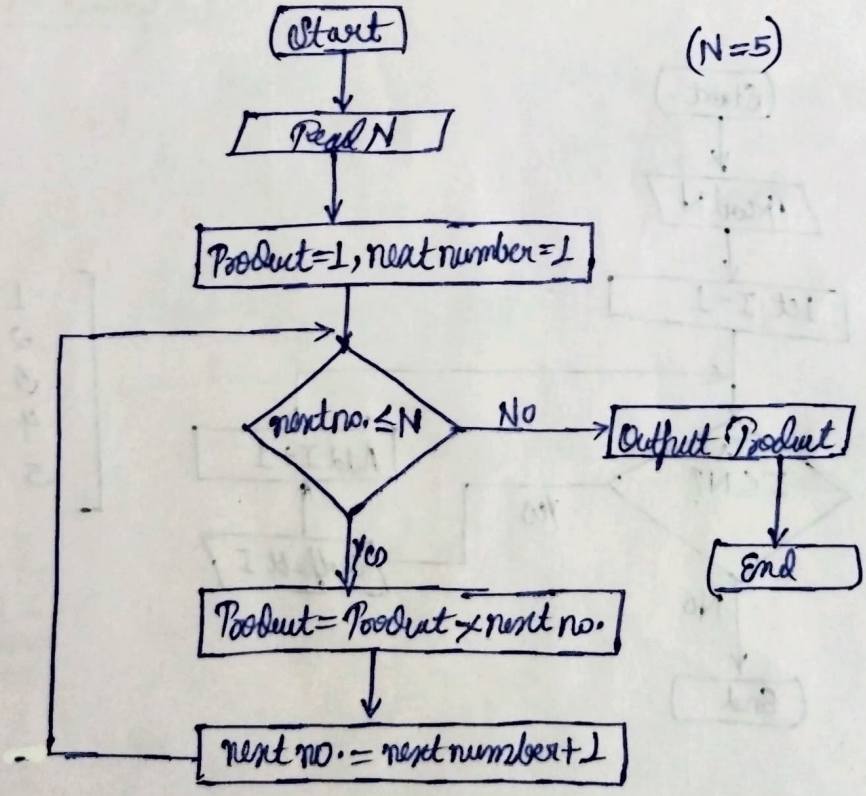
Q(1) Find sum of Numbers from 1 to N



[1+2+3+4+5 = 15]

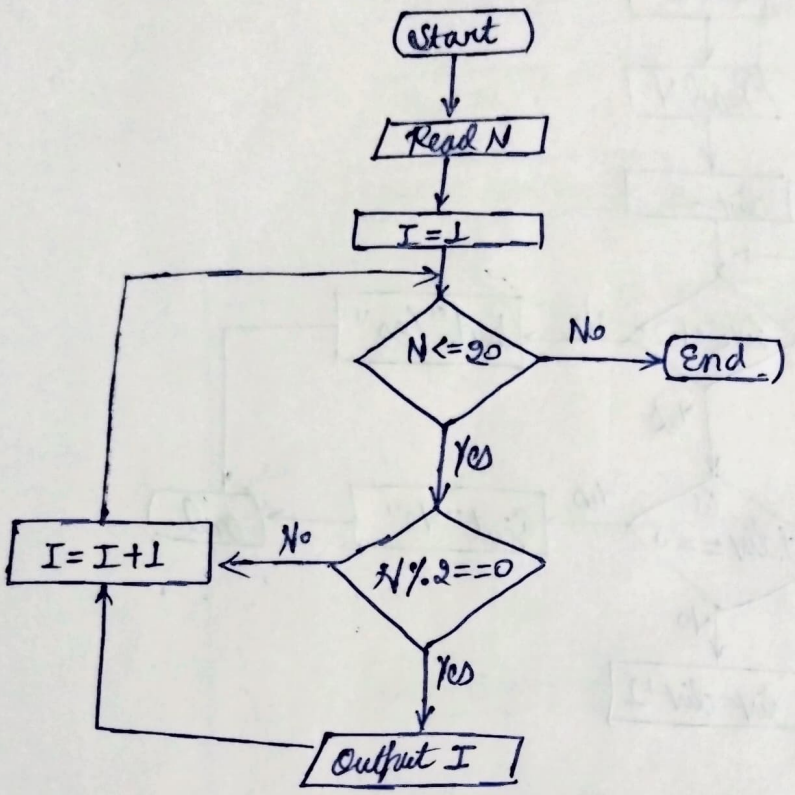
[Sum = 15]

Q(2) You are given a single non-ve integer, N. Calculate and print N!
N! is defined as the product of all integers from 1 to N



Q(3) Given a +ve integer, N. Then print all even integer that occur b/w 1 to N. (4)

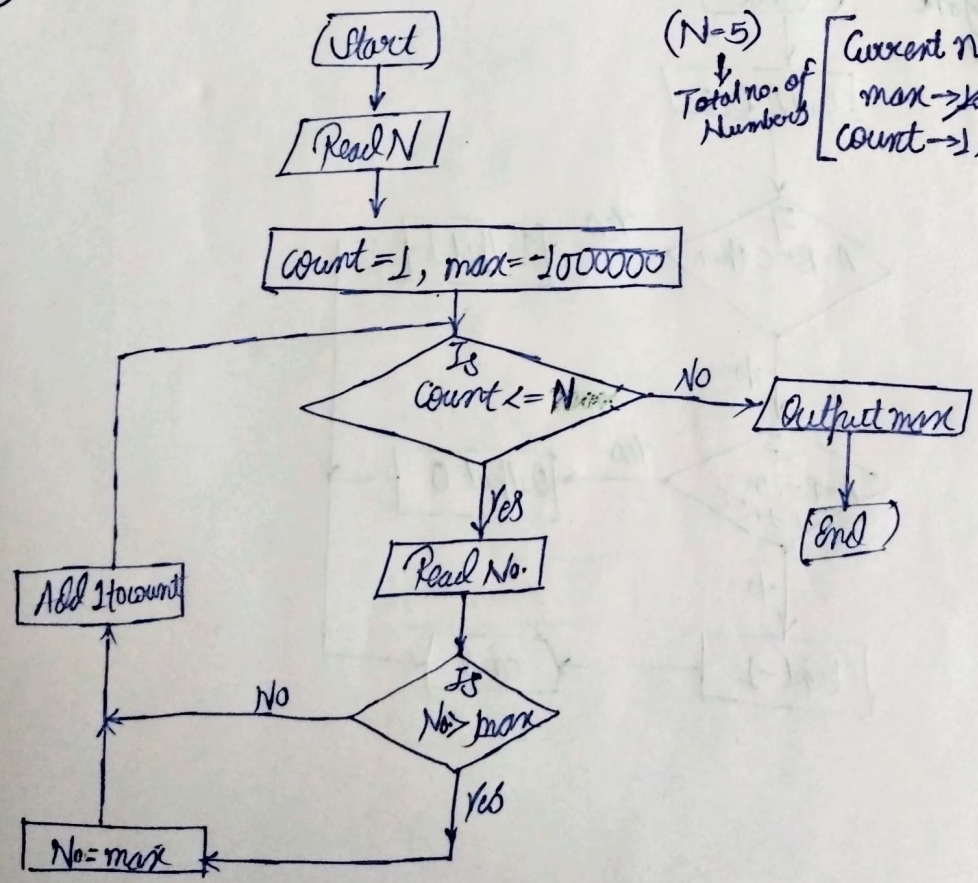
(N=20)



Output

2
4
6
8
10
12
14
16
18
20

Eg* Largest of N numbers.



(N=5)
↓
Total no. of Numbers

Current no. → 10 15 14 20 18
max → 10 15 20
count → 1, 2 3 4 5 —

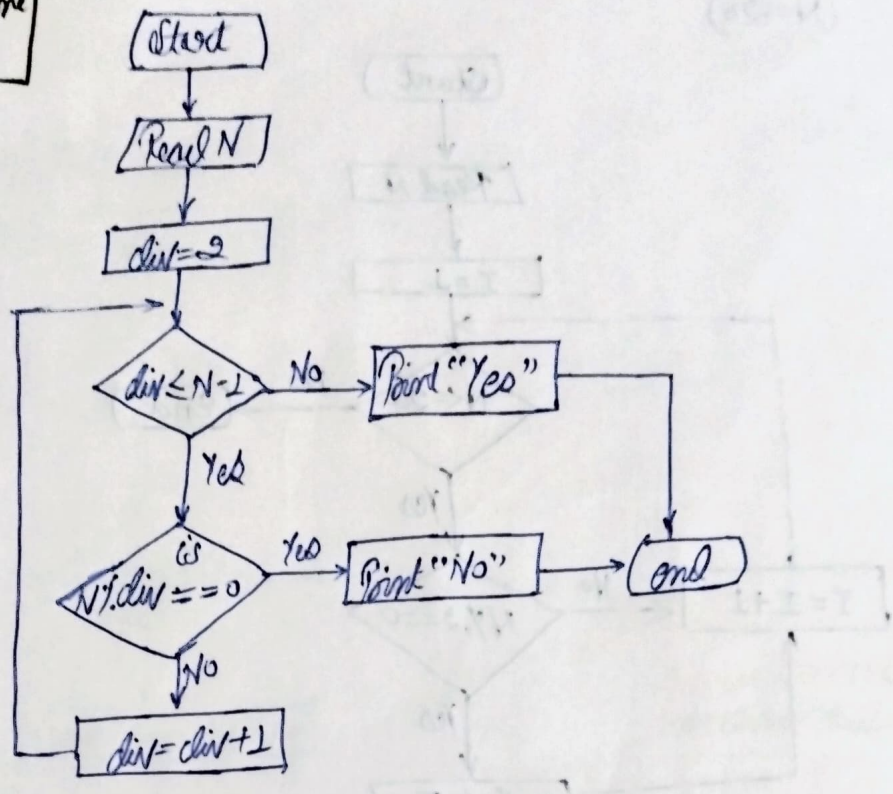
max=0
↳ then if we give -ve no. then it will give max as 0 that is incorrect

Read current no.
max = current no.
↳ can't go with this, because if the value of N=0 then it will ask for current no. but the value of N=0

- Check Prime if given a number N.

[Print "Yes" if number is prime
Print "No" if number is odd]

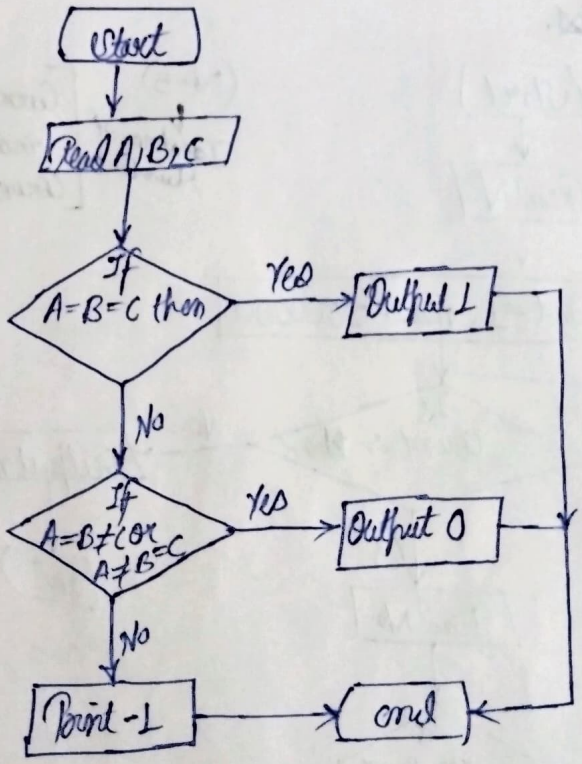
Assume N >= 2



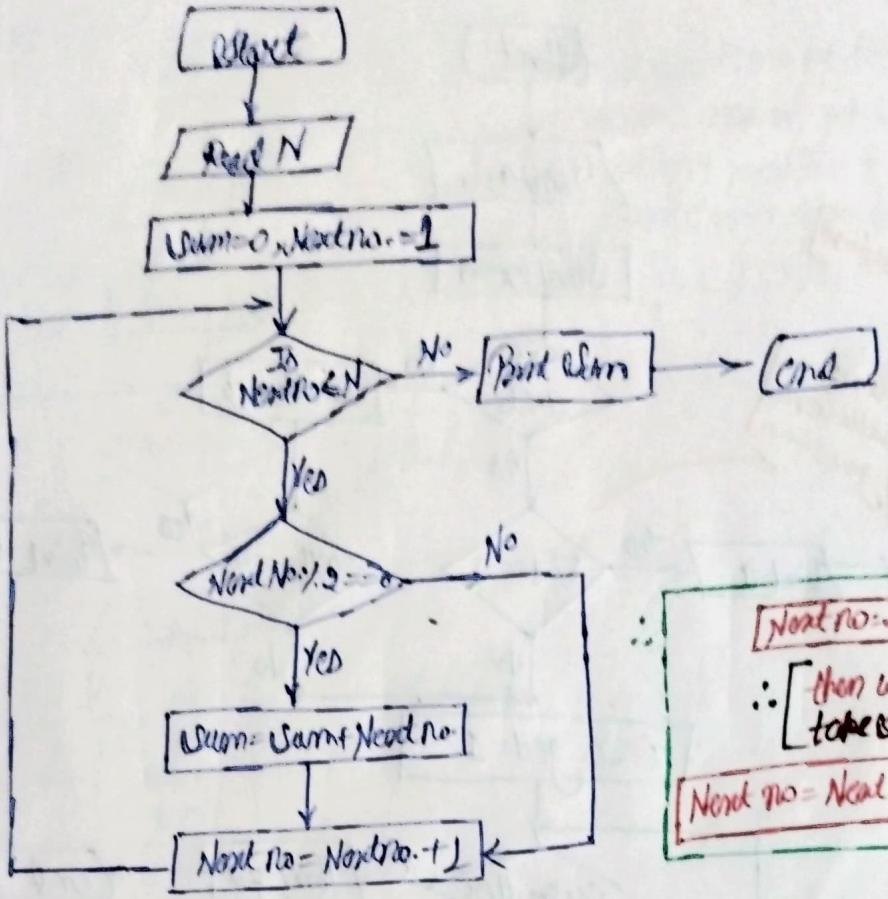
*
[div <= N/2
div <= √N]

Check Triangle:

- given the lengths of 3 sides of Valid Δ. You need to print any one
  Print 1, if Δ is equilateral
  Print 0, if Δ is isosceles
  Print -1, if Δ is scalene



Sum of evens:

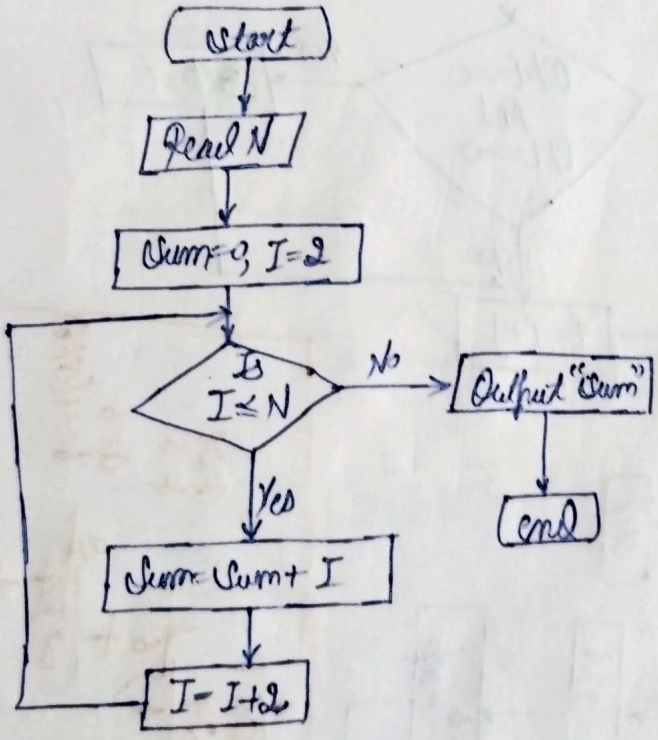


∴ Next no=2

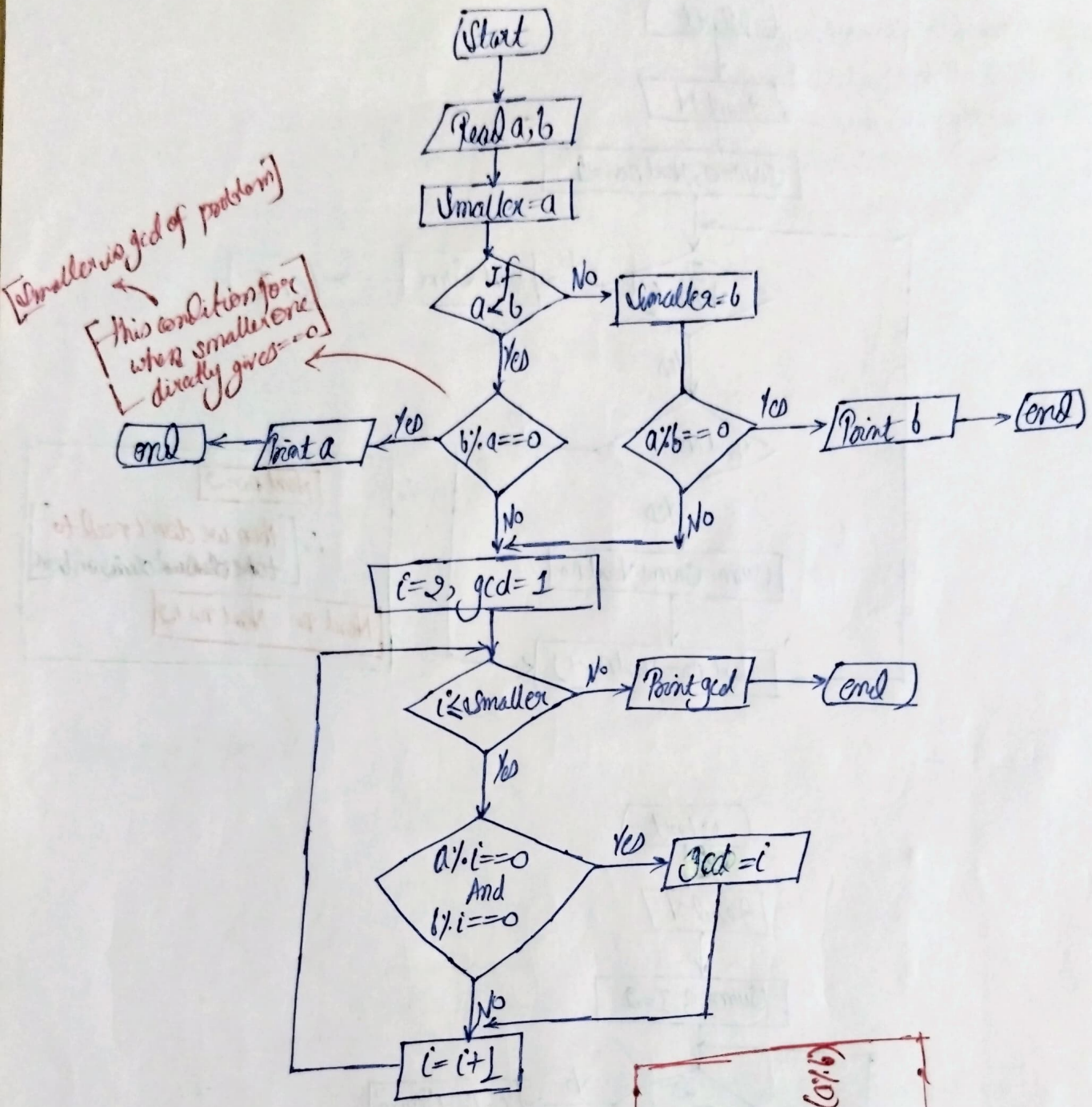
∴ [then we don't need to take second decision box]

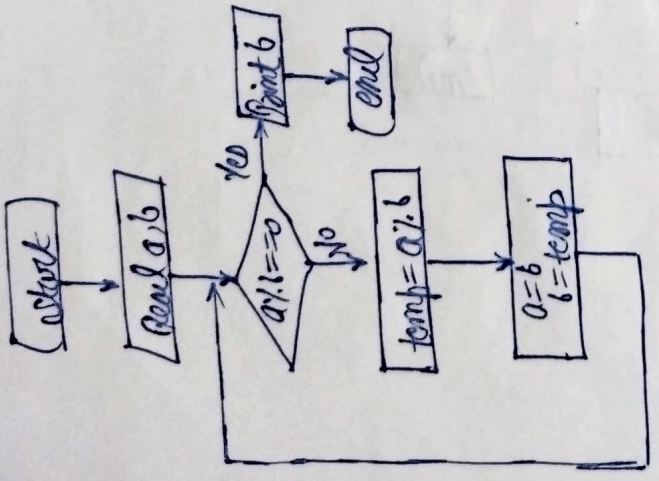
Next no=Next no+2

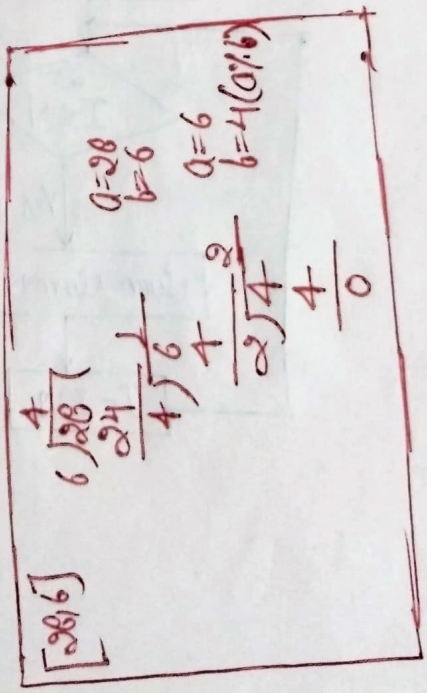
*



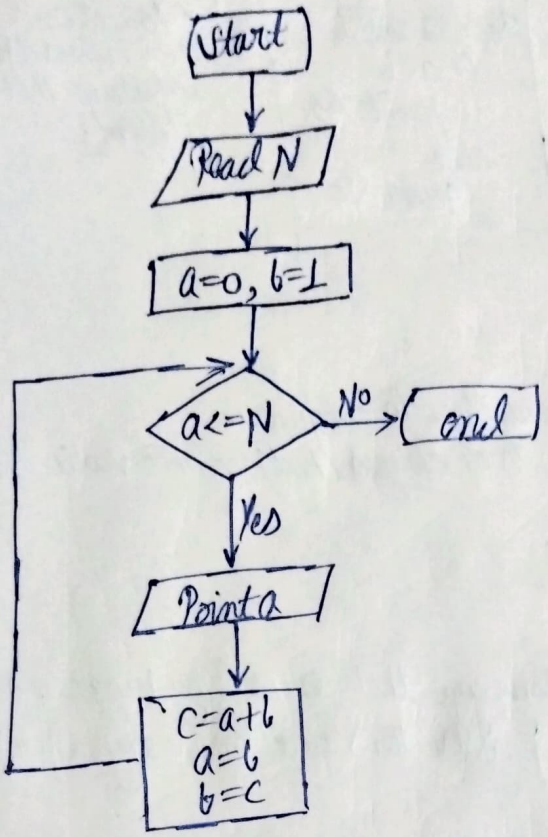
Find GCD : (Greatest Common Divisor)







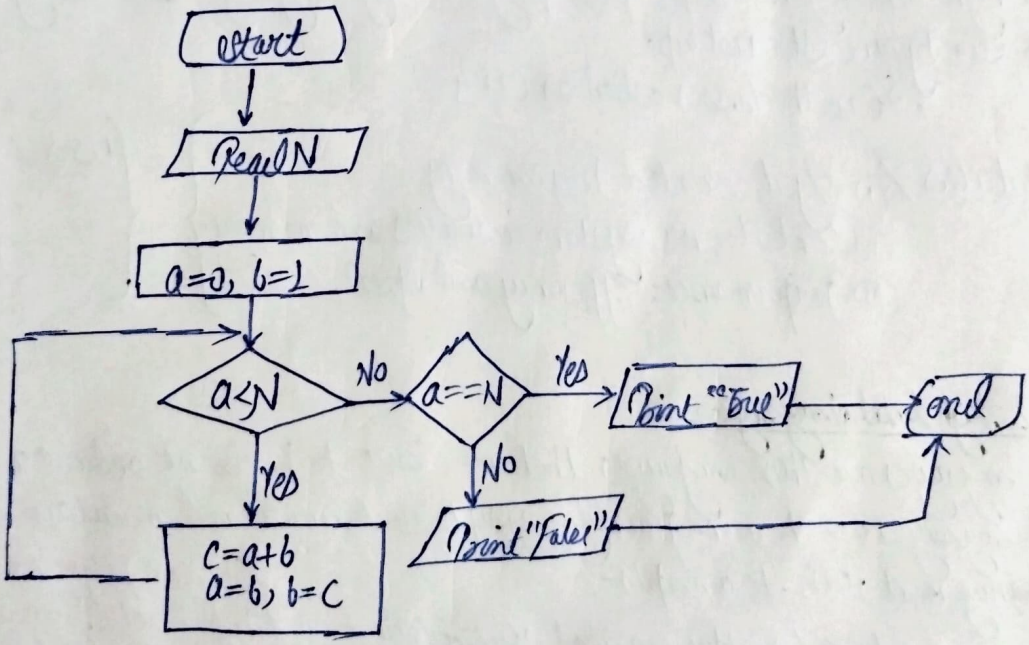
Fibonacci numbers:



[First two term of Fibonacci sequence is 0,1 and to get the third one by adding previous two numbers and so on, same for all upcoming no.]

[0, 1, 1, 2, 3, 5, 8, 13, 21 - - -]

Member of Fabonacci:



Pseudocode

It is a way to represent the implementation of an algorithm or logic or any approach.

ex. Sum of 2 no.

```
→read 2 no., a and b
→sum = a + b
→print sum
```

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

Both are correct

```
Read a and b
  if a<b
    print Yes
  else
    print No
```

[• Pseudocode is a very simple and high level form of computer language that is used in program design.]

What is Programming language?

- Programming language is a way to communicate with 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.

or

- A programming language is a computer language that is used by programmers (developer) to communicate with computers. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task.

C++:

- C++ is a high-level, general-purpose programming language created by → Danish computer scientist Sir Bjarne Stroustrup.
  - "C with classes called as C++"

Features ⇒ (i) Object oriented Programming
(ii) Control over system resources and memory.
(iii) Performance: Efficiency and speed.

Uses ⇒ • Software development
• Games development
• Real time Systems

Low level & High level languages:-

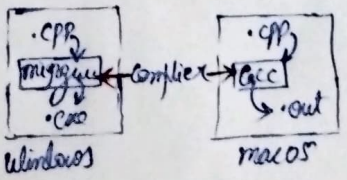
- Low level languages are those languages that are closer to hardware and are more machine oriented.
- High level languages are human-friendly, easy to read, write and maintain. It includes automatic memory management, OOPs etc concepts.

Ex: Low level ⇒ Assembly language & Machine code
High level ⇒ C++, Java, Python.

{C++ can perform both low-level and high-level programming.
• C++ is a high level programming language with low level capabilities.}

- C++ is platform dependent language.

{If someone wants to run a program then first it needs or have to be compile before running in process}



code.cpp ← source file
code.exe ← program