Lesson Number : 4

Flowcharting :

The most difficult and important task within programming is the systematic and careful analysis of a whole problem. Therefore before going to actual programming, a programmer should always go through the following steps in a sequential order.

i. Design an algorithm representing the process of solution of problem.

ii. Represent this algorithm through flowchart for better understanding of the algorithm.

iii. Code the flowchart i.e. write instruction in a programming language that a specific computer will accept.

iv. Run / Execute the program on the computer for the given data (input) and get the output.

Flowchart :

→ Flowchart is a graphical representation of an algorithm.

→ It makes use of the basic operations in programming.

→ All symbols are connected among themselves to indicate the flow of information and processing.

→ It is a quality improvement tool for specifying the process of the program.

→ It tends to provide people with a common language or reference point when dealing with development of a program.

[Reference: Computer fundamental & C programming, Amiya Kumar Rath]

History and Background :

As a whole, Flowcharting has been around for a very long time. In fact, Flowcharts have been used for so long that no one individual is specified as the "father of the flowchart". The reason for this is obvious. A flowchart can be opti customized to fit any need or purpose. For this reason, flowcharts can be recognized as a very unique quality improvement method.

Instructions for development of a flowchart :

Step-by-Step process of how to develop a flowchart :

- Gather information of how the process flows : use
  - i. Conservation
  - ii. experience
  - iii. product development codes.
- Trial process flow
  - i. Allow other more familiar personnel to check for accuracy.
  - ii. Make changes if necessary.
  - iii. Compare final actual flow with best possible flow.

Construction/ Interpretation tip for a flowchart :

- Define the boundaries of the process clearly.
- Use the simplest symbols possible.
- Make sure every feedback loop has an escape.
- There is usually only one output arrow out of a process box. Otherwise, it may require a decision diamond.

Interpretation :

- Analyze flowchart of actual process.
- Analyze flowchart of best process.
- Compare both charts, looking for areas where they are different. Most of the time, the stages where differences occur is considered to be the problem area of DR process.
- Take appropriate in-house steps to correct the differences between the two separate flows.

Flowchart Symbols :

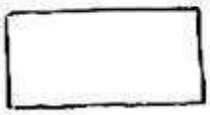
i. Terminal :

→ The oval represents any terminal point in a program and generally contains words such as BEGIN, START, END, or STOP.

ii. Input / Output :

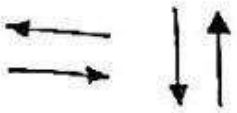
→ The parallelogram represents the Input/Output function i.e. making data available for processing (input) or recording of the processed information (Output). This step implies obtaining a number from an Input device (the keyboard and storing it in the storage location named 'A').

iii. Process :



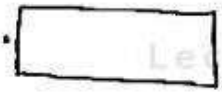
→ The rectangle represents the processing operation. A process changes or moves data. An assignment is normally represented by this symbol.

iv. Flow direction :



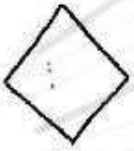
→ Lines or arrows represent the flow direction function – the flow of control. Normal flow direction is from left to right or from top to bottom.

v. Annotation :



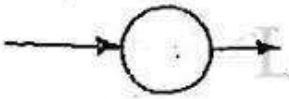
→ A broken line and bracket represents the annotation function – the addition of descriptive comments or explanatory notes for clarification of some statements.

vi. Decision making Symbol :



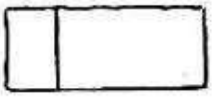
→ The diamond represents a decision or switching type of operations that determines which of the alternative paths is to be followed.

vii. Connector :



→ The circle represents a function in a flow line.

viii. pre-defined process :



→ The double sided rectangle represents a named process that consists of one or more operations or programming steps that are specified elsewhere, such as a module or subroutine. This flowcharting symbol is used for representing algorithm.

ix. Looping :

→ The oval represents the looping which is repeated based on a condition / value of a variable..

Example: Write the algorithm and draw the flowchart to find the sum and product of given two numbers.

Algorithm:

Step 1 : Read a, b

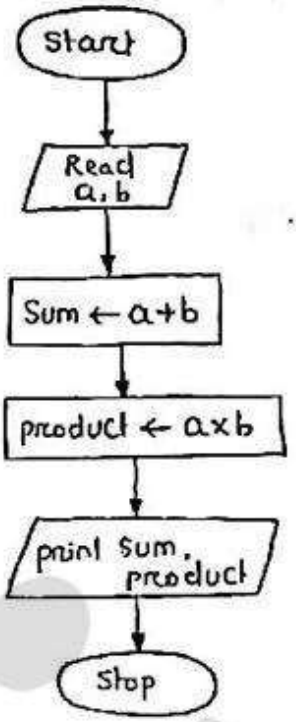
Step 2 : Sum ← a+b

Step 3 : product ← a x b

Step 4 : print sum, product

Step 5 : Stop.

Flowchart



Example: Write the algorithm and draw the flowchart to convert the temperature in F to C using the formula $C = \frac{5}{9}(F-32)$.

Algorithm:

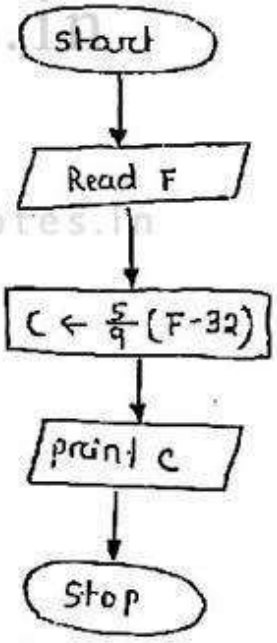
Step1 : Read F

Step2 : $C \leftarrow \frac{5}{9}(F-32)$

Step3 : print C

step 4 : stop

Flowchart



Example: write the algorithm and draw the flowchart to find the area of triangle whose sides are A, B, C.

Algorithm:

Step1 : Read A, B, C.

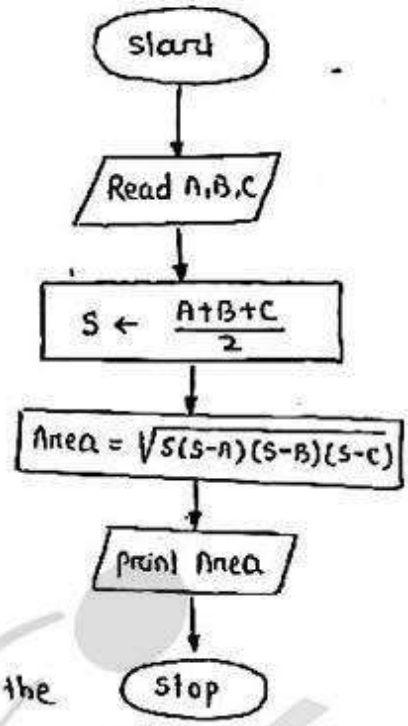
Step2 : $S \leftarrow \frac{A+B+C}{2}$

Step3 : $Area \leftarrow \sqrt{S(S-A)(S-B)(S-C)}$

Step4 : Print Area

Step5 : Stop

Flowchart



Example: write the algorithm and draw the flowchart to find the biggest of the given three numbers.

Algorithm:

Step1 : Read a, b, c

Step2 : if a > b then

big ← a

Step3 : if a > c then

big ← a

Step4 : else big ← c

Step5 : else if c > b then

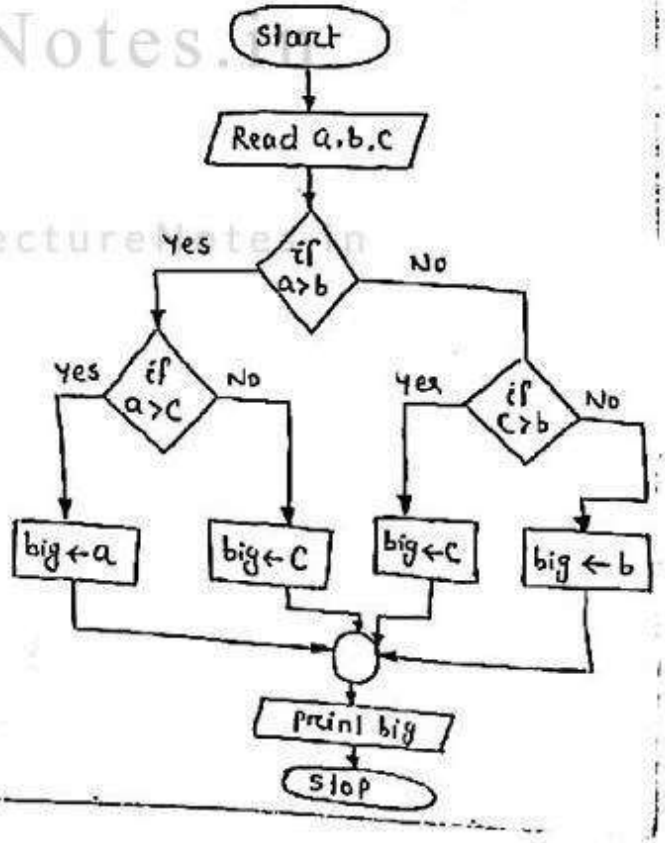
big ← c

Step6 : else big ← b

Step7 : print big

Step8 : Stop

Flowchart



Example : Write an algorithm and draw the flowchart to solve the following series (sinx).

$$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + \frac{x^n}{n!} \cdots$$

Algorithm :

Step1 : Read x, n

Step2 : $s \leftarrow 0$
$term \leftarrow x$
$i \leftarrow 1$

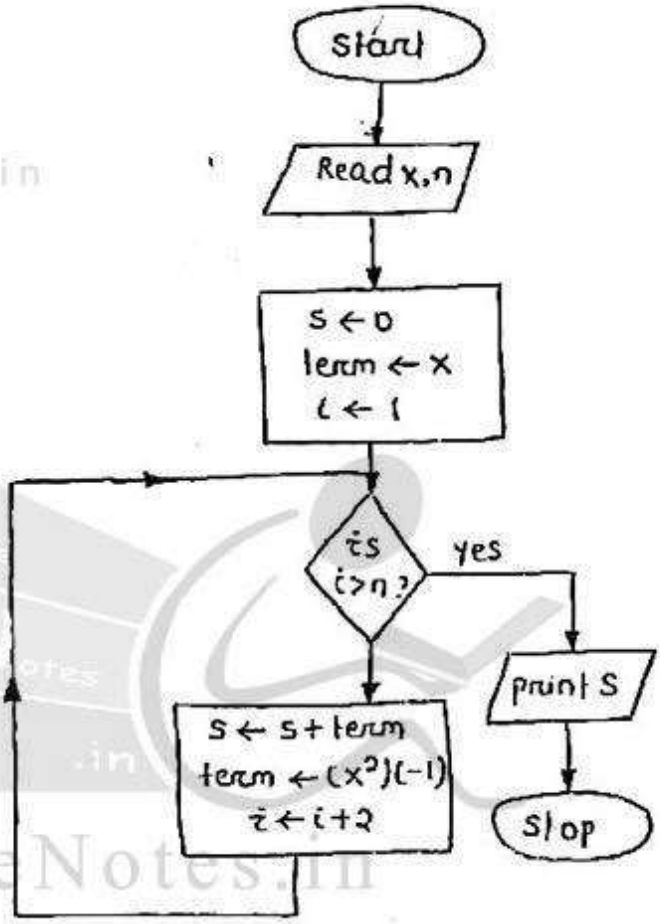
Step3 : if $i > n$ then print S.

Step4 : else $S \leftarrow S + term$
$term \leftarrow (x^2)(-1)$
$i \leftarrow i+2$

Step5 : goto step 3

Step6 : stop.

Flowchart -



Example :

write an algorithm and draw the flowchart to find the factorial of a given integer.

Algorithm :

Step 1 : Read n

Step 2 : fact ← 1

Step 3 : for i ← 1 to n step 1

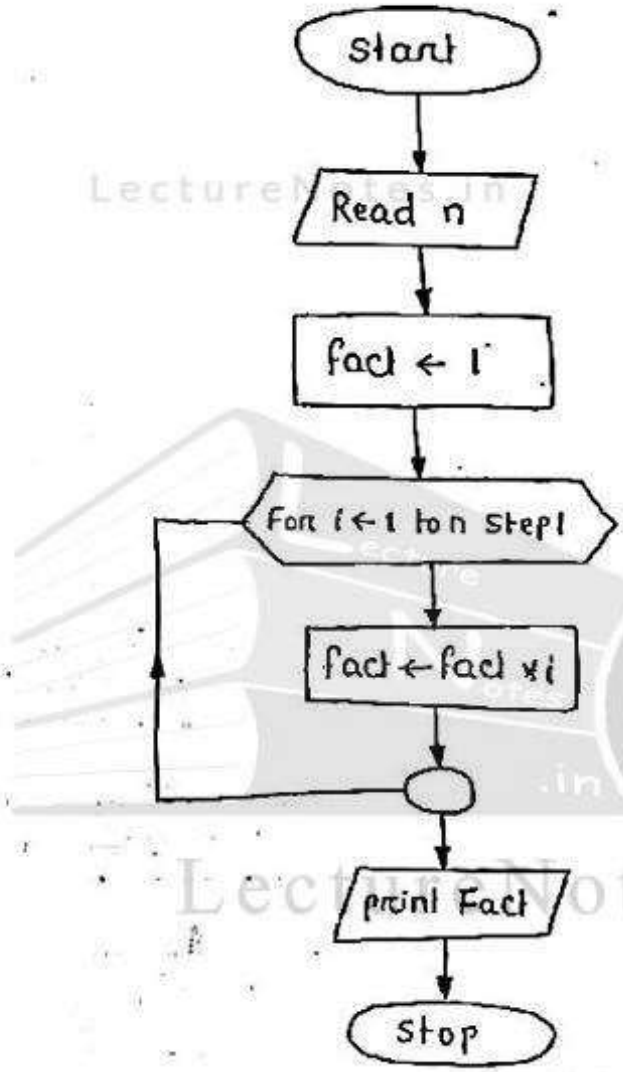
Step 4 : fact ← fact * i

Step 5 : goto step 3

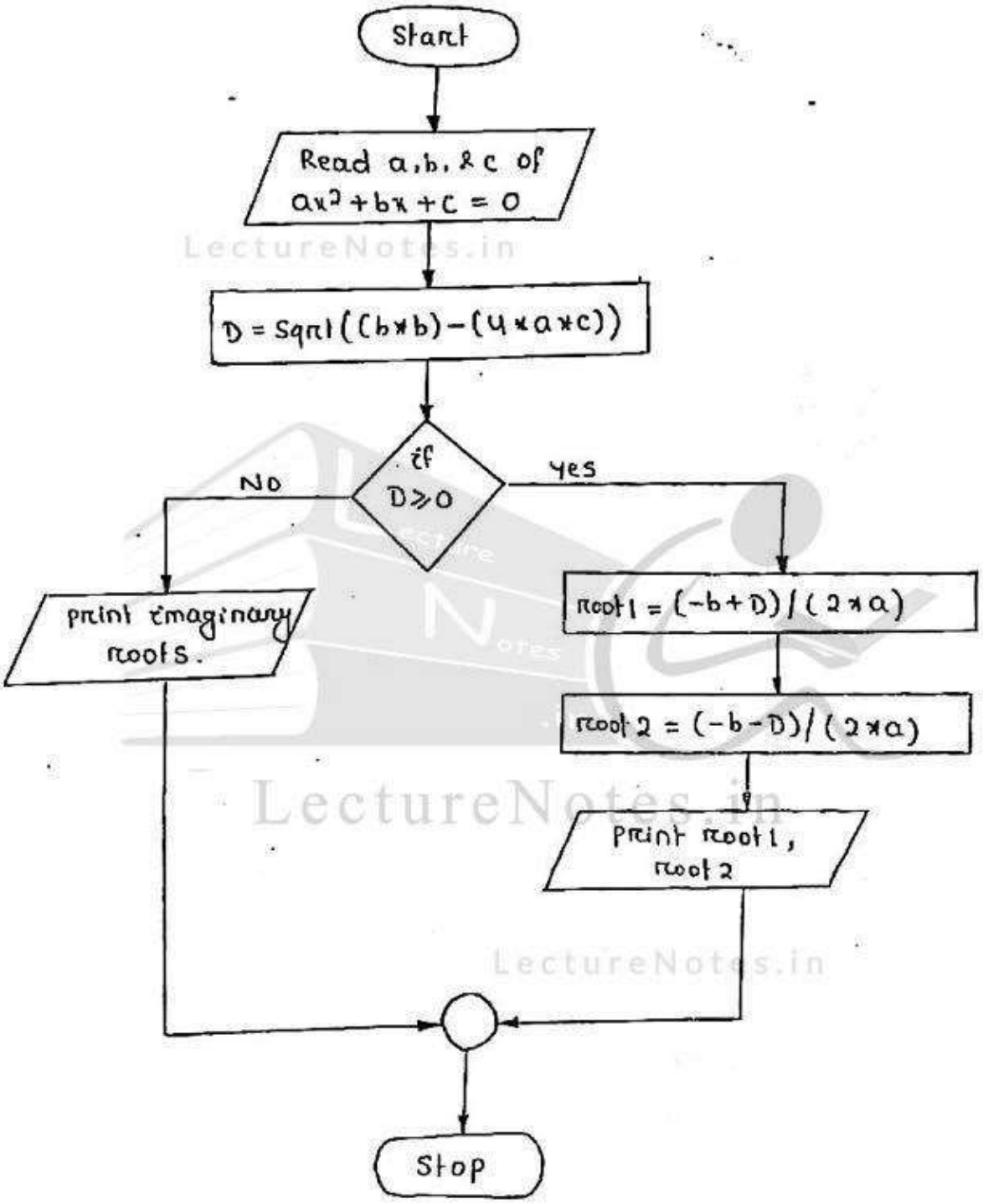
Step 6 : print fact

Step 7 : Stop

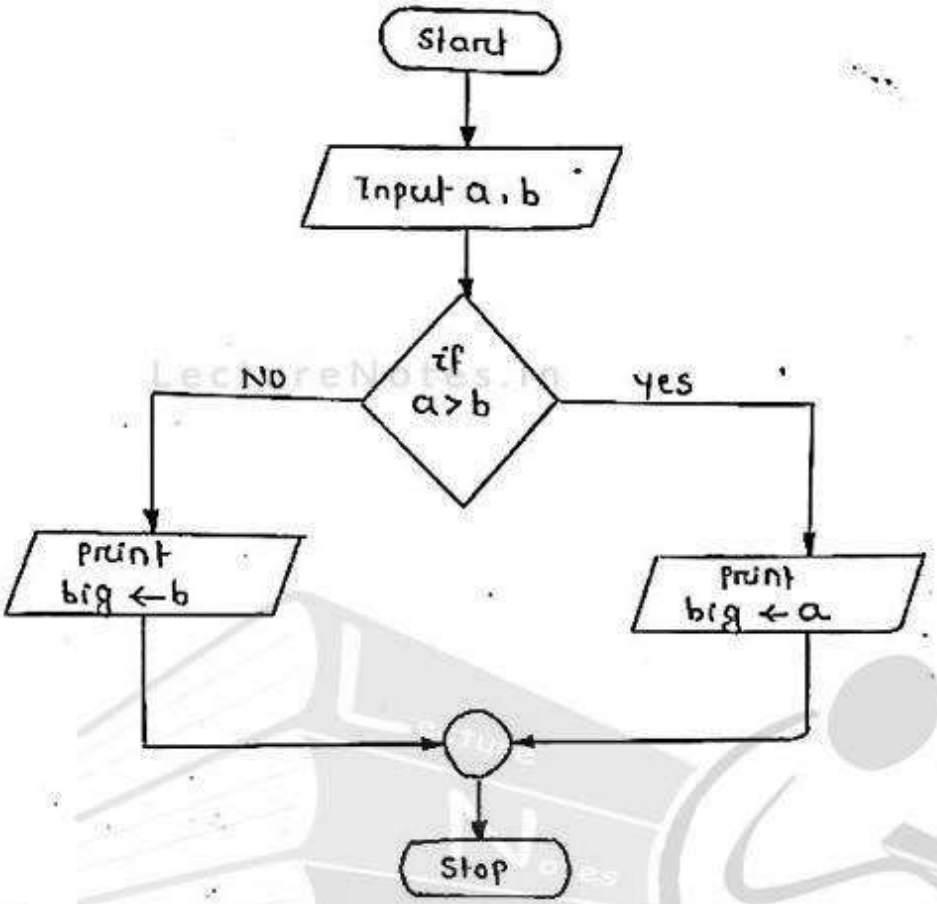
Flowchart :



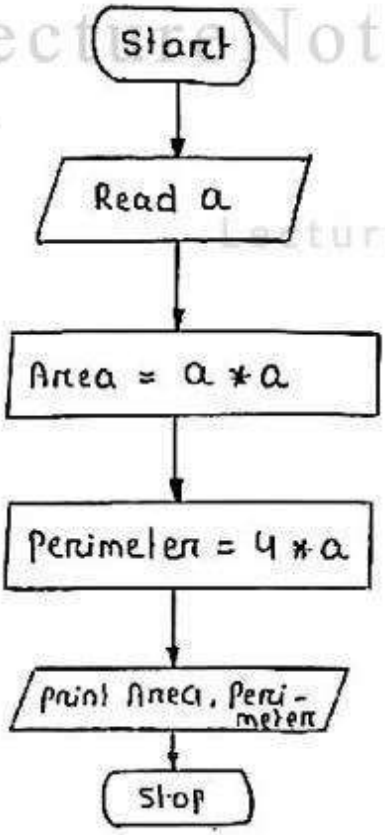
Assignment : Draw the flowchart to find the roots of a quadratic equation.



Assignment : Draw the flowchart to find the greatest between two numbers.



Assignment : Draw the flowchart to find the area and perimeter of a square



Assignment : write an algorithm and draw the flowchart to solve the following series (sinx)

$$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + \frac{x^n}{n!}$$ omitting those terms which are less than 10^{-5} in magnitude.

Algorithm :

Step 1 : Read x

Step 2 : $S \leftarrow 0$
$term \leftarrow x$
$i \leftarrow 1$

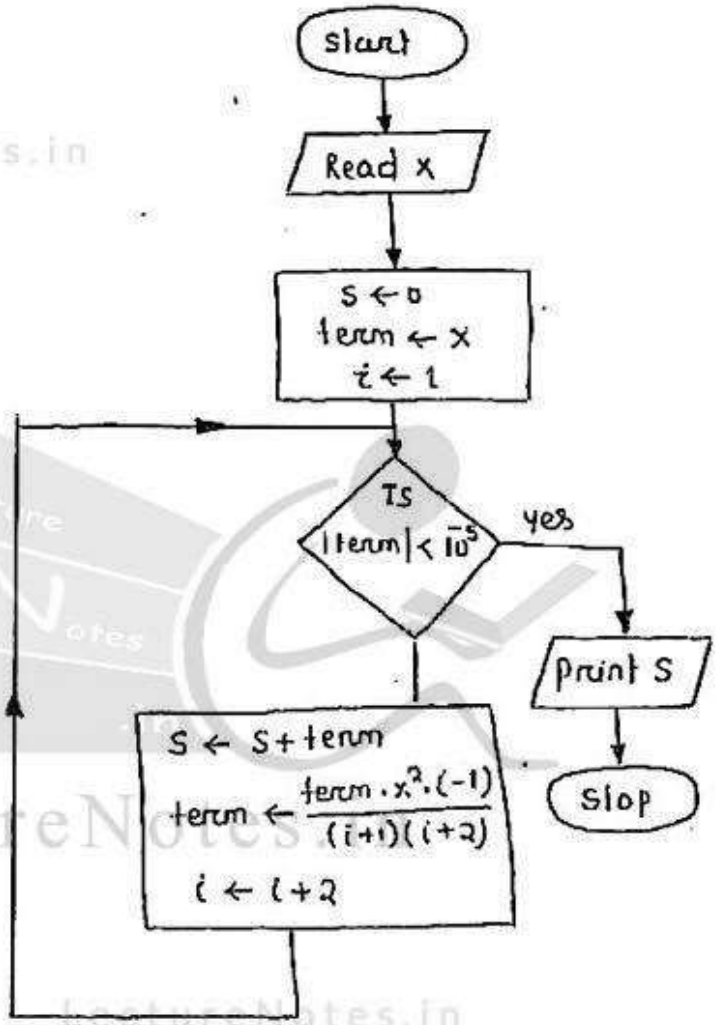
Step 3 : Is $|term| < 10^{-5}$
then print S.

Step 4 : else
$S \leftarrow S + term$
$term \leftarrow \frac{term \cdot x \cdot (-1)}{(i+1)(i+2)}$
$i \leftarrow i+2$

Step 5 : goto Step 3

Step 6 : stop

Flowchart



Assignment : write an algorithm and draw the flowchart to generate and print the fibonaci series 0 1 1 2 3 5 8 .....

Algorithm :

Step 1 : Read n

Step 2 : $P_1 \leftarrow 0$
$P_2 \leftarrow 1$

Step 3 : print $P_1 \cdot P_2$

Step 4 : newterm $\leftarrow P_1 + P_2$

Step 5 : while (newterm <= n)

Step 6 : print newterm.

Step 7 : $P_1 \leftarrow P_2$

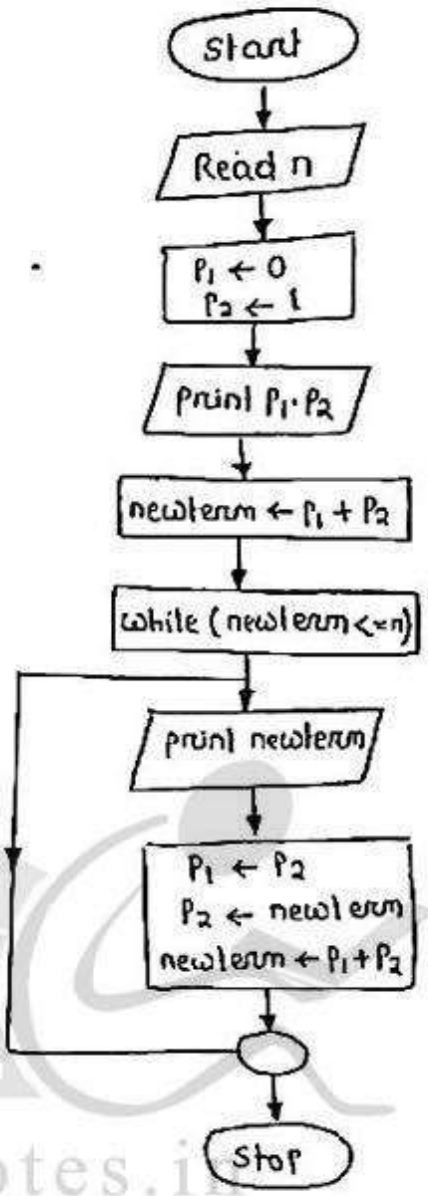
$P_2 \leftarrow$ newterm

newterm $\leftarrow P_1 + P_2$

Step 8 : goto Step 6

Step 9 : Stop.

Flowchart



Assignment :

Given three sides of a triangle.
Write an algorithm and draw the flowchart to determine whether a triangle can be formed using these three sides.

Algorithm:

Step 1 : Input A, B, C

Step 2 : If A + B <= C, print "A triangle cannot be formed", Otherwise go to the next step.

Step 3 : If B + C <= A, print "A triangle cannot be formed", Otherwise go to the next step.

Step 4 : If C+A <= B, print "A triangle cannot be formed". otherwise go to the next step.

Step 5 : print "A triangle can be formed".

Step 6 : End.

Flowchart :