Chapter Three

Flowcharts and Algorithms

(1) Introduction

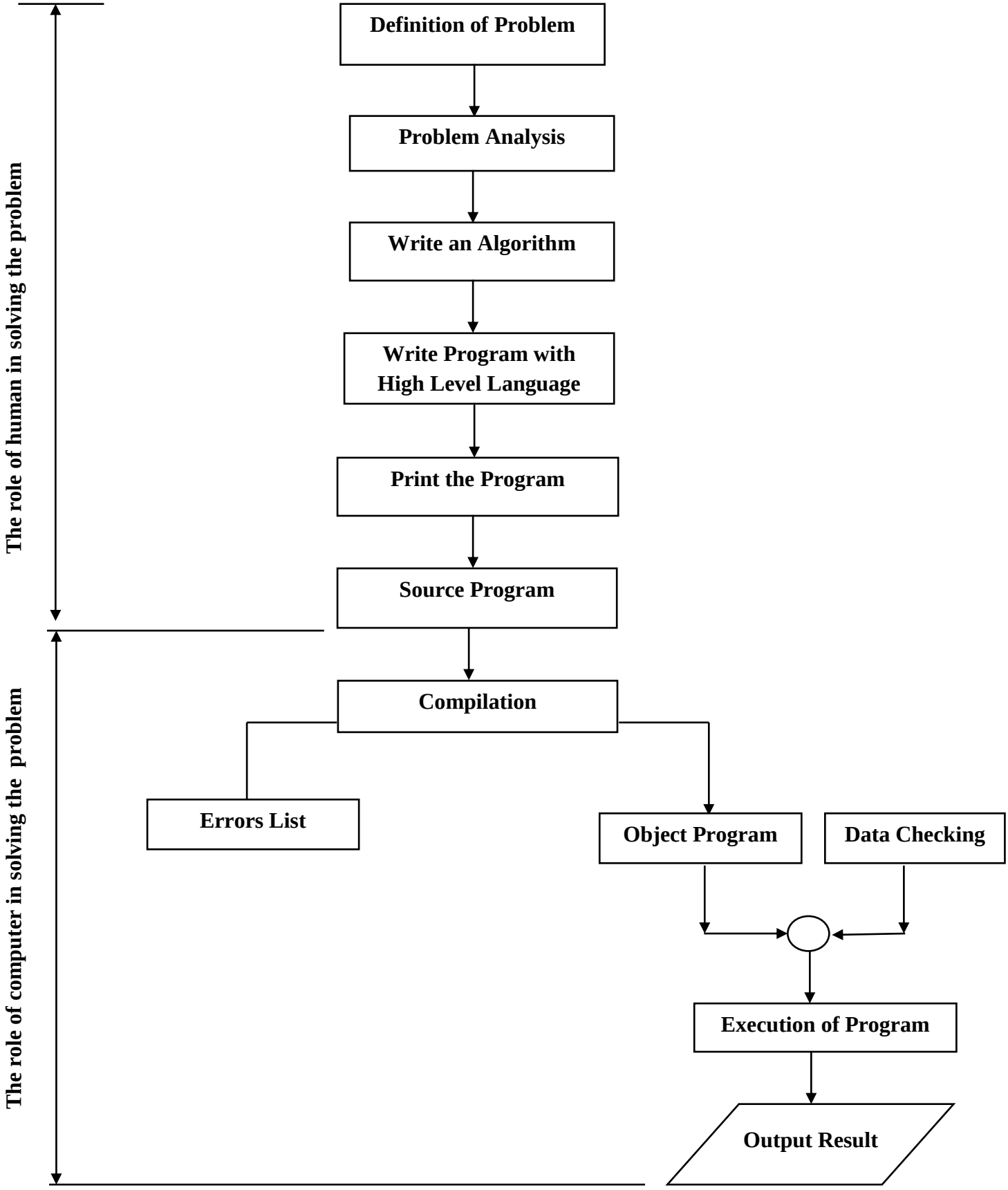
The algorithms are the hardest part in programming and when you learn it correctly it means that you are able to learn any programming language you want. In fact if you do not know how to deal with the code and address all the problems you face while writing the code you will give up and stop in your place when the first trivial problem appears to you in the code and you have thus wasted your time because you do not know how to solve the problem you are dealing with. So when you have finished analyzing your problem you can write it in any programming language you want with easy.

In fact, if you really understand the algorithm, you will be able to understand any programming language you want because you will be able to control and manipulate the code the way you want.

Algorithms are not programming language, but are the method of analysis and thinking that you have to follow in order to write the code correctly and therefore the algorithm can be defined as a set of steps or instruction arranged to perform a calculation or logical process sequentially and systematically.

There are two way to represented the algorithm (Pseudo code and flowchart) as it will be explained later.

In order to resolve any problem by using any high level language, there are two main stages as shown in the following form.


The role of human in solving the problem
The role of computer in solving the problem
Definition of Problem
Problem Analysis
Write an Algorithm
Write Program with High Level Language
Print the Program
Source Program
Compilation
Errors List
Object Program
Data Checking
Execution of Program
Output Result

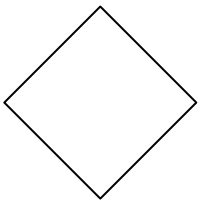
(2) Symbols and Idiomatic Forms in Flowcharts

(1) Start/ Stop

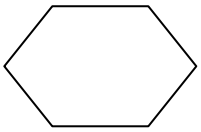
(2) Input/output

(3) Calculation and Store

(4) Decision

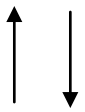


(5) Looping

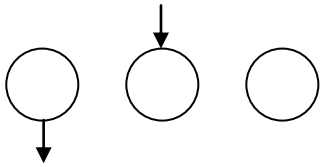


(6) Call Subroutine

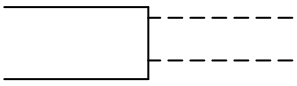
(7) Flow Line



(8) Connector



(9) Comment



(3) Algorithms (Pseudo Code)

A formula or set of steps for solving a problem. To be an algorithm, a set of rules must be unambiguous and have a clear stopping point". There may be more than one way to solve a problem, so there may be more than one algorithm for a problem. On other words, "A sequence of activities to be processed for getting desired output from a given input." Then we can say that:

1. Getting specified output is essential after algorithm is executed.
2. One will get output only if algorithm stops after finite time.
3. Activities in an algorithm to be clearly defined in other words for it to be unambiguous. Before writing an algorithm for a problem, one should find out:

- what is/are the inputs to the algorithm
- what is/are expected output after running the algorithm

The following symbol for different operations should be used while solving some problems: '+' for Addition, '-' for Subtraction, '*' for Multiplication, '/' for Division and '= ' for assignment.

For example, A =X*3 means A will have a value

of X*3.

Examples of Algorithm:

Problem 1 Find the area of a Circle of radius r.

Inputs: Radius r of the Circle and PI=3.14.

Expected output: Area of the Circle

Algorithm:

Step1: start

Step2: Read\input the Radius r of the Circle

Step3: Input PI=3.14

Step4: Calculate Area =PI*r*r

Step4: Print Area

Step5: End

Problem 2: Write an algorithm to read two numbers and find their sum.
Inputs: First num1. Second num2.
Expected output: Sum of the two numbers.

Algorithm:

Step1: Start
Step2: Read\input the first num1.
Step3: Read\input the second num2.
Step4: Calculate Sum =num1+num2
Step5: Print Sum
Step6: End

Problem 3: Write an algorithm to find the average of temperature T1, T2, and T3.
Inputs: The temperature T1, T2, And T3.
Expected output: The average of three temperature.
Step1: Start
Step2: Read the values f temperature T1, T2, and T3..
Step3: Calculate the average=(T1+T2+T3)/3
Step4: Print the average
Step5: End

Type of Algorithms

There are three types of control structure Algorithms:

1. Sequence
2. Branching (Selection)
3. Loop (Repetition)

The sequence is exemplified by sequence of statements place one after the other – the one above or before another gets executed first.

The branch refers to a binary decision based on some condition. If the condition is true, one of the two branches is explored; if the condition is false, the other alternative is taken. This is usually represented by the 'if-then' construct in pseudo-codes and programs.

Problem 4: write algorithm to find the greater number between two numbers
Step1: Start
Step2: Read/input A and B
Step3: If A greater than B then C=A
Step4: if B greater than A then C=B
Step5: Print C
Step6: End

The loop allows a statement or a sequence of statements to be repeatedly executed based on some loop condition. It is represented by the 'while' and 'for' constructs in most programming languages, for unbounded loops and bounded loops respectively.

A trip around the loop is known as iteration. You must ensure that the condition for the termination of the looping must be satisfied after some finite number of iterations, otherwise it ends up as an infinite loop, a common mistake made by inexperienced programmers.

Problem 5: Write An algorithm to calculate even numbers between 0 and 99

1. Start
2. I = 0
3. Print I in standard output
4. I = I+2
5. If (I <=98) then go to line 3
6. End

Note:

(Unbounded loops refer to those whose number of iterations depends on the eventuality that the termination condition is satisfied; bounded loops refer to those whose number of iterations is known before-hand.)

(4) Importance of Using Flowcharts

One of the most important benefits of using the flowcharts before writing any program is the following things:

1- Gives an integrated image of the steps required to solve the problem in the mind of the programmer so that it can fully inform all parts of the problem from the beginning to the end.

2- Help the programmer to diagnose the errors that occur in the program, especially the logical mistakes whose discovery depends on the logical sequence of steps to solve the problem.

3- Makes it easier for the programmer to make any adjustments in any part of the problem quickly and without having to re-examine the problem.

4- Makes it easier for the programmer to follow the parts of the program that have a lot of possibilities and branched.

5- The flowchart used in the design of solutions to some problems are a reference in resolving other similar problems and a key to resolving new problems related to old problems resolved.

(5) Flowcharts Types

In general, it can be said that there are two main types of flowcharts:

(a) System Flowcharts

This type of flowcharts is used when designing engineering devices in factories and others, which use self-control systems such as buoys in water tank, traffic lights, and pressure control devices and temperatures in oil distillation towers so that the flowcharts here are considered as the complete blueprint that shows the order, relationship and function of each stage before and after, within the framework of the integrated system.

The role of these flowcharts can be summarized as follow:

1- It is easy to detect any malfunction in the whole system just by looking at. This is led to simply maintenance process and at the lowest cost.

2- Facilitates future modification to the system program at any location.

3- Explain the details of the data to be entered into the system.

4- Explain the details of the types of expected or required results from the program prepared for the system.

5- To indicated how to link the system to the rest of system in the designated organization.

(b) Program Flowcharts

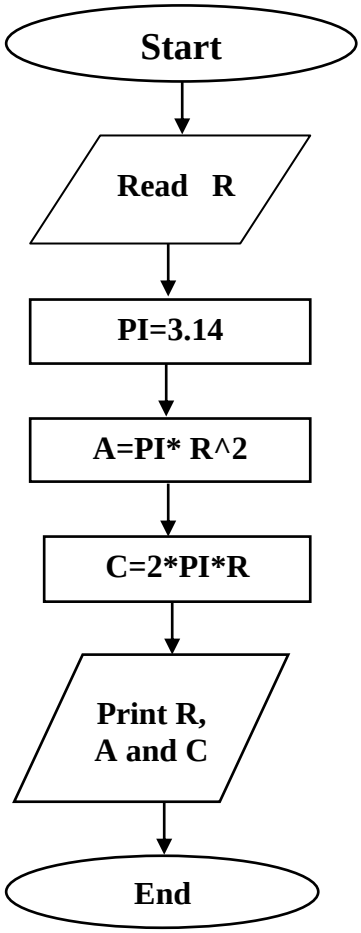
This type of flowcharts is used to illustrate the main steps that are being developed to resolve a problem in the form of idiomatic drawings showing the logical relationship between the other steps of the solution and the location and function of each of them within the framework of a comprehensive solution to the problem.

There are four types from program flowcharts :
1.Simple Sequential Flowcharts
2.Branched Flowcharts
3.Simple-Loop Flowcharts
4. Multi-Loop Flowcharts

(1) Simple Sequential Flowcharts

The solution steps for this type of flowchart are arranged in a straight series from the beginning of the program to the end so that it is free of branches and loops.

Example 1: Draw a flowchart to find the area and perimeter of circle have known radius (R).

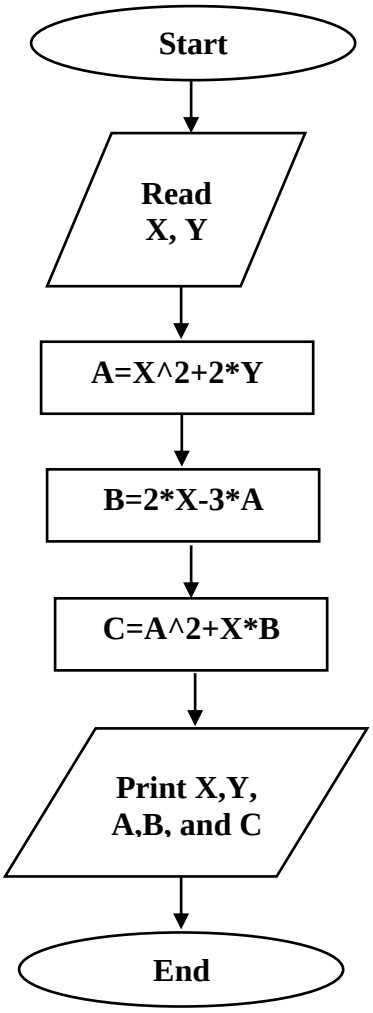


Example 2: Draw a flowchart to find the values of variables A, B, and C from the following equations.

1. $A=X^2+2Y$

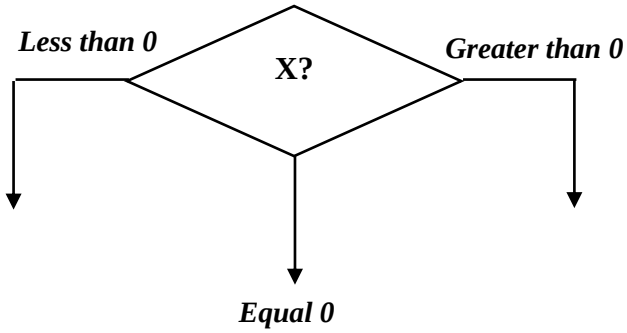
2. $B=2X-3A$

3. $C=A^2+XB$

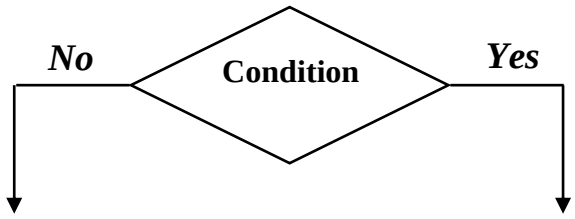


(2) Branched Flowcharts

This type of flowcharts are used when there is a need to make a decision or a trade-off between two or more choices. There are two types of resolution as shown below:



Three branches decision



Two branches decision

In general, the branch flowcharts can take one of the following two images:

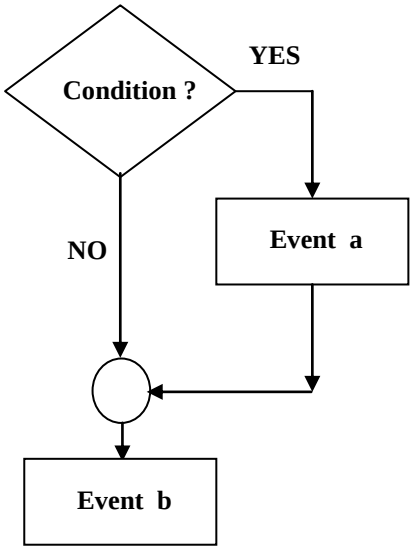


Fig.(2)

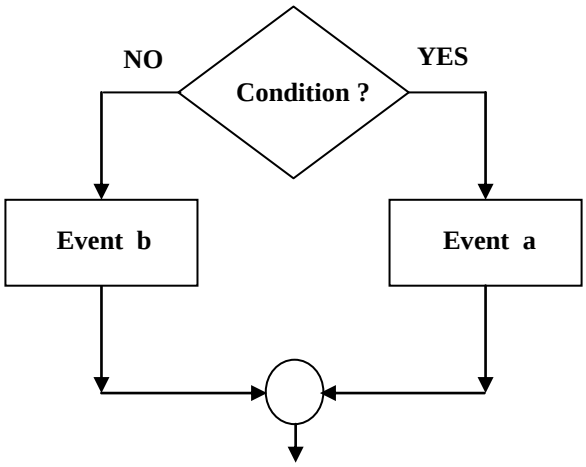


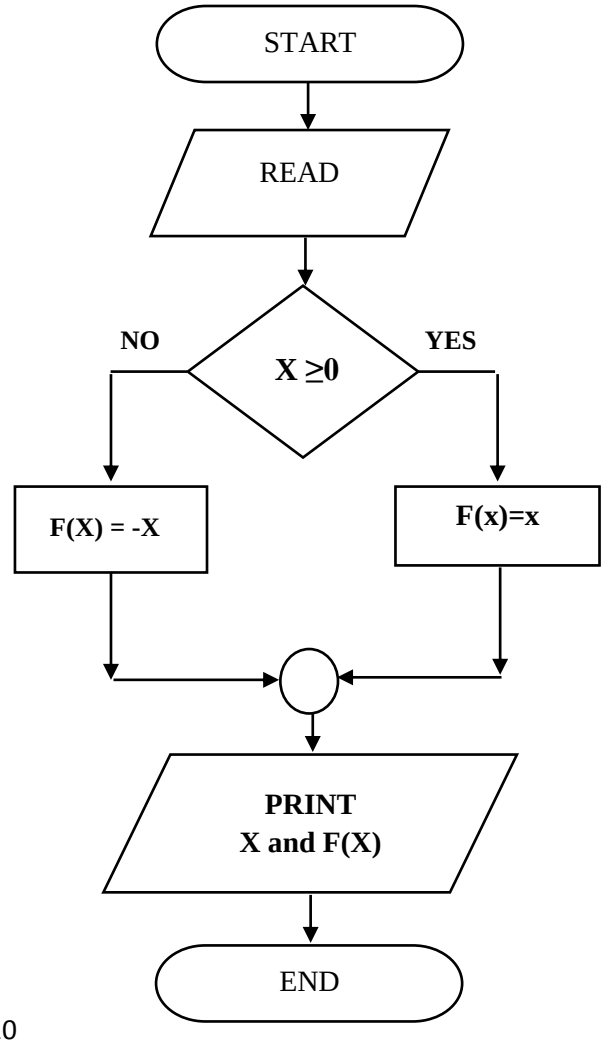
Fig.(1)

Fig.(1) shows that if the answer of condition is (Yes) the next event in the execution is event (a), but if the answer is (No), the next event is event (b) either in Fig.(2) if the answer of condition is (Yes), the next event in the execution is event (a) and then followed by event (b), but if the answer of condition is (No), the next event will be event (b) directly.

Example (3):

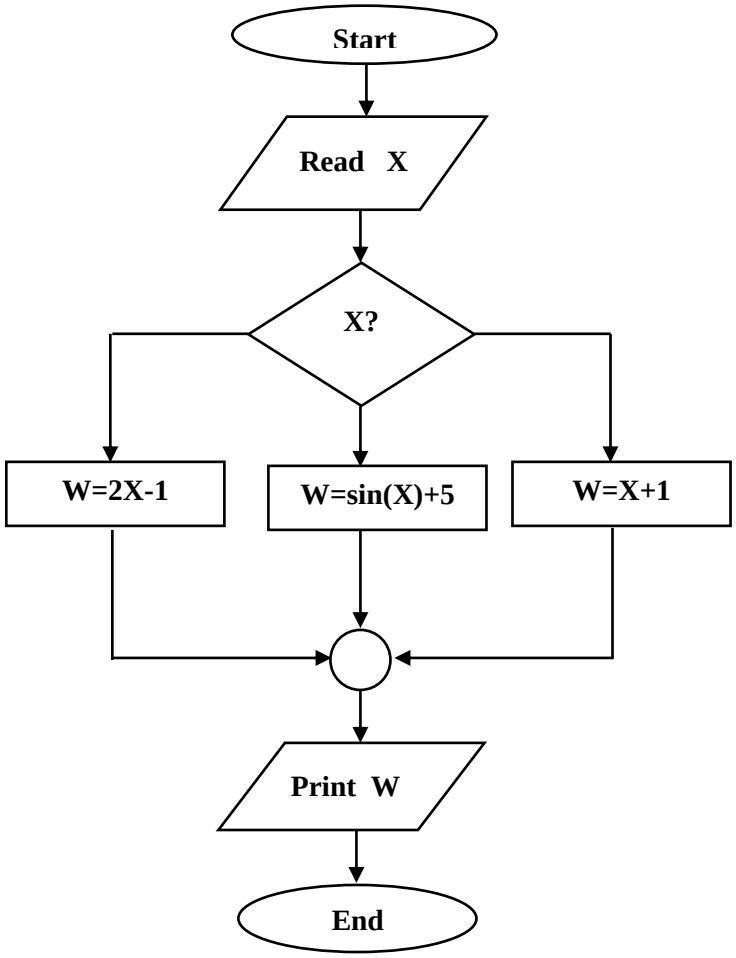
Draw a flowchart to find the F(x) ?

$$F(x) = \begin{cases} x & x \geq 0 \\ x & x < 0 \end{cases}$$



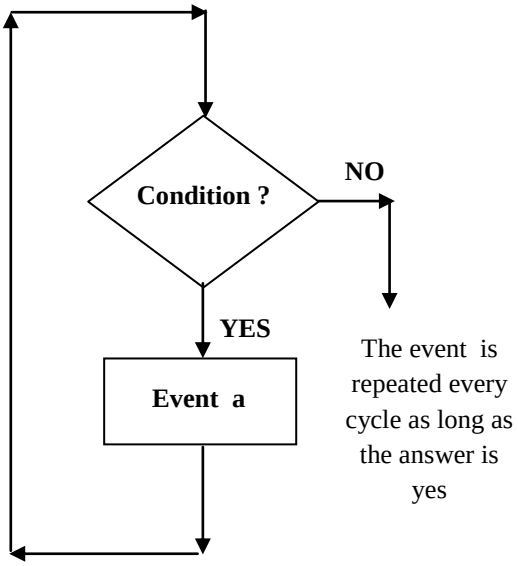
Example (4): Draw a flowchart to find the value of W from the following equations, note that the value of variable X is known ?

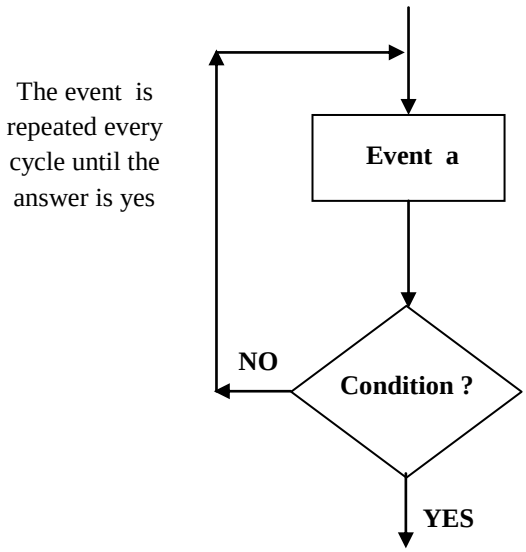
$$W = \begin{cases} X + 1 & X > 0 \\ \sin(X) + 5 & X = 0 \\ 2X - 1 & X < 0 \end{cases}$$



(3) Simple-Loop Flowcharts

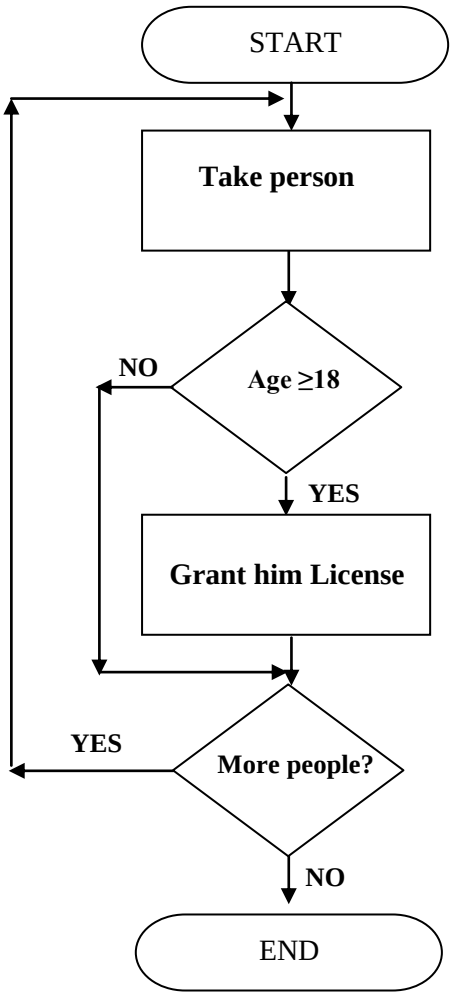
This type of flowcharts we need to return a process or set of processes in the program a specific or unspecified number of times, and the overall form of these flowcharts is as follows:





And this flowchart is called a single-loop flowcharts because it uses a single rotation loop and is sometimes called a simple-loop flowcharts.

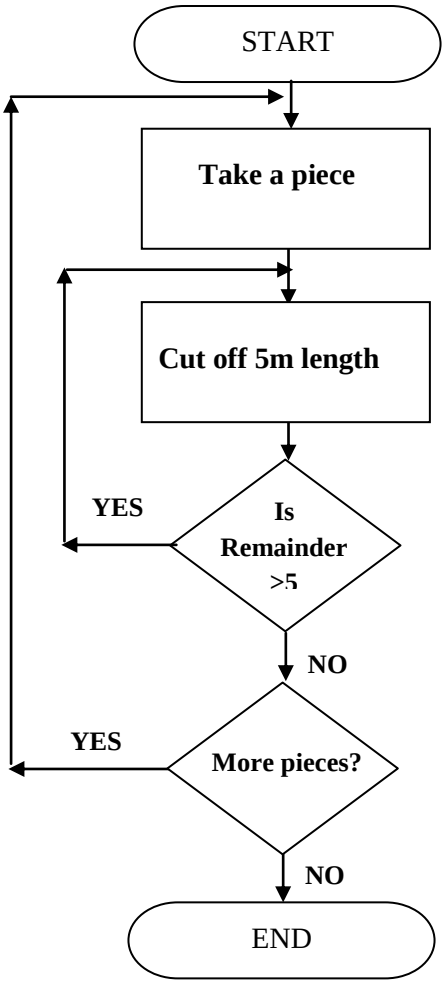
Example (5): The general Traffic Directorate in Iraq is submitting a project to obtain online driving permits, provided that the applicant not under 18 years of age . Draw the flowchart for this project?



(4) Multi-Loop Flowcharts

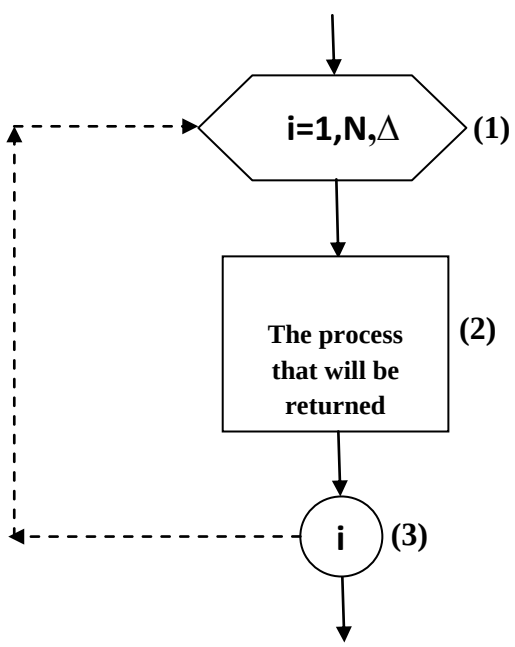
In this type of flowcharts the rotations are within each other so that do not intersect.

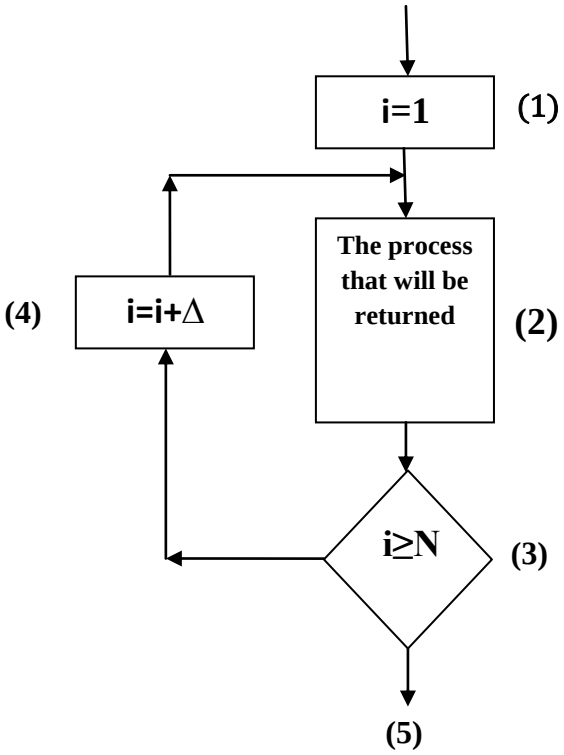
Example (6): A merchant wants to cut off a set of cloth sized more than 5 meters long into small pieces of which are 5 meters long . Draw the flowchart for this project ?



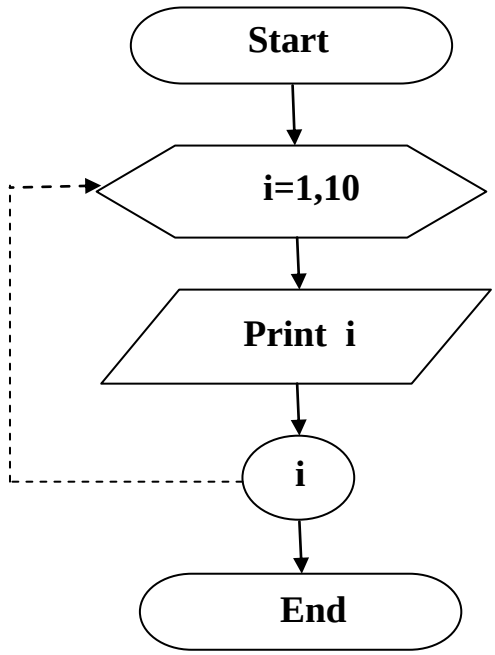
In this example we observed the presence of two loops one internal and called internal loop and the other is external loop and the functioning of these two loops are coordinated so that the priority of implementation is for internal loop.

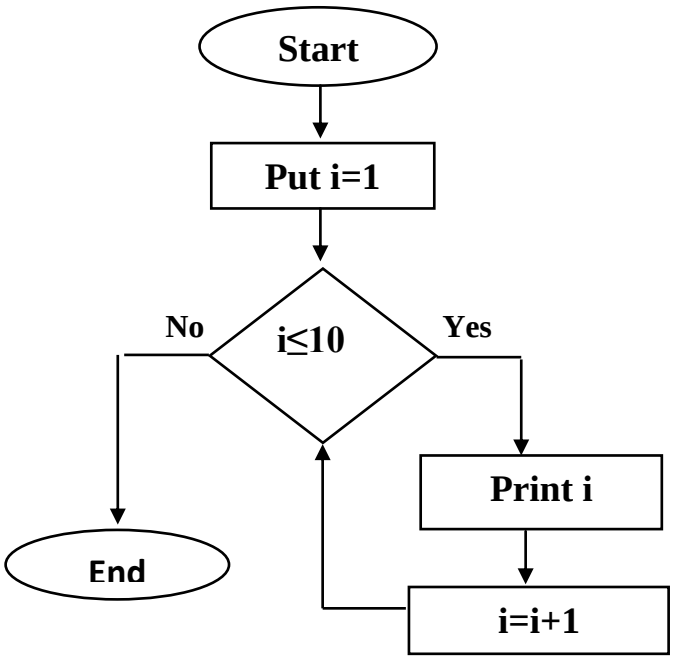
The Idiomatic Form For Counters





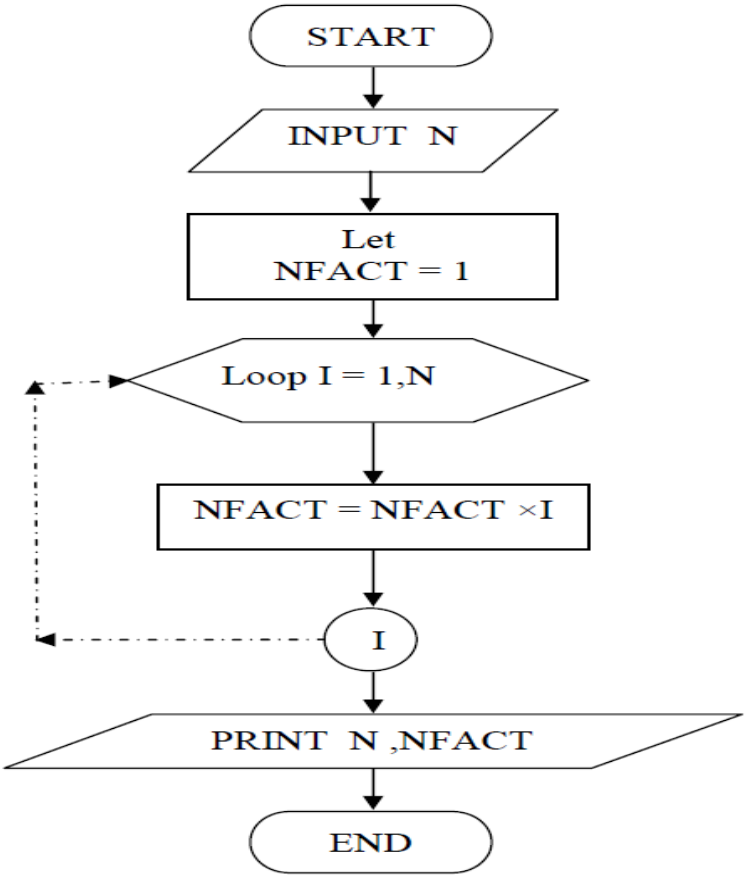
Example (7): Draw a flowchart for program print the first ten natural numbers ?





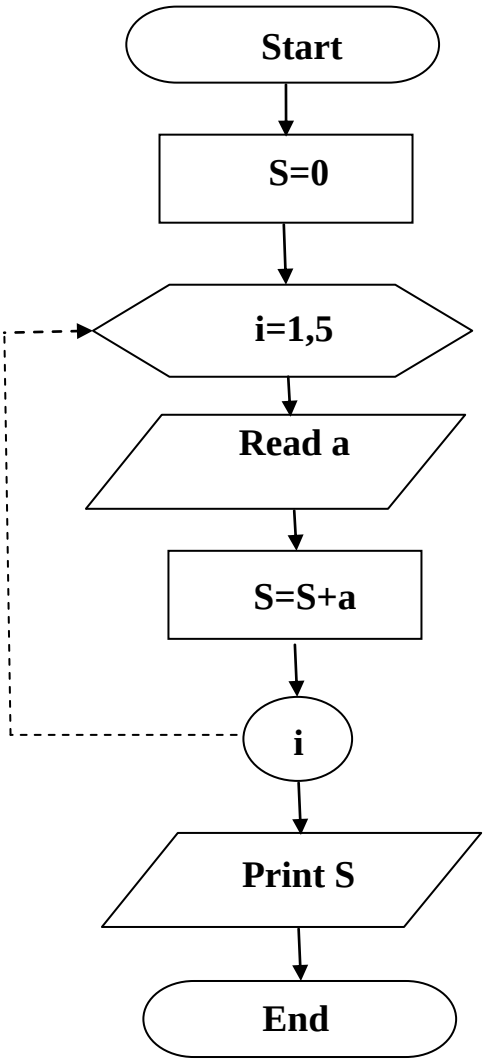
Example (8): Draw a flowchart to read the number N and find its factorial ?

N!=N(N-1)(N-2)(N-3)…….



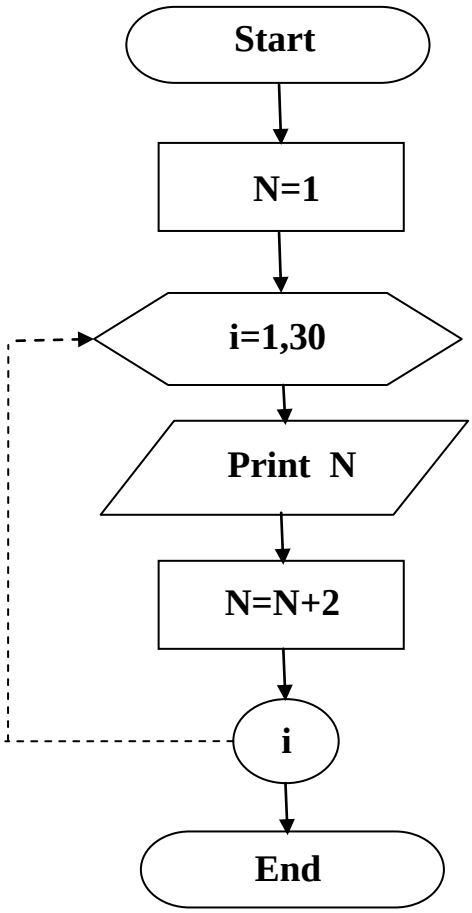
Example (9): Draw a flowchart to find the sum of the limits in the following amount ?

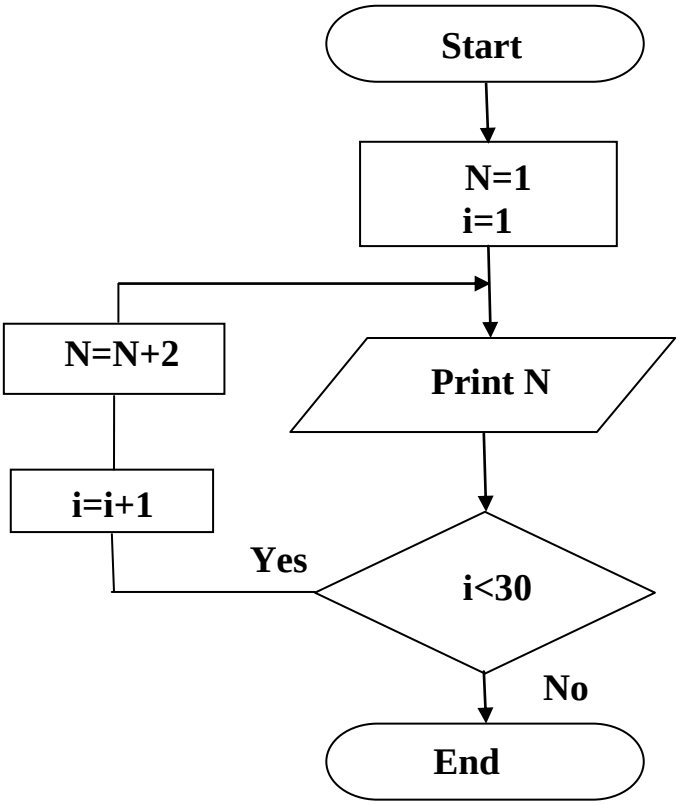
$$S = \sum_{i=1}^{5} a_i$$



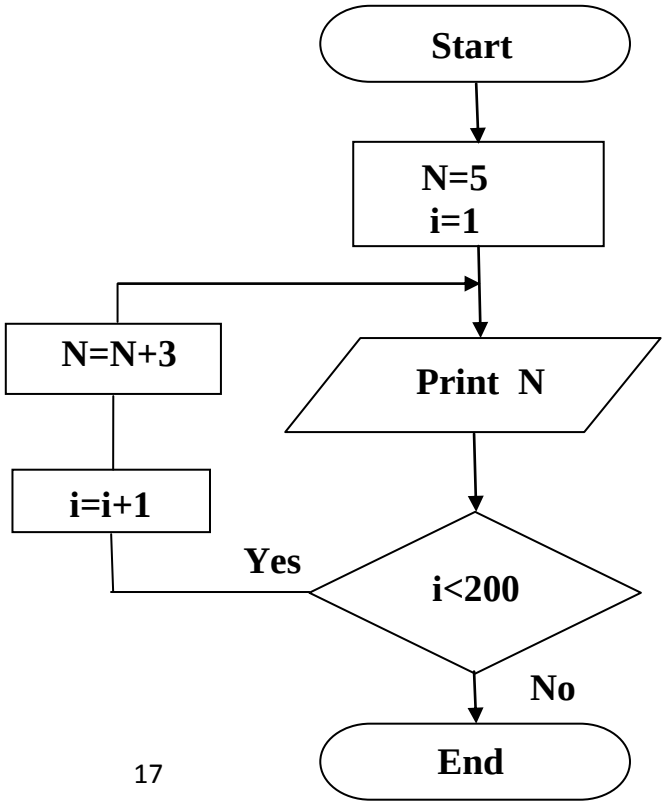
Example (10): Draw a flowchart to print the first thirty limits of the following series ?

1, 3, 5, …….





Example (11): Draw a flowchart to print the first two hundred limit of the engineering series that it start with 5 so that the change rate is 3 ?



Exercises:

Q1: Draw a flowchart for work the traffic lights ?

Q2: Write the algorithm required to find the largest number from the numbers A, B, and C ?

Q3: Write the algorithm for a program that calculates the cost of the phone bill according to the following three rules ?

(1) The first 50 minute: each minute with 10 DI.

(2) The next 100 minute: each minute with 15 DI.

(3) Whats more: each minute with 20 DI.

Q4: Write the algorithm to find the sum of the number from 1 to 20 ?

Q5: Draw a flowchart for the following algorithm ?

(1) Put value of sum equal zero and value of N equal 1.

(2) Add N to the sum

(3) If N<6 add 1 to the current value of N, then go to step 2.

(4) Print value of sum.

(5) Stop

Q6: Draw a flowchart to find the sum m of real numbers (X_1, X_2, X_3,……X_m) .

So that the result is the sum of the numbers T such that $= \sum_{i=1}^{m} X_i$?

Q7: Write an algorithm to print the multiplication table for number 9?

Q8: Write an algorithm to calculate the average numbers that the factorial result of last number equal to 5040?

Q9: Draw a flowchart to find the greater common factor between two numbers?

Q10: Draw a flowchart to find the prime numbers for N numbers?