Flowchart

A flow chart is a step-by-step diagrammatic representation of the logic paths to solve a given problem. A flowchart is **visual or graphical representation of an algorithm**. The flowcharts are pictorial representation of the methods to be used to solve a given problem and help a great deal to analyze the problem and plan its solution in a systematic and orderly manner. A flowchart when translated into a proper computer language, results in a complete program.

Advantages of Flowcharts

1. The flowchart shows the logic of a problem displayed in pictorial fashion which felicitates easier checking of an algorithm.

2. The Flowchart is good means of communication to other users. It is also a compact means of recording an algorithm solution to a problem.

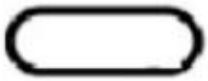
3. The flowchart allows the problem solver to break the problem into parts. These parts can be connected to make master chart.

4. The flowchart is a permanent record of the solution which can be consulted at a later time.

Differences between Algorithm and Flowchart

Algorithm	**Flowchart**
1. A method of representing the step-by-step logical procedure for solving a problem .	1. Flowchart is diagrammatic representation of an algorithm. It is constructed using different types of boxes and symbols.
2. It contains step-by-step English descriptions, each step representing a particular operation leading to solution of problem .	2. The flowchart employs a series of blocks and arrows, each of which represents a particular step in an algorithm
3. These are particularly useful for small problems.	3. These are useful for detailed representations of complicated programs
4. For complex programs, algorithms prove to be Inadequate.	4. For complex programs, Flowcharts prove to be adequate

Symbols used in Flow-Charts

Symbol	Symbol Name	Description
	Flow Lines	Used to connect symbols
	Terminal	Used to start, pause or halt in the program logic
	Input/output	Represents the information entering or leaving the system
	Processing	Represents arithmetic and logical instructions
	Decision	Represents a decision to be made
	Connector	Used to Join different flow lines
	Sub function	used to call function

The symbols that we make use while drawing flowcharts as given below are as per conventions followed by International Standard Organization (ISO).

a. Oval: Rectangle with rounded sides (Terminal) is used to indicate either **START/ STOP** of the program.

b. Input and output indicators: Parallelograms are used to represent input and output operations. Statements like INPUT, READ and PRINT are represented in these Parallelograms.

c. Processing: - Rectangle is used to indicate any set of processing operation such as for storing arithmetic operations.

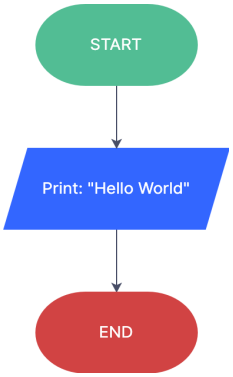
d. Decision Makers: The **diamond** is used for indicating the step of **decision making** and therefore known as decision box. Decision boxes are used to test the conditions or ask questions and depending upon the answers, the appropriate actions are taken by the computer.

e. Flow Lines: Flow lines indicate the direction being followed in the flowchart. In a Flowchart, every line must have an arrow on it to indicate the direction. The arrows may be in any direction

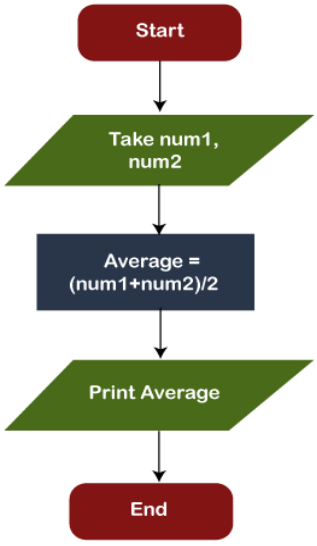
f. connectors: Circles are used to join the different parts of a flowchart and these circles are called on-page connectors. The uses of these connectors give a neat shape to the flowcharts. In a complicated problems, a flowchart may run in to several pages. The parts of the flowchart on different are to be joined with each other. The parts to be joined are indicated by the circle.

Examples:

Example1: Draw a flowchart to print a message



Example 2: Draw a flowchart to calculate the average of two numbers



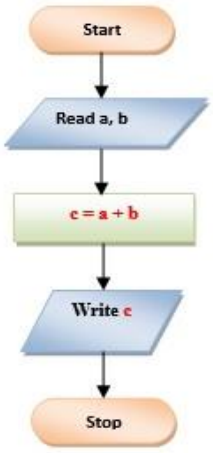
Example3

To find sum of two numbers

Algorithm

1. Start
2. Read a, b
3. c = a + b
4. Print or display c
5. Stop

Flowchart



Program

```c
#include<stdio.h>

int main()
{
    int a, b, c;

    printf("Enter value of a: ");
    scanf("%d", &a);

    printf("Enter value of b: ");
    scanf("%d", &b);
    c = a+b;

    printf("Sum of given two numbers is: %d", c);

return 0;
}
```

Example 4:

Finding Area of the square

Algorithm

1. Start
2. Read length, L
3. area = L*L
4. Print or display area
5. Stop

Flowchart

Program

```c
#include<stdio.h>

int main()
{

    int L, area;

    printf("Enter length of square L: ");
    scanf("%d", &L);

    area = L*L;

    printf("Area of square is: %d", area);

    return 0;
}
```

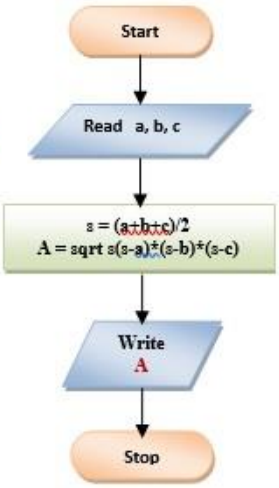
Example 5:

Area of a triangle where three sides are given

Algorithm

1. Start
2. Read a, b, c
3. s = (a+b+c)/2
4. A=sqrt (s *(s-a)*(s-b)*(s-c))
5. Print or display A
6. Stop

Flowchart



Program

```
#include<stdio.h>
#include<math.h>

int main()
{
    int a, b, c;
    float s, A;

    printf("Enter values of a, b, c: \n");
    scanf("%d %d %d", &a, &b, &c);

    s = (a+b+c)/2;
    A = sqrt (s*(s-a)*(s-b)*(s-c));

    printf("Area of triangle is: %f", A);

    return 0;
}
```

```
E:\WACHEMO\Sept 2016-Jan 2017\p
Enter values of a, b, c:
5
8
10
Area of triangle is: 14.071247
--------------------------------
```

Example 6:

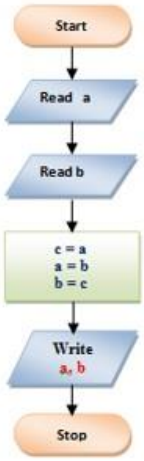
Swapping of 2 numbers using third variable

Interchange the value of two numbers

Algorithm

1. Start
2. Read two values into two variables a, b
3. Declare third variable, c
   - **c = a**
   - **a = b**
   - **b = c**
4. Print or display a, b
5. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{

    int a, b, c;
    printf("Enter value of a:");
    scanf("%d", &a);

    printf("Enter value of b:");
    scanf("%d", &b);

    c = a;
    a = b;
    b = c;

    printf("Values of a & b after swapping: ");
    printf("a = %d\n", a);
    printf("b = %d", b);

    return 0;
}
```

Example 7:

Calculate simple interest using the expression (SI=PNR/100)

Algorithm

1. Start
2. Read P, N, R
3. SI=(PNR)/100
4. Print SI
5. Stop

Flowchart

Program

```
#include<stdio.h>

int main()
{
    int P, N, R;
    float SI;

    printf("Enter values of P, N, R: ");
    scanf("%d %d %d", &P, &N, &R);

    SI = (P*N*R/100);
    printf("Simple Interest is: %f", SI);

    return 0;
}
```

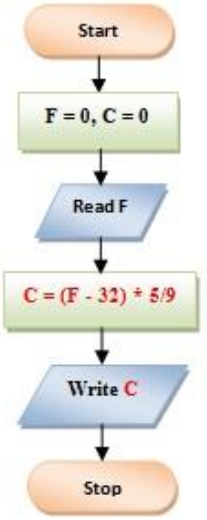
Example 8

Convert temperature from Fahrenheit to Celsius

Algorithm

1. Start
2. Initialize F = 0, C = 0
3. Read F
4. C = (F-32) * 5/9
5. Write C
6. Stop

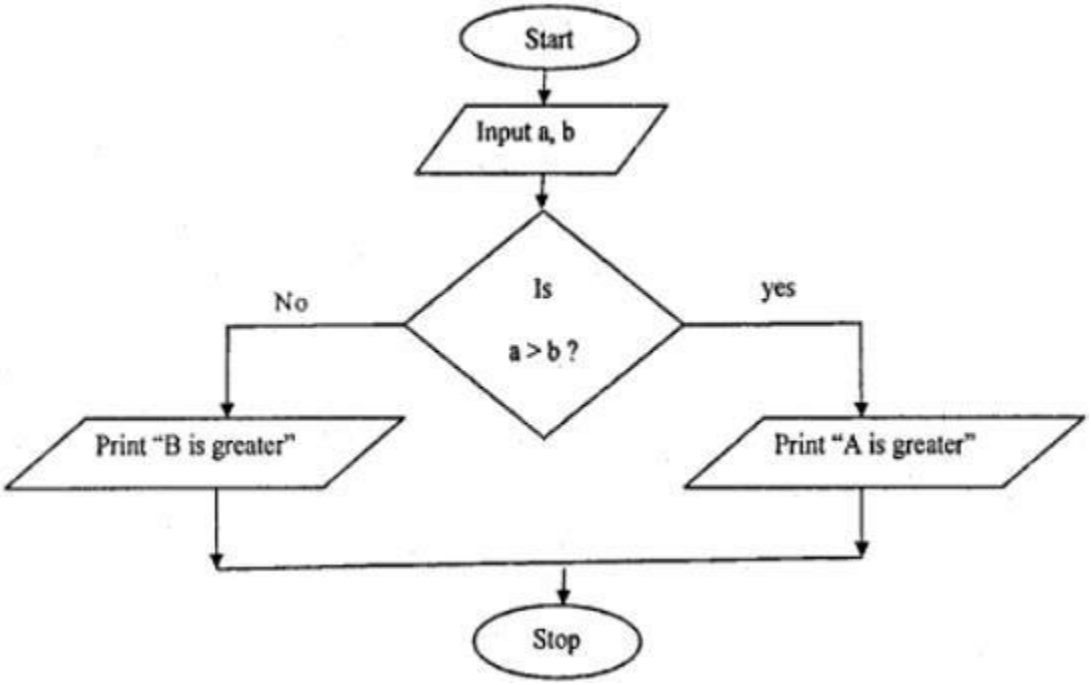
Flowchart



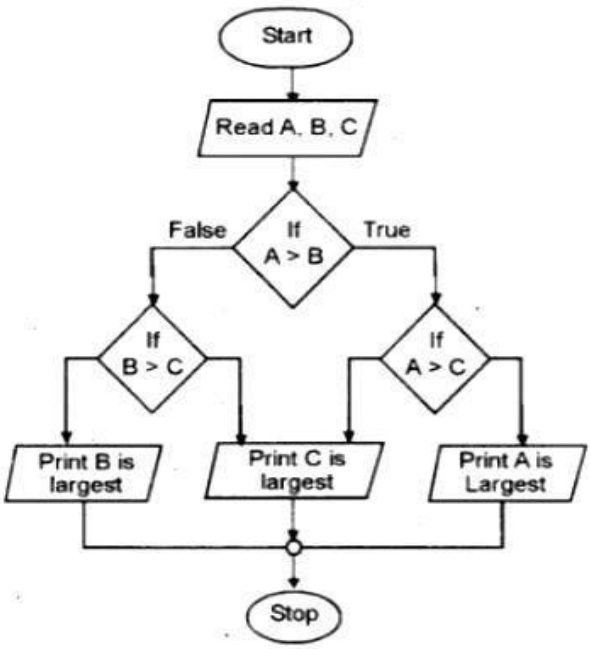
Program

```
#include<stdio.h>

int main()
{
    float F, C;

    printf("Enter Fahrenheit: ");
    scanf("%f", &F);

    C = (F-32)*5/9;
    printf("Temparature in Celsius is: %f", C);

    return 0;
}
```

Example 9: Draw a flowchart to find greater among two numbers



Example10: Draw a flowchart to find greatest among three numbers



Pseudo code

The Pseudo code is neither an algorithm nor a program. It is an abstract form of a program. It consists of English like statements which perform the specific operations. It is defined for an algorithm. It does not use any graphical representation. In pseudo code, the program is represented in terms of words and phrases, but the syntax of program is not strictly followed.

Pseudocode is text- based detail design tool. Pseudocode is an intermediary between an algorithm and implementation. Pseudocode is an outline of program that can be easily converted into programming statements.

Advantages:

* Easy to read,

* Easy to understand,

* Easy to modify.

Example: Write Pseudocode to add two numbers

Procedure Add(a,b)

// a and b are two integer variables

{

Sum:= a+b; return sum;

}