RV College of Engineering®

Go, change the world

UNIT-I
Algorithm and Flowchart

Algorithm

Definition

In programming, algorithm is a set of well defined instructions in sequence to solve the problem.

Qualities of a good algorithm

- Input and output should be defined precisely.
- Each steps in algorithm should be clear and unambiguous.
- Algorithm should be most effective among many different ways to solve a problem.
- An algorithm shouldn't have computer code. Instead, the algorithm should be written in such a way that, it can be used in similar programming languages.

Algorithm

Write an algorithm to add two numbers entered by user.

Step 1: Start

Step 2: Declare variables num1, num2 and sum.

Step 3: Read values num1 and num2.

Step 4: Add num1 and num2 and assign the result to sum.
sum←num1+num2

Step 5: Display sum

Step 6: Stop

Algorithm

Write an algorithm to find the largest among three different numbers entered by user. **Step 1**: Start

Step 2: Declare variables a,b and c.

Step 3: Read variables a,b and c.

Step 4: If a>b If a>c Display a is the largest number.

Else Display c is the largest number.

Else If b>c Display b is the largest number.

Else Display c is the greatest number.

Step 5: Stop

Advantages of Algorithm

- It is a step-wise representation of a solution to a given problem, which makes it easy to understand.
- An algorithm uses a definite procedure.
- It is not dependent on any programming language, so it is easy to understand for anyone even without programming knowledge.
- Every step in an algorithm has its own logical sequence so it is easy to debug.
- By using algorithm, the problem is broken down into smaller pieces or steps hence, it is easier for programmer to convert it into an actual program

Disadvantages of Algorithm

- Writing algorithm takes a long time.
- An Algorithm is not a computer program, it is rather a concept of how a program should be.

Flowchart

A **flowchart** is a type of diagram that represents an algorithm, workflow or process. The **flowchart** shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. …

Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Flowchart

Symbol	Name	Function
	Process	Indicates any type of internal operation inside the Processor or Memory
	input/output	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results
	Decision	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Connector	Allows the flowchart to be drawn without intersecting lines or without a reverse flow.
	Predefined Process	Used to invoke a subroutine or an Interrupt program.
	Terminal	Indicates the starting or ending of the program, process, or interrupt program
	Flow Lines	Shows direction of flow.

Advantages of Flowchart

- The Flowchart is an excellent way of communicating the logic of a program.
- It is easy and efficient to analyze problem using flowchart.
- During program development cycle, the flowchart plays the role of a guide or a blueprint. Which makes program development process easier.
- After successful development of a program, it needs continuous timely maintenance during the course of its operation. The flowchart makes program or system maintenance easier.
- It helps the programmer to write the program code.
- It is easy to convert the flowchart into any programming language code as it does not use any specific programming language concept.

Disadvantages of Flowchart

- The flowchart can be complex when the logic of a program is quite complicated.
- Drawing flowchart is a time-consuming task.
- Difficult to alter the flowchart. Sometimes, the designer needs to redraw the complete flowchart to change the logic of the flowchart or to alter the flowchart.
- Since it uses special sets of symbols for every action, it is quite a tedious task to develop a flowchart as it requires special tools to draw the necessary symbols.
- In the case of a complex flowchart, other programmers might have a difficult time understanding the logic and process of the flowchart.
- It is just a visualization of a program, it cannot function like an actual program

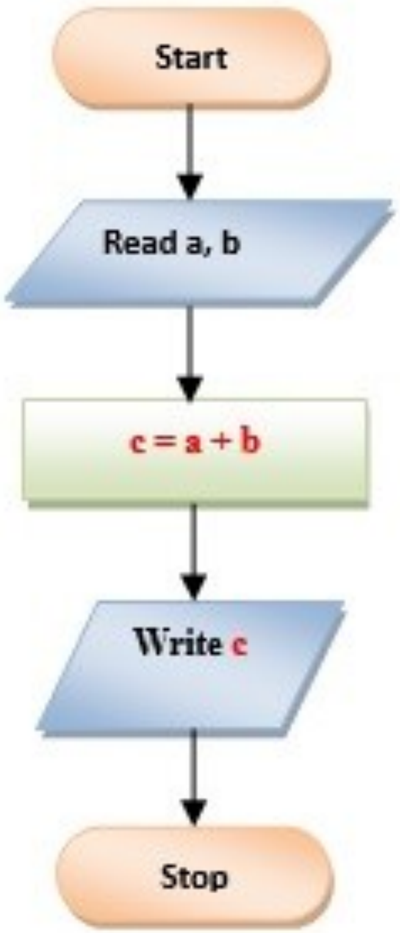
Example

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

```
#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;
}
```

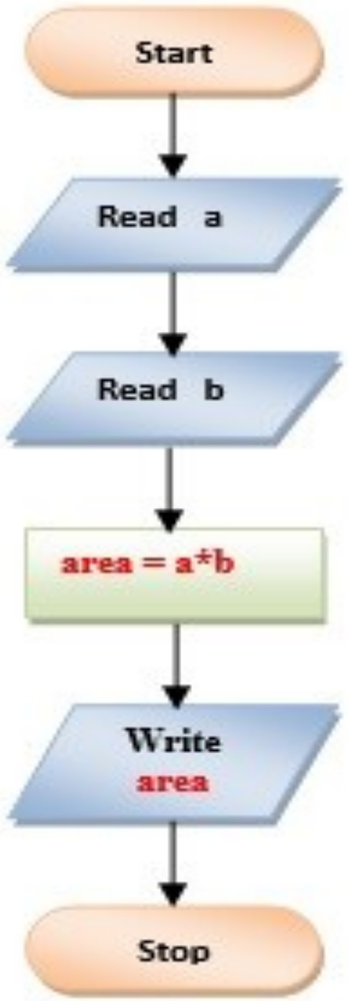
Example

Finding Area of the rectangle

Algorithm

1. Start
2. Read side length, a
3. Read side length b
4. area = a*b
5. Print or display area
6. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{

    int a, b, area;
    printf("Enter side length a: \n");
    scanf("%d", &a);

    printf("Enter side length b: \n");
    scanf("%d", &b);

    area = a*b;

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

    return 0;
}
```

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;
}
```

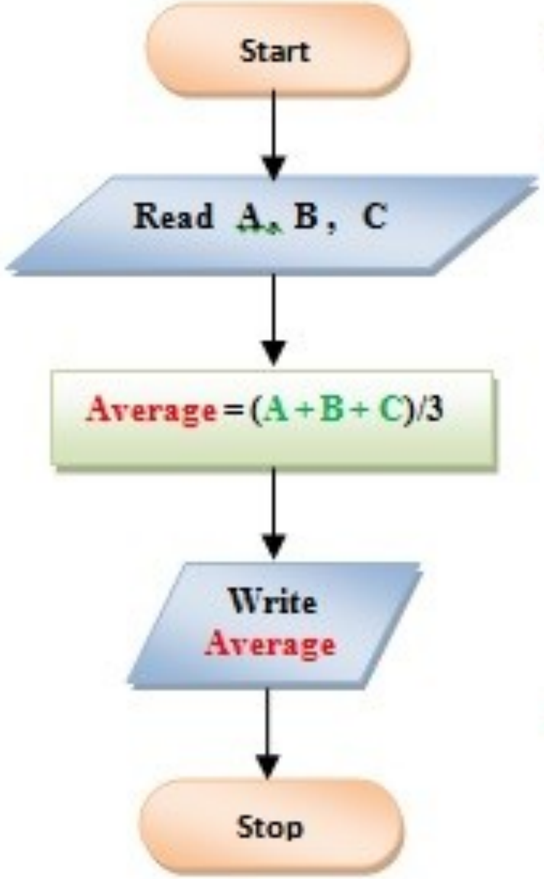
Example

Calculating the average for 3 numbers

Algorithm

1. Start
2. Read 3 numbers A, B, C
3. Calculate the average by the equation:
   Average = (A + B + C)/3
4. Print average
5. Stop

Flowchart



Program

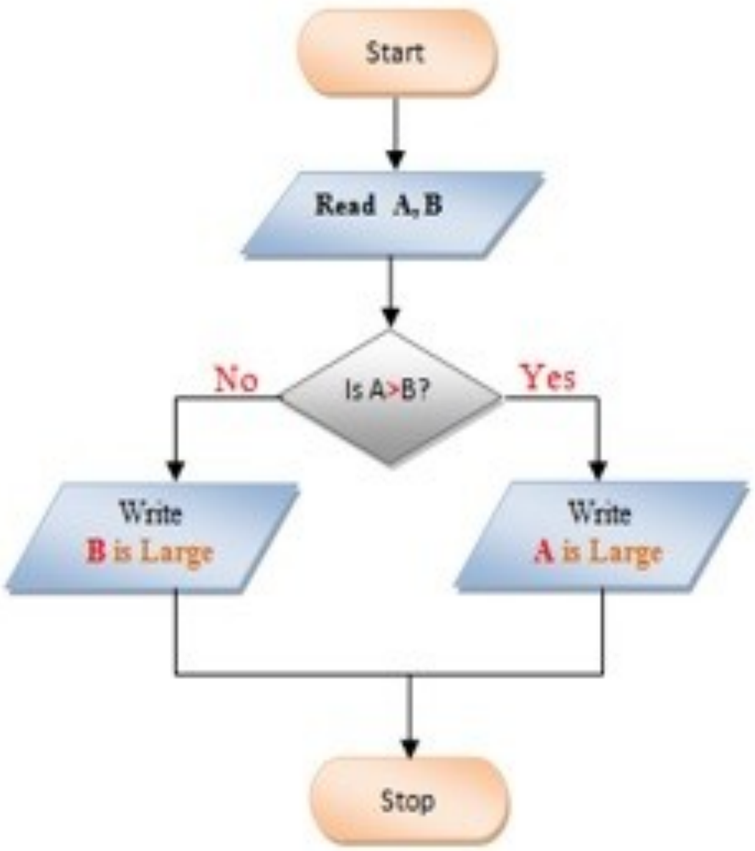
```
#include<stdio.h>

int main()
{

    int A, B, C;
    float Average;

    printf("Enter values of A, B, C: \n");
    scanf("%d %d %d", &A, &B, &C);

    Average = (A+B+C)/3;

    printf("Average of given 3 numbers is: %f", Average);

    return 0;
}
```

Greatest of two numbers

Algorithm

1. Start
2. Read A,B
3. If A > B then
   Print A is large
   else
   Print B is large
4. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{

    int A, B;

    printf("Enter values of A, B: ");
    scanf("%d %d", &A, &B);

    if (A>B)
    printf("A is Larger");
    else
    printf("B is Larger");

    return 0;
}
```

Example

Interchange the value of two numbers

Algorithm

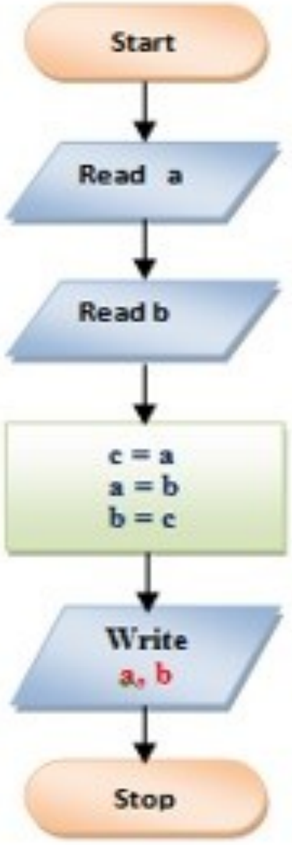
1. Start
2. Read two values into two variables a, b
3. Declare third variable, c

   c = a

   a = b

   b = c
4. Print or display a, b
5. Stop

Flowchart



Program

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

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

    return 0;
}
```

Example

Write an algorithm to find the largest among three different numbers entered by user.

- Step 1: Start
- Step 2: Declare variables a,b and c.
- Step 3: Read variables a,b and c.
- Step 4: If a>b

```
        If a>c
            Display a is the largest number.
        Else
            Display c is the largest number.
    Else
        If b>c
            Display b is the largest number.
        Else
            Display c is the greatest number.
```

- Step 5: Stop