BUILDING BLOCKS OF AN ALGORITHM

Algorithm is defined as the effective step-by-step procedure to solve the problem in a finite number of steps. Algorithms can be designed using the following components. They are

i) Instructions/Statements

ii) State

iii) Control Flow

iv) Functions

Instructions/Statements

Instructions/Statements are the steps which solves a particular problem. Most algorithms have the basic parts of statements.

→Description of the problem.

⟶ Setup

⟶ Parameters

⟶ Execution

⟶ Conclusion

Example: To find the sum of two numbers.

Description of the problem:

To find the sum of two numbers.

Setup:

Two numbers required for addition and one variable for storing the result.

Parameters:

- Read first number.
- Read the second number.

Execution:

Calculate the sum of two numbers (a,b) result = a + b

Conclusion:

The desired output is the sum of two numbers which is stored in the variable 'result'.

State

State is defined as the condition of algorithm at a moment in a time. Algorithm can be in any of the following states.

1. START state
2. INPUT state
3. PROCESS state
4. OUTPUT state
5. STOP state

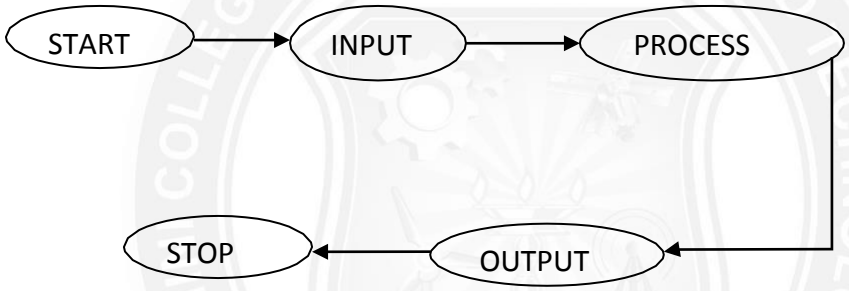


Fig 1: States of an algorithm

1. **START state:**

   It represents the starting of the program.

2. **INPUT state:**

   It represents the process of reading the input from the user.

3. **PROCESS state:**

   It represents the process done in the program. E.g. Addition, Subtraction etc..

4. **OUTPUT state:**

   It represents the output displayed on the screen.

5. **STOP state:**

   It represents the ending of the program.

Example: Algorithm to find the sum of two numbers.

a. Start (START state)

b. Read A, B (INPUT state)

c. C=A+B (PROCESS state)

d. Print C (OUTPUT state)

e. Stop (STOP state)

Control flow

It represents the order of statement execution and direction of flow of programs.

There are three types of control flow.

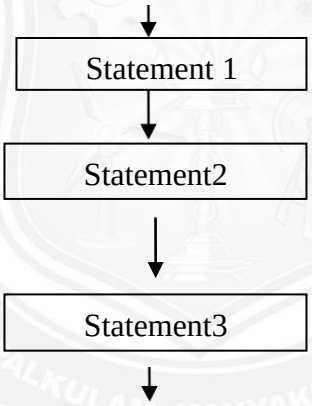
i) Sequential control flow.

ii) Selection control flow.

iii) Repetition control flow.

i) Sequential control flow

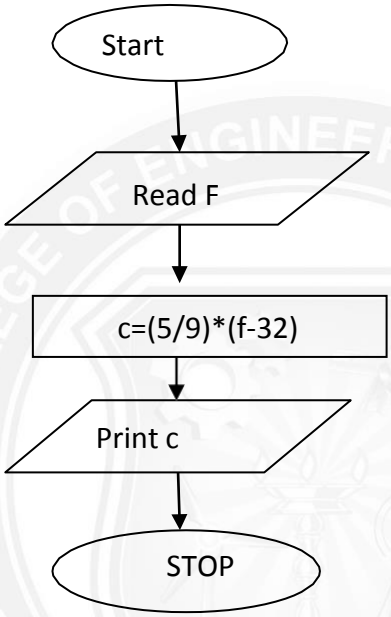
The steps of an algorithm are carried out in a sequential manner.



***Example*:** Temperature conversion

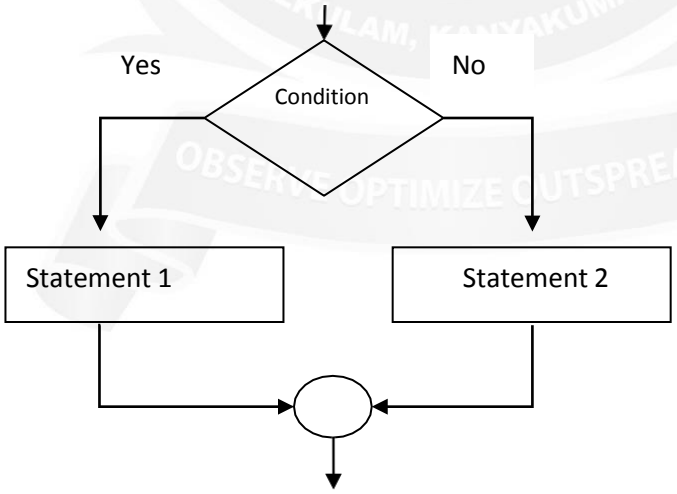
1. Start
2. Read temperature in fahrenheit f.
3. c = (5/9)*(f-32)
4. Print c
5. Stop

Flowchart:



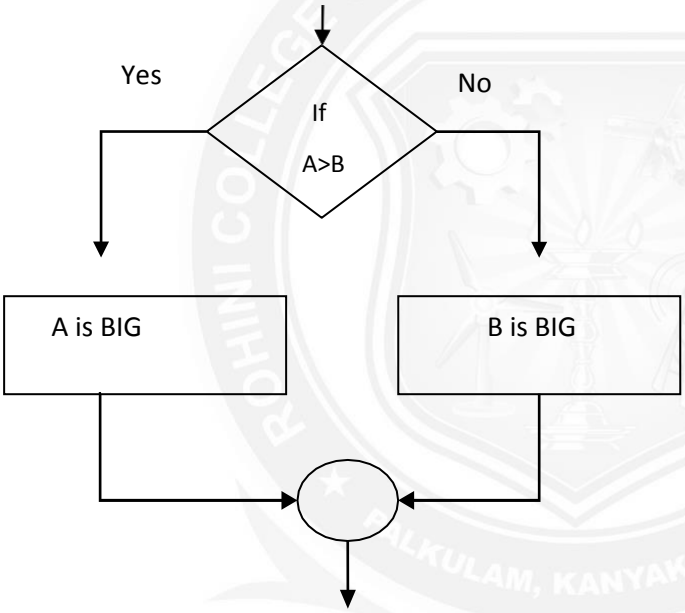
ii) Selection control flow

Only one of the alternative steps is executed based on the condition.



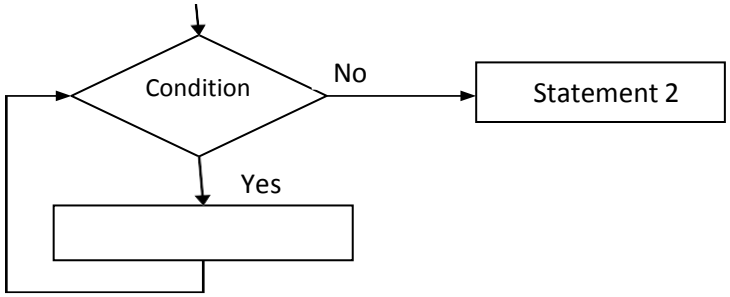
Example: Algorithm to find the biggest among two numbers.

1. Start
2. Read a, b
3. If a>b then Print “a is big”
4. Else Print “b is big”
5. Stop



iii) Repetition control flow

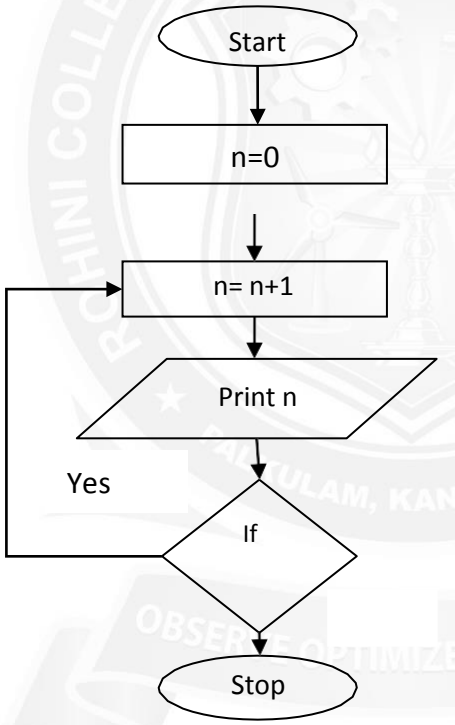
One or more steps are performed repeatedly. This logic is used for producing loops in the program. The looping process can either be one time or multiple times based on the condition.



Example: Algorithm to print numbers from 1 to 10.

1. Start
2. Initialize n=0
3. Increment n=n+1
4. Print n
5. If n<=10 then goto step 3
6. Else goto step 7
7. Stop

Flowchart:



Functions

→A function is a block of organized, reusable code which is used to perform a task or action.

→It provides better modularity.

→They are also called as methods, routines, procedures etc.

→"Add" is a function and it performs addition operation.

Example: Algorithm to find the biggest among N numbers**.**

1. Start
2. Read l1, l2, l3…ln into array.
3. Initialize max=l1
4. Read next number l1, do the following
    i. If max< l1 then
        I) max = l1
    ii. Repeat step" i" until n
5. Print max
6. Stop

Advantages of algorithm

- Easy to understand.
- Easy to debug.
- It is easy for a programmer to convert algorithm into actual program.

Disadvantages of algorithm

- It is time consuming. (An algorithm is developed first, which is then converted into flow chart and then program).
- It is difficult to show branching and looping.