



SNS COLLEGE OF TECHNOLOGY



Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A+’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME :23IT101 C Programming and Data structures

I YEAR/ I SEMESTER

UNIT – I INTRODUCTION TO C

Topic: Algorithm,Flow chart ,Pseudocode

Mrs. G.VANITHA

Assistant Professor

Department of Computer Science and Engineering



Algorithms

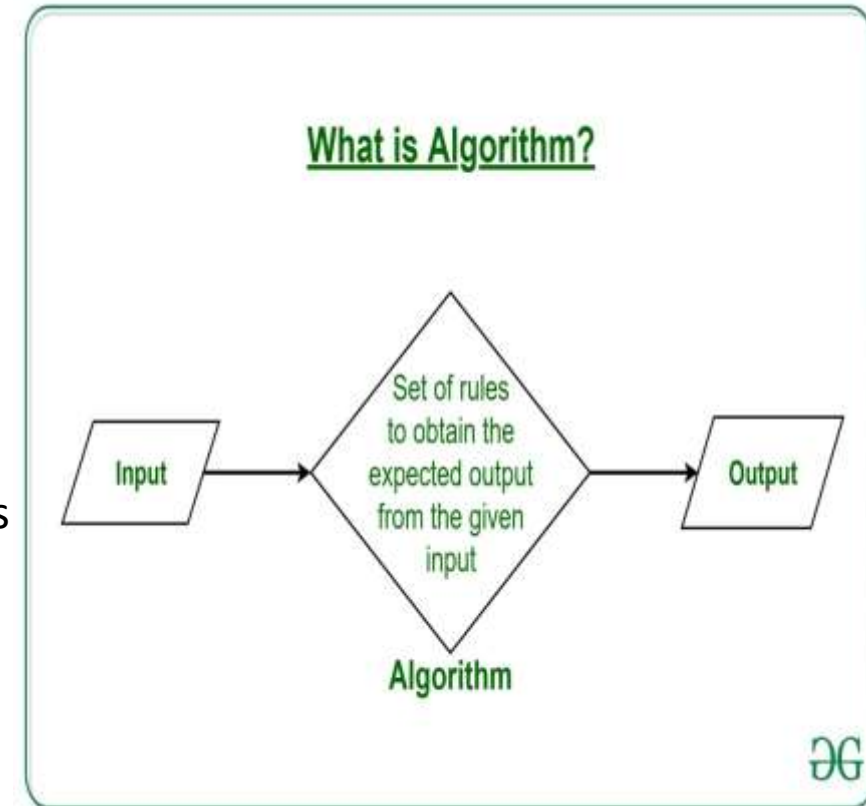
Algorithm is a sequence of instructions that describe a method for solving a problem. It is a step by step procedure for solving a problem

Properties of Algorithms

- Should be written in simple English
- Each and every instruction should be precise and unambiguous
- Instructions in an algorithm should not be repeated infinitely
- Algorithm should conclude after a finite number of steps
- Should have an end point
- Derived results should be obtained only after the algorithm terminates

Qualities of a good algorithm

1. Time
2. Memory
3. Accuracy





Algorithms

Example: C program for Print the “WELCOME TO SNSCT”

Program:

```
1. #include <stdio.h>
//where the execution of program begins
1. Int main()
2. {
3. Printf(“ WELCOME TO SNSCT”);
4. Return 0;
5. }
```

Output:

WELCOME TO SNSCT

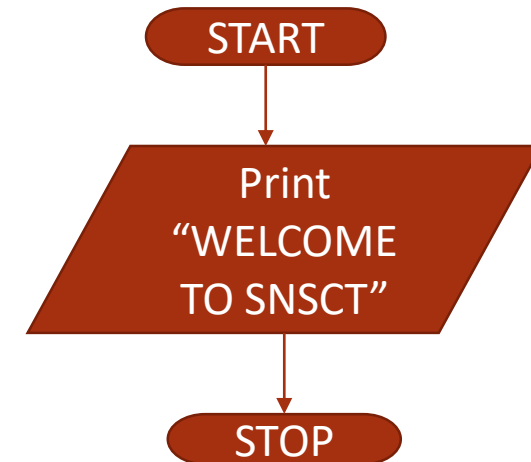
Algorithm:

Step 1: Start

Step 2: Print “WELCOME TO SNSCT”

Step 3: Stop

Flowchart:





Building Blocks of Algorithms

Algorithms can be constructed from basic building blocks namely,

1. Statements:
2. State
3. Control Flow
4. Functions





Building Blocks of Algorithms

1.Statements:

Statement is a single action in a computer.

1. Input Data
2. Process Data
3. Output Data

2.State:

Transition from one process to another process under specified condition with in a time is called state

3.Control flow:

The process of executing the individual statements in a given order is called control flow







The control can be executed in three ways

1. Sequence
2. Selection
3. Iteration



FLOWCHART SYMBOLS

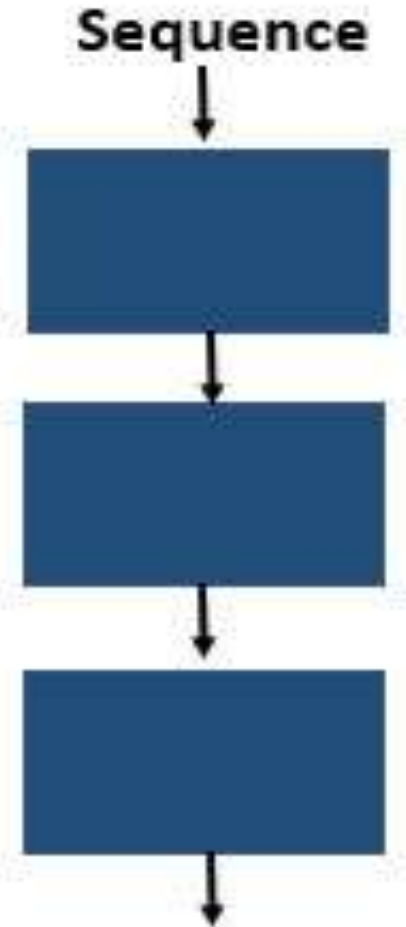
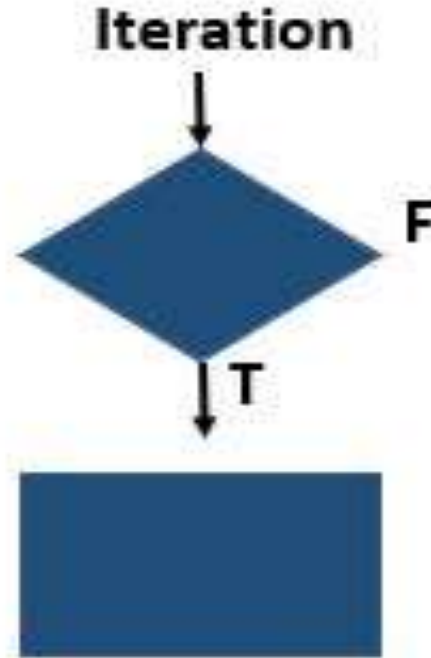
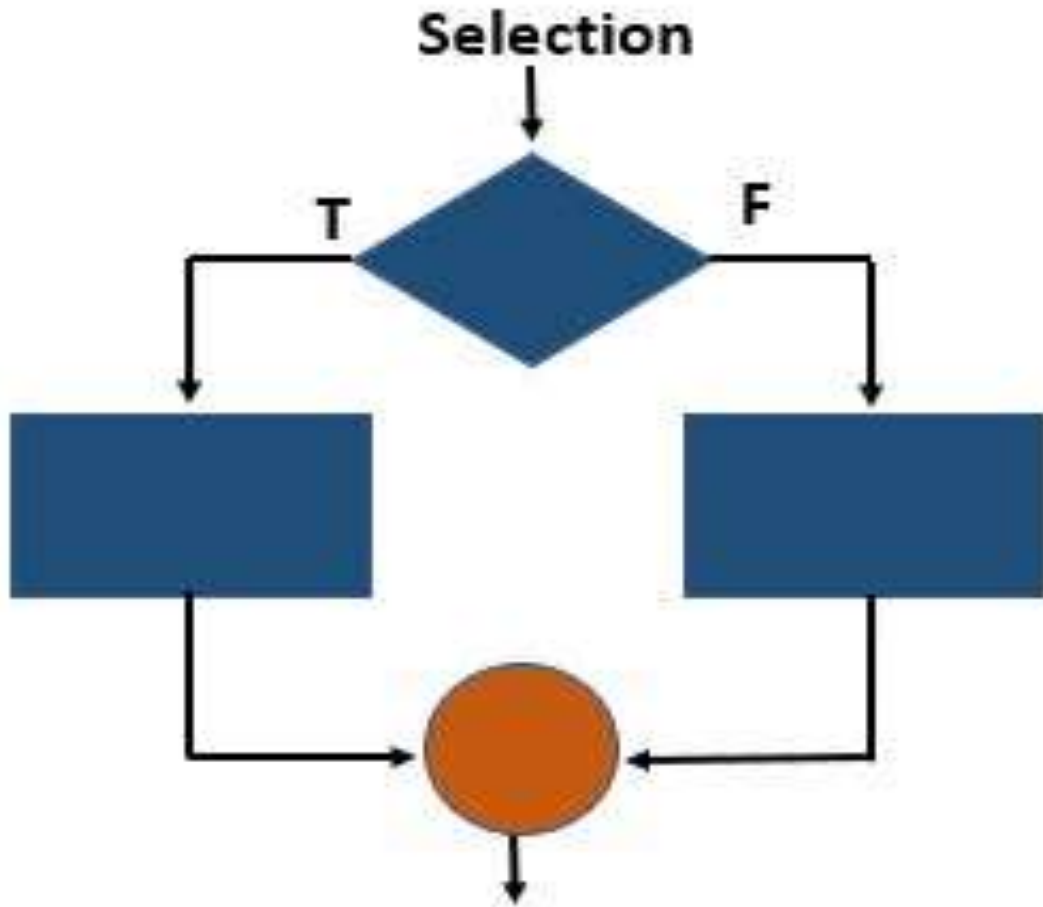
Basic

Name	Symbol	Use in Flowchart
Oval		Denotes the beginning or end of the program
Parallelogram		Denotes an input operation
Rectangle		Denotes a process to be carried out e.g. addition, subtraction, division etc.
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes. (e.g. IF/THEN/ELSE)
Hybrid		Denotes an output operation
Flow line		Denotes the direction of logic flow in the program





Building Blocks of Algorithms



www.educba.com



STEPS IN PROBLEM SOLVING

- First produce a general algorithm (one can use ***pseudo code***)
- Refine the algorithm successively to get step by step detailed ***algorithm*** that is very close to a computer language.
- ***Pseudo code*** is an artificial and informal language that helps programmers develop algorithms. Pseudo code is very similar to everyday English.



Building Blocks of Algorithms

1.Sequence: All the instructions are executed one after another is called sequence execution

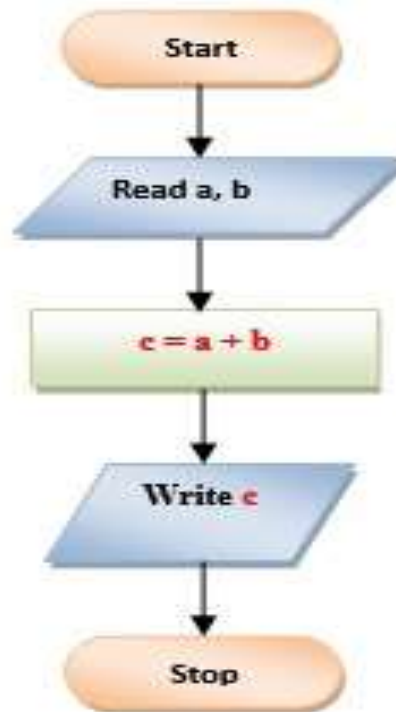
Example: Algorithm for Addition of TWO NUMBERS

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);

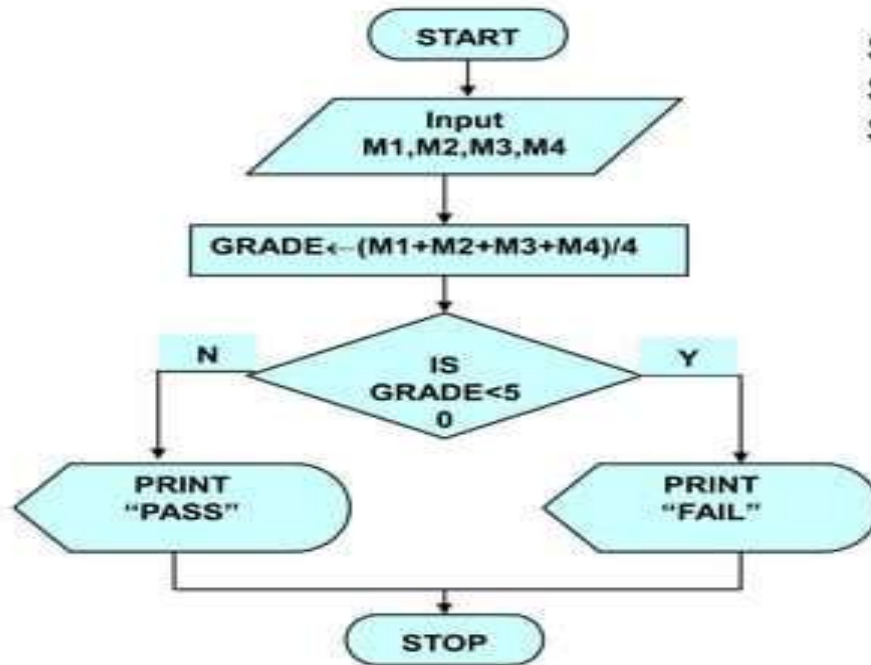
    printf("Enter value of b: ");
    scanf("%d", &b);
    c = a+b;

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

    return 0;
}
```



EXAMPLE 1



Step 1: Input M1, M2, M3, M4
Step 2: $GRADE \leftarrow (M1+M2+M3+M4)/4$
Step 3: if (GRADE < 50) then
 Print "FAIL"
 else
 Print "PASS"
 endif



EXAMPLE 2

- Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

Pseudo code:

- *Input the length in feet (Lft)*
- *Calculate the length in cm (Lcm) by multiplying LFT with 30*
- *Print length in cm (LCM)*

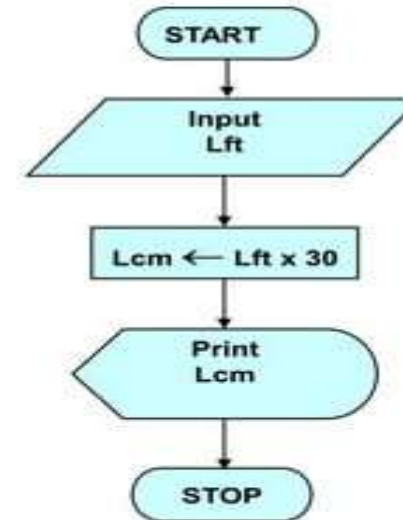




Algorithm

- Step 1: Input Lft
- Step 2: $Lcm \leftarrow Lft \times 30$
- Step 3: Print Lcm

Flowchart





EXAMPLE 3

Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.

Pseudocode

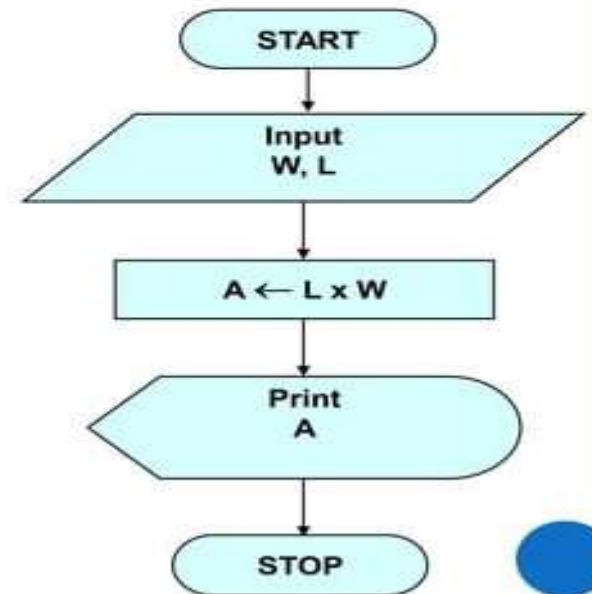
- *Input the width (W) and Length (L) of a rectangle*
- *Calculate the area (A) by multiplying L with W*
- *Print A*





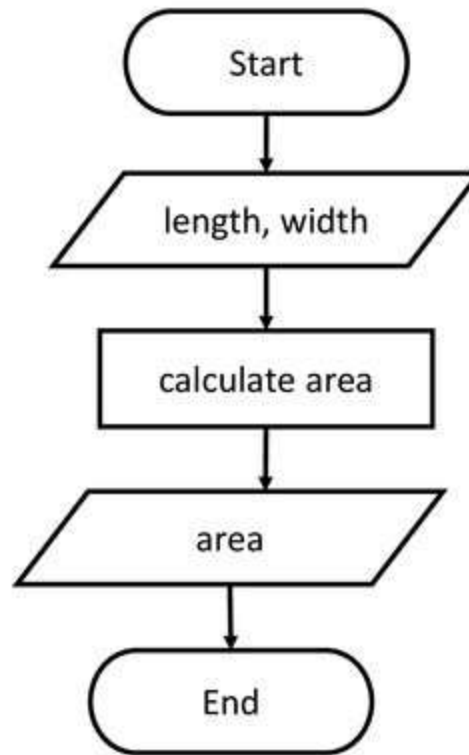
Algorithm

- Step 1: Input W,L
- Step 2: $A \leftarrow L \times W$
- Step 3: Print A





Flowchart Example



Design an algorithm to find the area of a rectangle

The formulas: $\text{area} = \text{length} * \text{width}$

Input	Process	Output
<u>Input variable:</u> length width	<u>Processing item:</u> area <u>Formula:</u> $\text{area} = \text{length} \times \text{width}$ <u>Step / Solution algorithm:</u> get input calculate area display output	<u>Output:</u> area



Flow Chart



Rules for drawing a flowchart

- The flowchart should be clear, neat and easy to follow.
- The flowchart must have a logical start and finish.
- Only one flow line should come out from a process symbol.
- Only one flow line should enter a decision symbol.
- two or three flow lines may leave the decision symbol
- Only one flow line is used with a terminal symbol.
- Intersection of flow lines should be avoided.

Advantages of flowchart:

1. Communication
2. Effective analysis
3. Proper documentation
4. Efficient Coding
5. Proper Debugging
6. Efficient Program Maintenance

Disadvantages of flowchart:

1. Complex logic
2. Alterations and Modifications
3. Reproduction
4. Cost



Building Blocks of Algorithms

2.Selection: A selection statement causes the program control to be transferred to a specific part of the program based upon the condition. If the conditional test is true, one part of the program will be executed, otherwise it will execute the other part of the program.

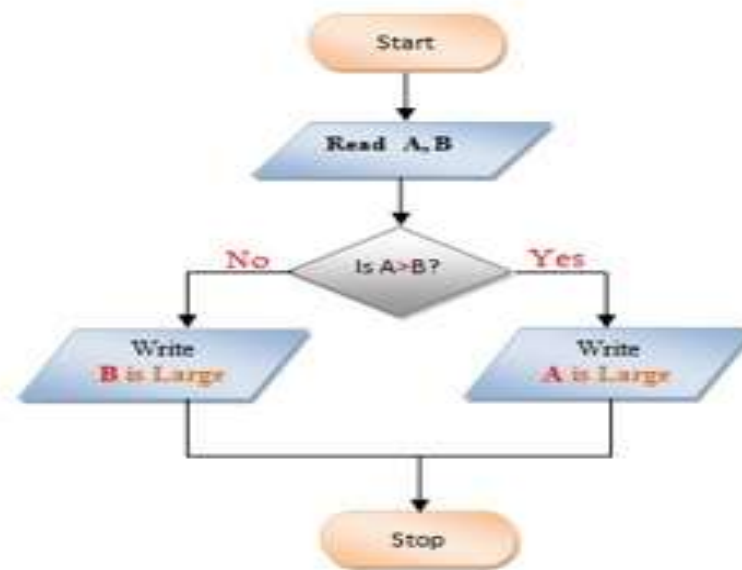
Example: Algorithm for Greatest of TWO NUMBERS

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;
}
```



Building Blocks of Algorithms

3.Iteration:In programs, certain set of statements are executed again and again based upon conditional test. It executed more than one time. This type of execution is called looping or iteration.

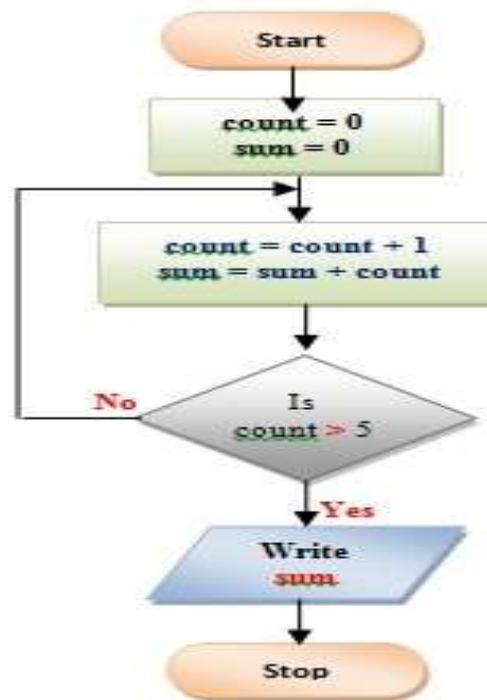
Example: Algorithm for sum of FIRST FIVE NATURAL NUMBERS

Find the Sum of First Five Natural Numbers

Algorithm

1. Start
2. Initialize count = 0, sum = 0
3. count = count + 1
4. sum = sum + count
5. Repeat steps 3,4 until count > 5
6. Print sum
7. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int count, sum;
    sum = 0;
    for (count = 1; count<=5; count++)
    {
        sum = sum +count;
    }
    printf("Sum of 1st 5 numbers is: %d", sum);
    return 0;
}
```



Pseudo Code



What is Pseudo Code?



- Pseudo code consists of short, readable and formally styled English languages used for explain an algorithm.
- It does not include details like variable declaration, subroutines.
- It is easier to understand for the programmer or non programmer to understand the general working of the program.
- It is not a machine readable
- Pseudo code can't be compiled and executed.
- No standard syntax.

Pseudocode

```
IF a1 > a2
    PRINT a1
ELSE
    PRINT a2
```

```
if a1 > a2:
    print(a1)
else:
    print(a2)
```

```
if a1 > a2{
    print(a1)
}
else{
    print(a2)
}
```



WTMatter



Pseudo Code

Guidelines for writing pseudo code:

- Write one statement per line
- Capitalize initial keyword
- End multiline structure
- Keep statements language independent

Common keywords used in pseudocode

begin ... end: These keywords are used to start and finish pseudocode. Begin is the first line and end is the last line of pseudocode.

accept: This keyword is used to obtain an input from a user.

display: This keyword is used to present a result or an output.

if ... else... endif: These keywords are used in decision-making.

//: Comment

Do ... while, for ..., repeat ... until: Represent loop



Pseudo Code

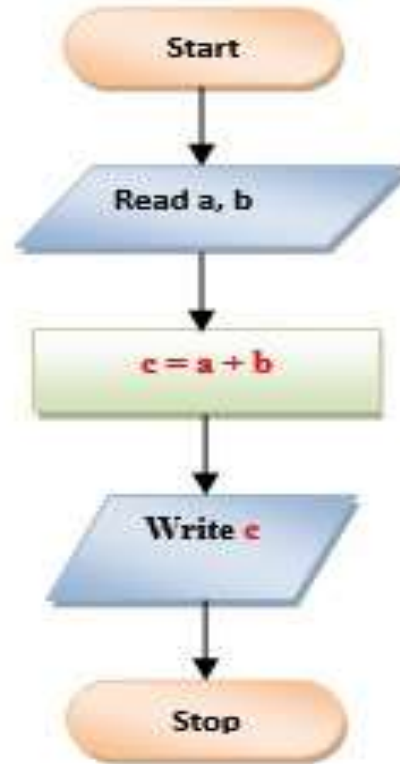
Example for Sequence Method:

To find sum of two numbers

Pseudo code

BEGIN
GET a,b
ADD $c=a+b$
PRINT c
END

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;
}
```



Pseudo Code

Example for Selection Method:

Greatest of two numbers

Pseudocode

```
PROGRAM PrintBiggerOfTwo:  
  Read A;  
  Read B;  
  IF (A>B)  
    THEN Print A;  
    ELSE Print B;  
  ENDIF;  
END.
```

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;  
}
```




Comparisons

Algorithm	Flowchart	Pseudo code
An algorithm is a sequence of instructions used to solve a problem	It is a graphical representation of algorithm	It is a language representation of algorithm.
User needs knowledge to write algorithm.	not need knowledge of program to draw or understand flowchart	Not need knowledge of program language to understand or write a pseudo code.



Thank
you



Thank you!