ALGORITHM & FLOWCHART MANUAL for STUDENTS
Part (1)

Algorithm in Programming:

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

HOW TO WRITE ALGORITHMS:

Step 1 **Define your algorithms input**: Many algorithms take in data to be processed, e.g. to calculate the area of rectangle input may be the rectangle height and rectangle width.
Step 2 **Define the variables**: Algorithm's variables allow you to use it for more than one place. We can define two variables for rectangle height and rectangle width as HEIGHT and WIDTH (or H & W). We should use meaningful variable name e.g. instead of using H & W use HEIGHT and WIDTH as variable name.
Step 3 **Outline the algorithm's operations:** Use input variable for computation purpose, e.g. to find area of rectangle multiply the HEIGHT and WIDTH variable and store the value in new variable (say) AREA. An algorithm's operations can take the form of multiple steps and even branch, depending on the value of the input variables.
Step 4 **Output the results of your algorithm's operations**: In case of area of rectangle output will be the value stored in variable AREA. if the input variables described a rectangle with a HEIGHT of 2 and a WIDTH of 3, the algorithm would output the value of 6.

Flowchart:

Flowchart is diagrammatic /Graphical representation of sequence of steps to solve a problem. To draw a flowchart following standard symbols are use

Symbol	Purpose	Description
	Flow line	**Used to indicate the flow of logic by connecting symbols.**

Symbol	Purpose	Description
	Terminal(Stop/Start)	Used to represent start and end of flowchart.
	Input/Output	Used for input and output operation.
	Processing	Used for airthmetic operations and data-manipulations.
	Decision	Used to represent the operation in which there are two alternatives, true and false.
	On-page Connector	Used to join different flowline
	Off-page Connector	Used to connect flowchart portion on different page.
	Predefined Process/Function	Used to represent a group of statements performing one processing task.

Example (1)

Write the Algorithm and Draw a flowchart to add two numbers?

Algorithm

Step-1 Start

Step-2 Input first number say A

Step-3 Input second number say B

Step-4 SUM = A + B

Step-5 Display SUM

Step-6 Stop

Or

Algorithm

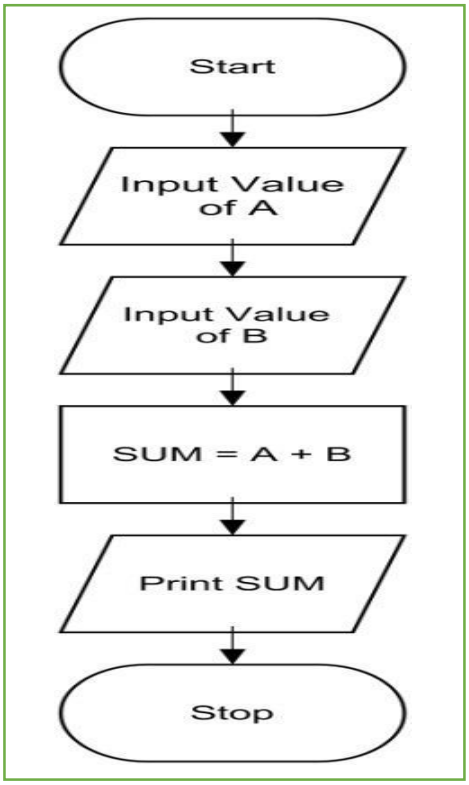
Step-1 Start

Step-2 Input two numbers say A & B

Step-3 SUM = A + B

Step-4 Display SUM

Step-5 Stop



Example (2)

Convert temperature from Celsius to Fahrenheit

C : temperature in Celsius
F : temperature Fahrenheit

Algorithm

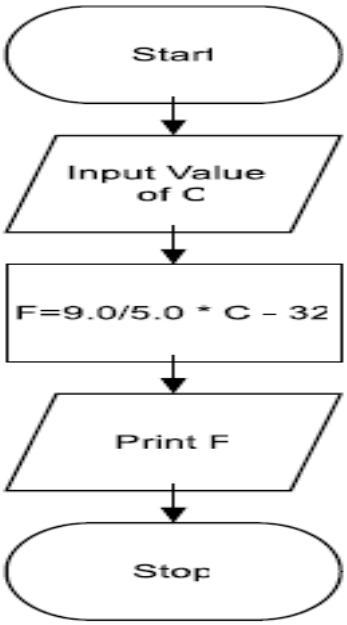
Step-1 Start

Step-2 Input temperature in Celsius say C

Step-3 F = (9.0/5.0 x C) + 32

Step-4 Display Temperature in Fahrenheit F

Step-5 Stop



Example (3)

Find Area and Perimeter of Square:

L : Side Length of Square
AREA : Area of Square
PERIMETER : Perimeter of Square

Algorithm

Step-1 Start

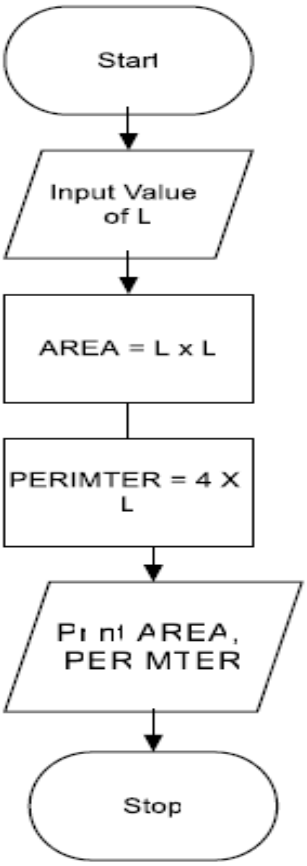
Step-2 Input Side Length of Square say L

Step-3 Area = L x L

Step-4 PERIMETER = 4 x L

Step-5 Display AREA, PERIMETER

Step-6 Stop



Example (4)

Find Area and Perimeter of Rectangle:

L : Length of Rectangle طول المستطيل
B : Breadth of Rectangle عرض المستطيل
AREA : Area of Rectangle
PERIMETER : Perimeter of Rectangle

Algorithm

Step-1 Start

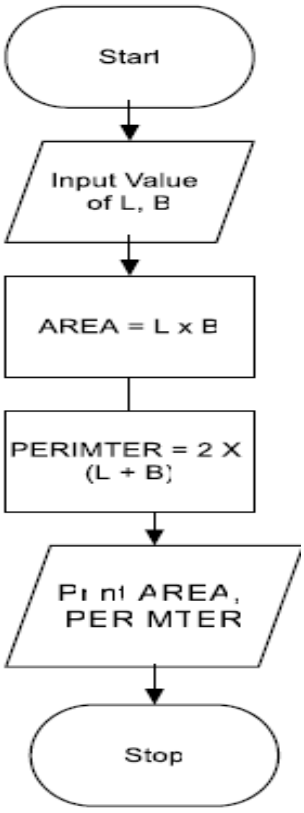
Step-2 Input Side Length & Breadth say L, B

Step-3 Area = L x B

Step-4 PERIMETER = 2 x (L + B)

Step-5 Display AREA, PERIMETER

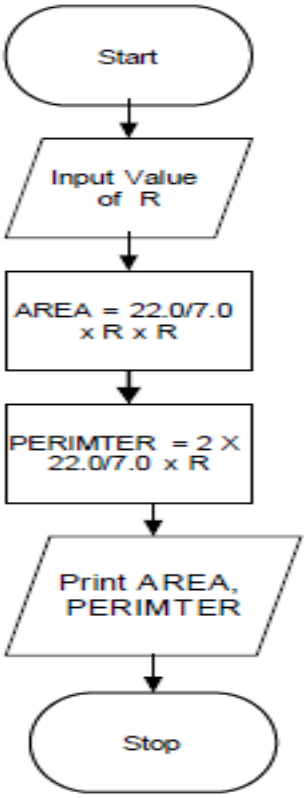
Step-6 Stop



Example (5)

Find Area and Perimeter of Circle:

R : Radius of Circle
AREA : Area of Circle
PERIMETER : Perimeter of Circle



Algorithm

Step-1 Start

Step-2 Input Radius of Circle say R

Step-3 Area = 22.0/7.0 x R x R **(or) (3.14 ×R ×R)**

Step-4 PERIMETER = 2 x 22.0/7.0 x R**(2 × 3.14 ×R)**

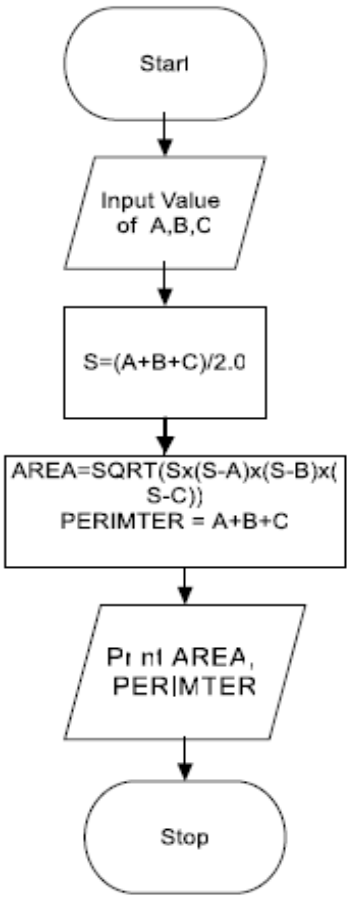
Step-5 Display AREA, PERIMETER

Step-6 Stop

Example (6)

Find Area and Perimeter of Triangle:

A : First Side of Triangle
B : Second Side of Triangle
C : Third Side of Triangle
AREA : Area of Triangle
PERIMETER : Perimeter of Triangle



Algorithm

Step-1 Start

Step-2 Input Sides of Triangle A,B,C

Step-3 S= (A + B + C)/ 2.0

Step-4 AREA = SQRT(S x (S-A) x (S-B) x(S-C))

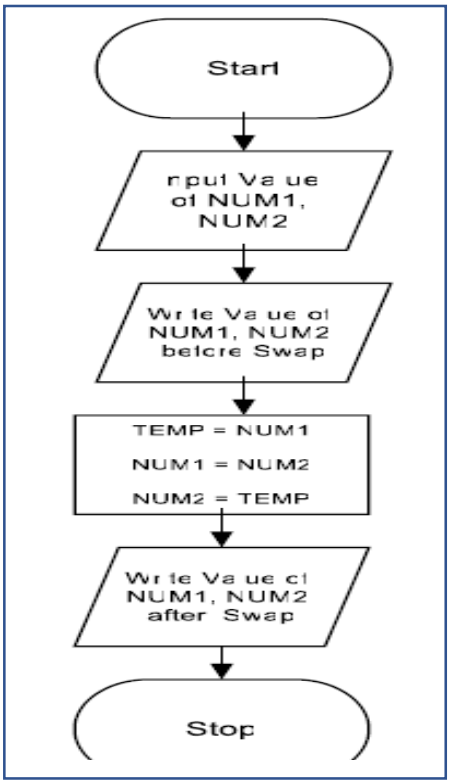
Step-5 PERIMETER = A + B + C

Step-6 Display AREA, PERIMETER

Step-7 Stop

Example (7)

Algorithm & Flowchart to Swap Two Numbers using Temporary Variable :



Algorithm

Step-1 Start

Step-2 Input Two Numbers Say NUM1,NUM2

Step-3 Display Before Swap Values NUM1, NUM2

Step-4 TEMP = NUM1

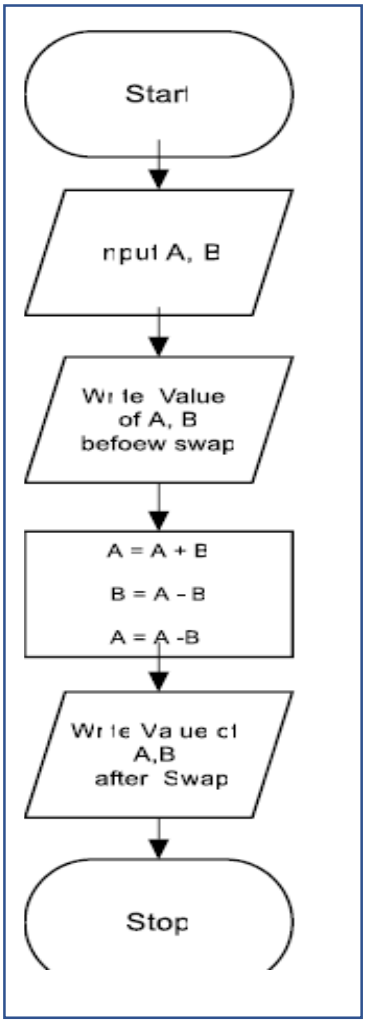
Step-5 NUM1 = NUM2

Step-6 NUM2 = TEMP

Step-7 Display After Swap Values NUM1,NUM

Step-8 Stop

Algorithm & Flowchart to Swap Two Numbers without using temporary variable :



Algorithm

Step-1 Start

Step-2 Input Two Numbers Say A,B

Step-3 Display Before Swap Values A, B

Step-4 A = A + B

Step-5 B = A - B

Step-6 A = A - B

Step-7 Display After Swap Values A, B

Step-8 Stop

Conditional Statements : if, else, switch

Conditional statements help you to make a decision based on certain conditions. These conditions are specified by a set of conditional statements having Boolean expressions which are evaluated to a Boolean value true or false.

1. if statement
2. If-Else statement
3. Nested If-else statement
4. If-Else If
5. Switch statement

If statement

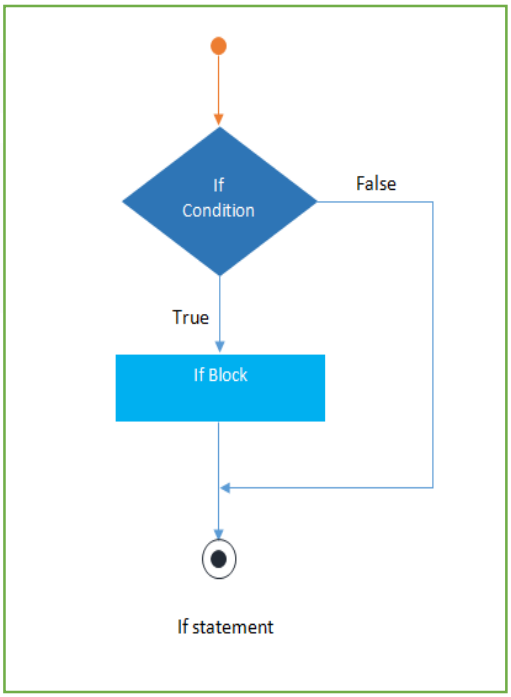
The single if statement in is used to execute the code if a condition is true. It is also called one-way selection statement.

Syntax

```
If(condition)Then
   'statement(s) 'will execute if the condition is true
End If
```

How "if" statement works

- If the expression is evaluated to nonzero (true) then if block statement(s) are executed.
- If the expression is evaluated to zero (false) then Control passes to the next statement following it.

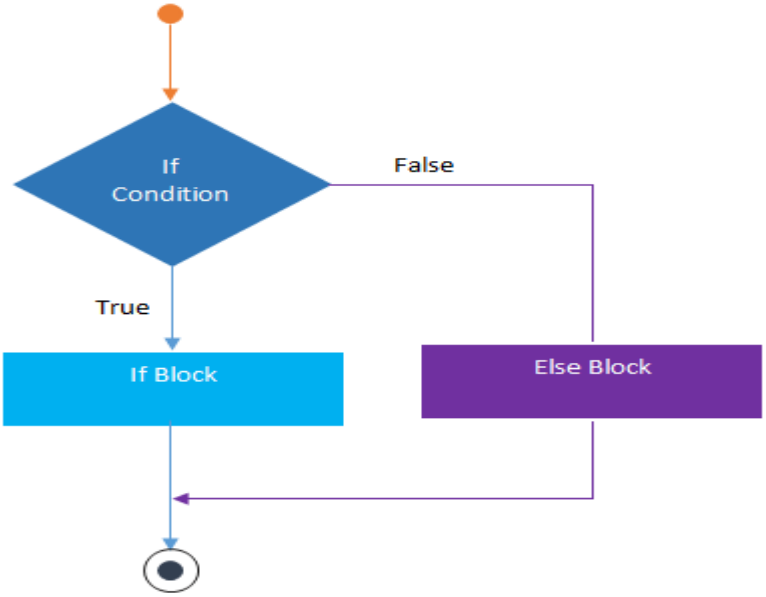


If-else statement

The if-else statement is used to execute the code if condition is true or false. It is also called two-way selection statement.

Syntax

```
If(condition)Then
   'statement(s) will execute if the condition is true
Else
  'statement(s) will execute if the condition  is false
End If
```



If-Else statement

How "if..else" statement works..

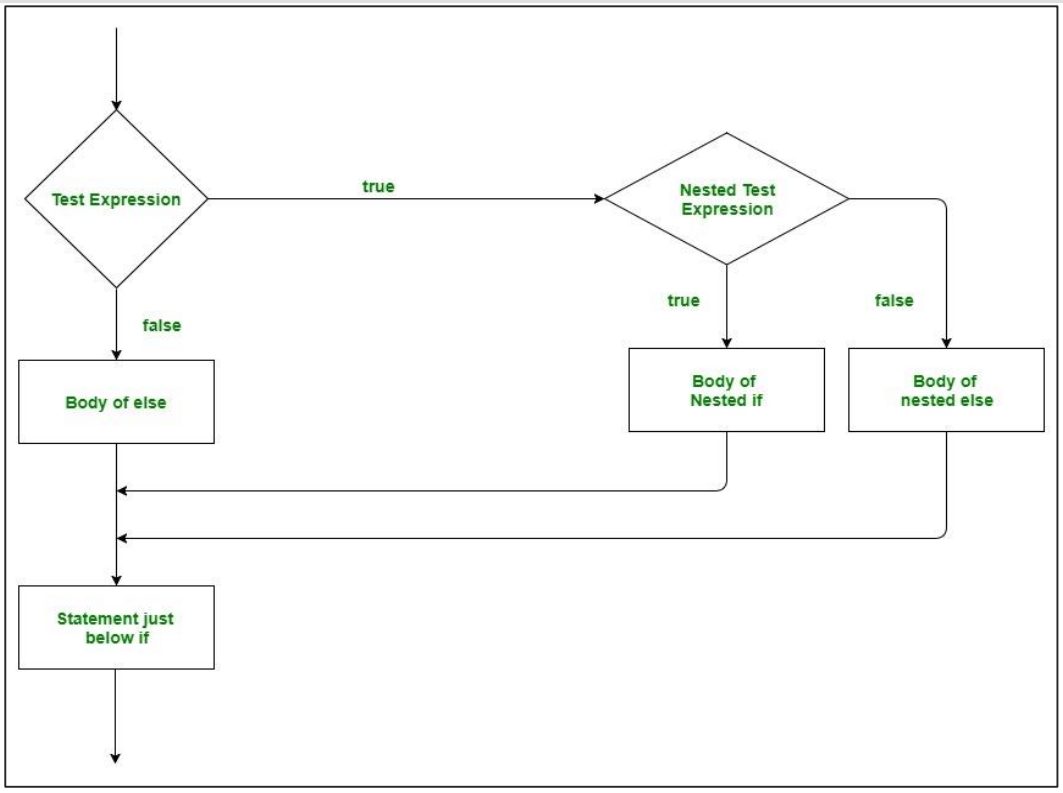
- If the expression is evaluated to nonzero (true) then if block statement(s) are executed.
- If the expression is evaluated to zero (false) then else block statement(s) are executed.

Nested If statement

The nested if statement is used when a program requires more than one test expression. It is also called a multi-way selection statement. When a series of the decision are involved in a statement, we use if else statement in nested form

Syntax

```
if (condition1) {

   // Executes when condition1 is true

   if (condition2) {

      // Executes when condition2 is true
   }
}
```

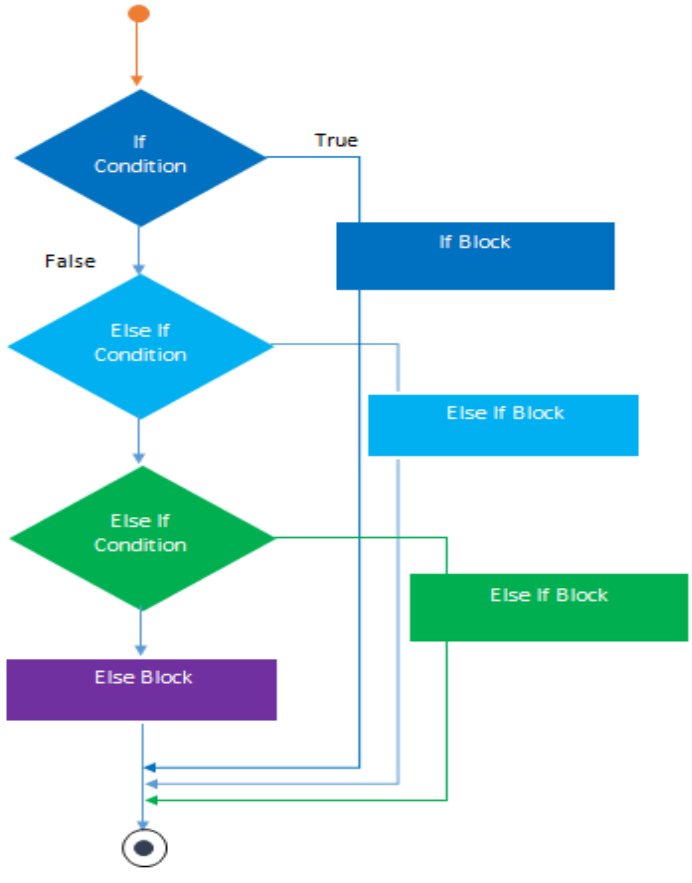


If..else If

The if-else-if statement is used to execute one code from multiple conditions. It is also called multipath decision statement. It is a chain of if..else statements in which each if statement is associated with else if statement and last would be an else statement.

Syntax

```
if(condition_1) {
     // this block will execute
     // when condition_1 is true
} else if(condition_2) {
    // this block will execute
    // when condition2 is true
}
.....
. else {
      // this block will execute when none
     // of the condition is true
}
}
```



If-Else-If Ladder

Example (8)

Algorithm & Flowchart to find the smallest of two numbers:

Algorithm

Step-1 Start

Step-2 Input two numbers say

NUM1,NUM2

Step-3 IF NUM1 < NUM2 THEN

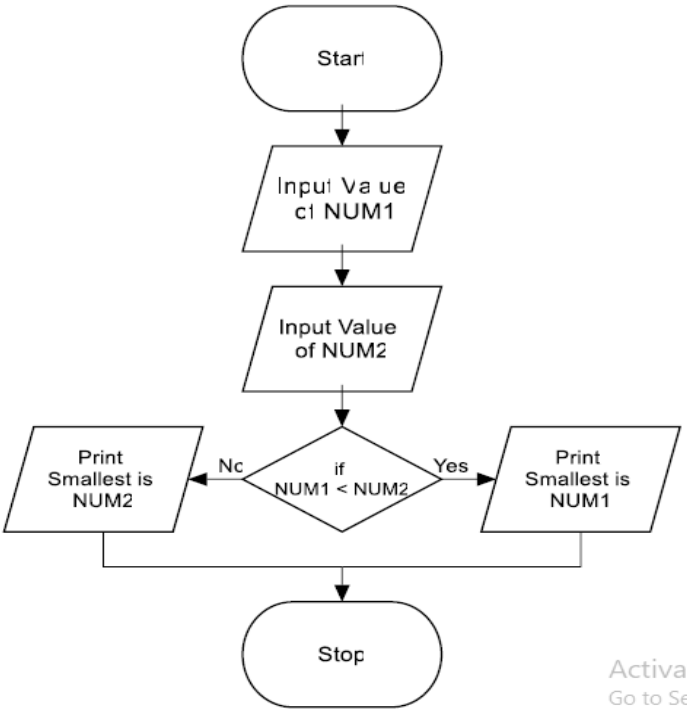
print smallest is NUM1

ELSE

print smallest is NUM2

ENDIF

Step-4 Stop



Algorithm & Flowchart to find the largest of two numbers:

Algorithm

Step-1 Start

Step-2 Input two numbers say

NUM1,NUM2

Step-3 IF NUM1 > NUM2 THEN

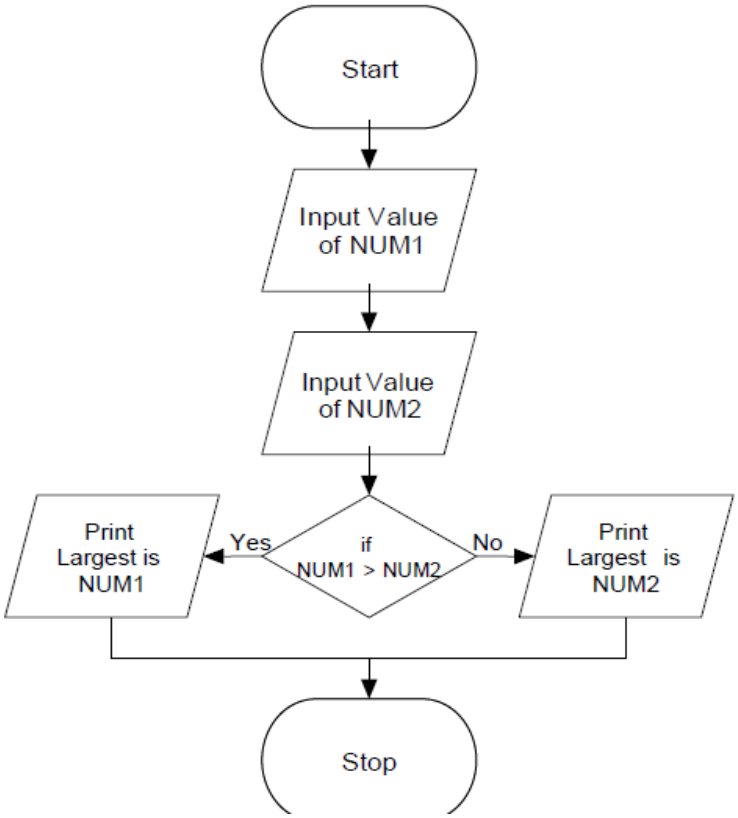
print largest is NUM1

ELSE

print largest is NUM2

ENDIF

Step-4 Stop



Example (9)

Algorithm & Flowchart to find the largest of three numbers:

Step 1: *Input* N1, N2, N3
Step 2: *if* (N1>N2) *then*
if (N1>N3) *then*
MAX N1 [N1>N2, N1>N3]
else
MAX N3 [N3>N1>N2]
endif
else
if (N2>N3) *then*
MAX N2 [N2>N1, N2>N3]
else
MAX N3 [N3>N2>N1]
endif
endif
Step 3: *Print "The largest number is", MAX*

Step4: Stop

Another method

Step-1 Start

Step-2 Read three numbers say num1,num2, num3

Step-3 if (num1>num2) and (num1>num3) then

Print num1 is largest

Else if (num2>num1) and (num2>num3) then

Print num2 is largest

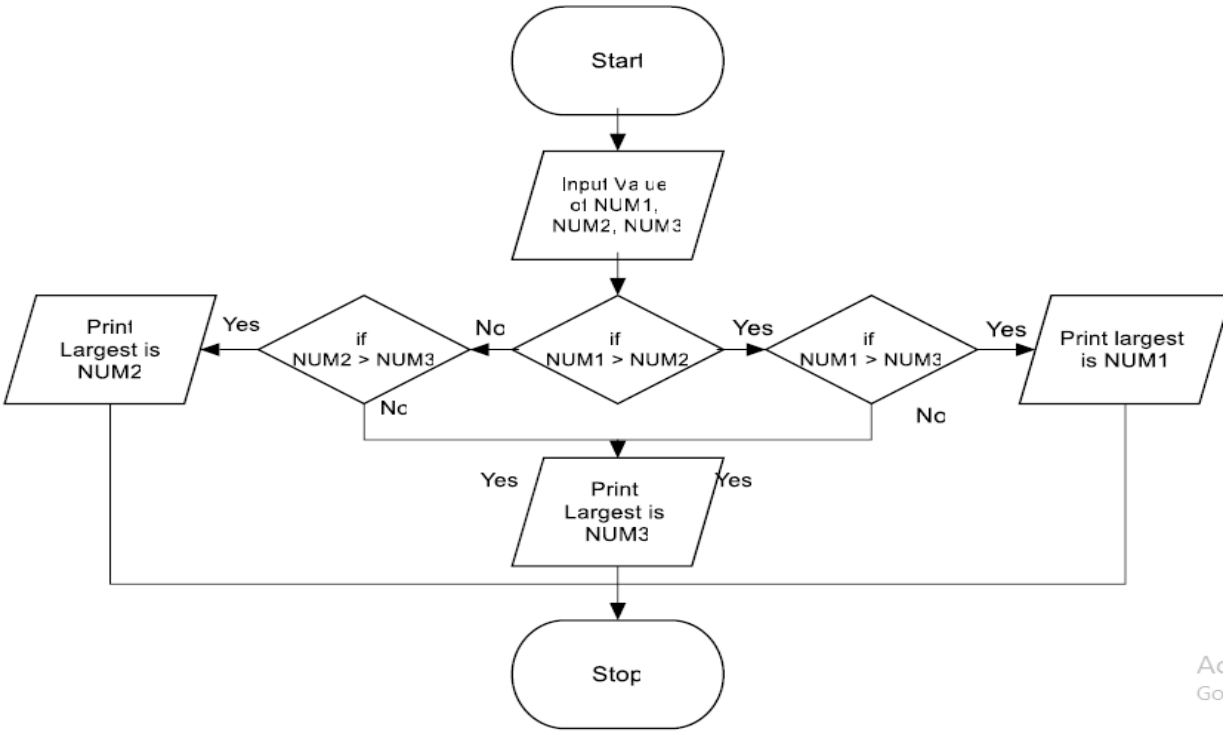
Else

Print num3 is largest

End if

End if

Step-4 Stop



Algorithm

Step-1 Start

Step-2 Read three numbers say A,B,C

Step-3 BIG = A

Step-4 IF B > BIG THEN

BIG = B

ENDIF

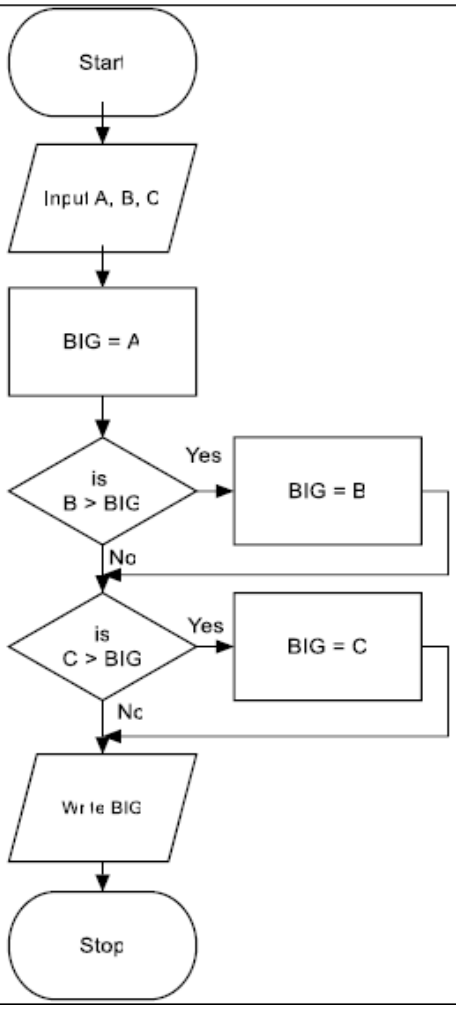
Step-5 IF C >BIG THEN

BIG = C

ENDIF

Step-6 Write BIG

Step-7 Stop



Tasks

1. Find Even numbers between 1 to 50

Algorithm

Step-1 Start

Step-2 I = 1

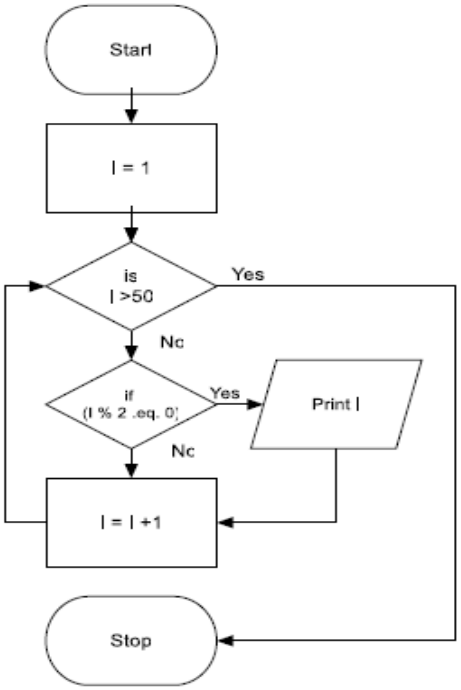
Step-3 IF (I >50) THEN
GO TO Step-7
ENDIF

Step-4 IF ((I % 2) =0) THEN
Display I
ENDIF

Step-5 I = I + 1

Step-6 GO TO Step--3

Step-7 Stop



2. write algorithm and drow flowchart to find the result of equation:

$$f(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Step1: Start

Step2: Read/input x

Step3: if X greater than or equal zero then

F=X

Else

F=-X

End if

Step4: Print F

Step5: End

3. Write and algorithm and draw a flowchart to:

a) Read an employee name (NAME), overtime hours worked (OVERTIME), hours absent (ABSENT)

b) Determine the bonus payment (PAYMENT).

Bonus Schedule	
OVERTIME – (2/3)*ABSENT	Bonus Paid
>40 hours >30 but ≤ 40 hours >20 but ≤ 30 hours >10 but ≤ 20 hours ≤ 10 hours	$50 $40 $30 $20 $10

Step 1: Input NAME, OVERTIME, ABSENT

Step 2: if (OVERTIME–(2/3)*ABSENT > 40) then

PAYMENT 50

else if (OVERTIME–(2/3)*ABSENT > 30) then

PAYMENT 40

else if (OVERTIME–(2/3)*ABSENT > 20) then

PAYMENT 30

else if (OVERTIME–(2/3)*ABSENT > 10) then

PAYMENT 20

else

PAYMENT 10

End if

Step 3: Print “Bonus for”, NAME “is $”, PAYMENT

Step 4: Stop

Example 10

Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

Algorithm

Step 1: Input M1, M2, M3, M4

Step 2: GRADE = (M1+M2+M3+M4)/4

Step 3: if (GRADE < 60) then

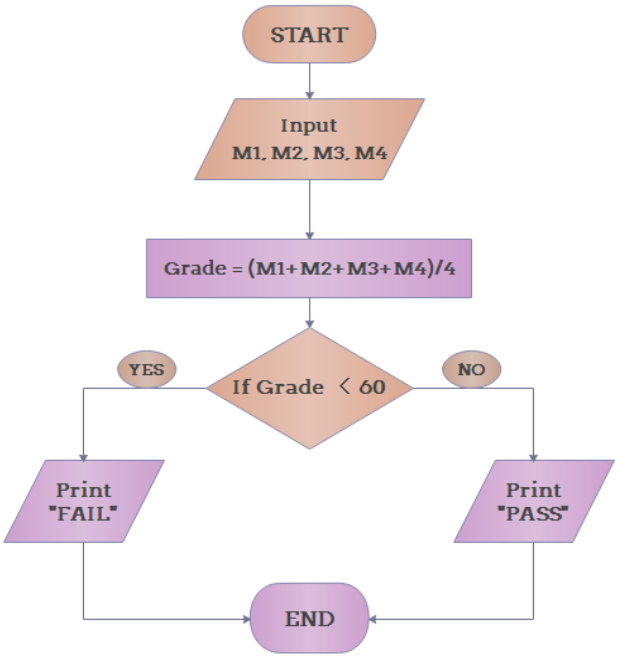
Print "FAIL"

else

Print "PASS"

Endif

Step 4: Stop



Example 11

Write an algorithm that accept an integer number from the user, in case of the number is Positive, check and print out whether it is Even or Odd number.

Algorithm

Step 1: Read number from user say N
Step 2: If (N > 0) then
If (N % 2 == 0) then
Print "Number is Even"
else
Print "Number is Odd"
End if
End if
Step 4: Stop

Example 12

Write An Algorithm That Read Student's Mark As Integer Then Print The Equivalent Grade Depends On The Following Table:

0≤Mark<60	60≤Mark<65	65≤Mark<75	75≤Mark<85	85≤Mark ≤100
Fail	OK	Good	Very Good	Excellent

Algorithm

Step 1: Start
Step 2: Read mark
Step 3: If (mark < 60) then
Print "Fail"
Else If (mark < 65) then
Print "Accept"
Else If (mark < 75) then
Print "Good"
Else If (mark < 85) then
Print " Very Good"
Else If (mark <= 100) then
Print " Excellent"
Else
Print "Invalid Mark! Try again!"
Step 4: stop

Example 13

Write an algorithm and draw a flowchart that will calculate the roots of a quadratic equation:

$$ax^2+bx+c=0$$

Where:

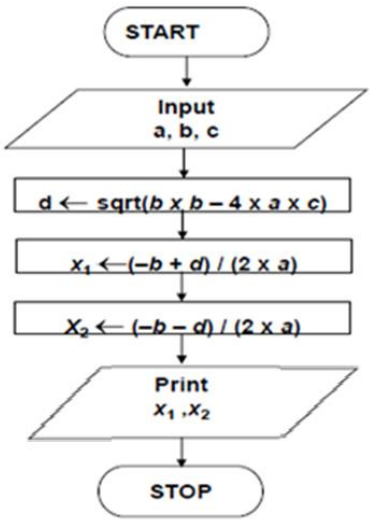
$$\mathbf{d} = \sqrt{b^2 - 4ac}\text{ , and the roots are:}$$

$$\mathbf{x1} = \frac{(-b + d)}{2a} \text{ and } \mathbf{x2} = \frac{(-b - d)}{2a}$$

Algorithm:

- Step 1: Input a, b, c
- Step 2: d = sqrt (b*b – 4 * a * c)

- Step 3: x1 = (–b + d) / (2 * a)
- Step 4: x2 = (–b – d) / (2 * a)
- Step 5: Print x1, x2
- Step 6: Stop

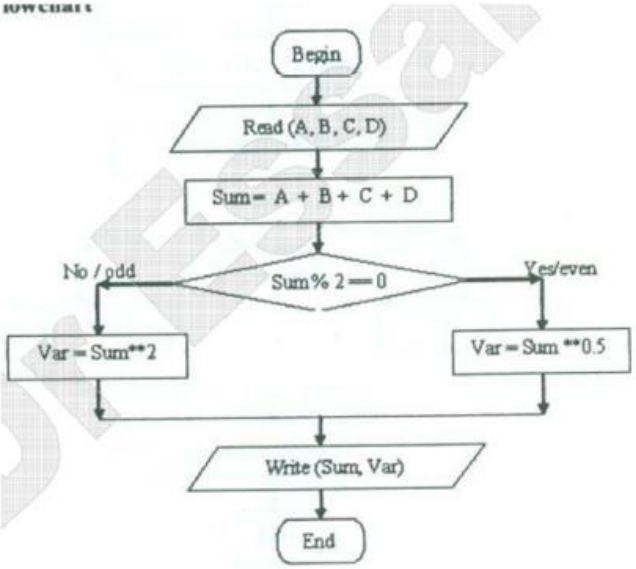


Task

أوجد المخطط الانسيابي (Flowchart) وكود الشفرة (Pseudo code) لخوارزم يقرأ أربعة أرقام (A, B, C, and D) ثم يقوم بإيجاد مجموع هذه الأرقام (Sum) . إذا كان المجموع رقم زوجي فان قيمة المتغير (Var) تساوي الجذر التربيعي للمجموع وإذا كان المجموع فردي فان قيمة المتغير (Var) تساوي مربع المجموع . الخوارزم يقوم بطباعة المجموع (Sum) وقيمة المتغير (Var)

Pseudo code

```
Variable A, B, C, D, Sum : integer
Variable Var: real
Begin
Read (A, B, C, D)
Sum = A + B + C + D
If (Sum Mod 2 == 0) then
        Var = Sum ** 0.5
Else
        Var = Sum ** 2
End if
Write (Sum, Var)
End
```

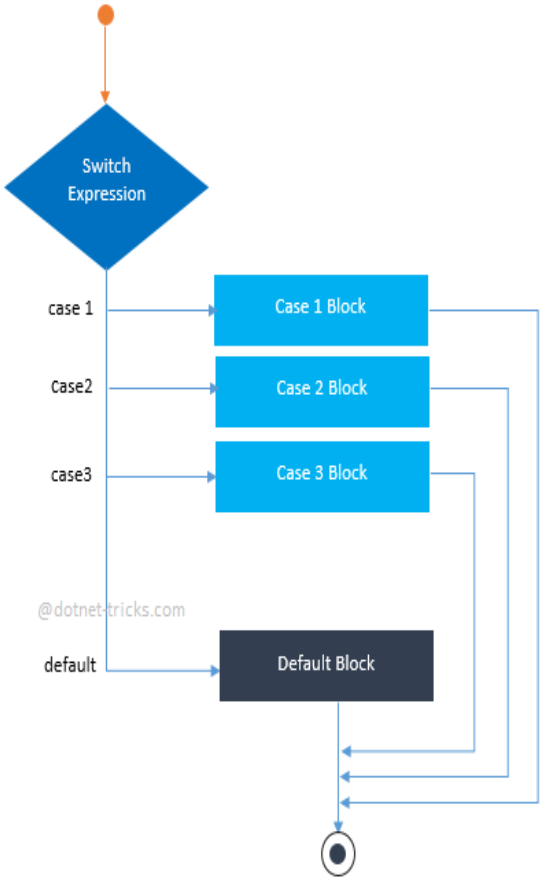


Switch Statement

Switch statement acts as a substitute for a long if-else-if that is used to test a list of cases. A switch statement contains one or more case labels which are tested against the switch expression. When the expression match to a case then the associated statements with that case would be executed.

Syntax

```
Switch (expression)

{

 case value1:

     //Statements

    break;

 case value 2:

     //Statements

      break;

 case value 3:

     //Statements

 case value n:

     //Statements

     break;

 Default:

    //Statements

}
```



Switch Statement

Example 14

Write algorithm which read the numbers from 1 to 7 and display their correspondence day of week.

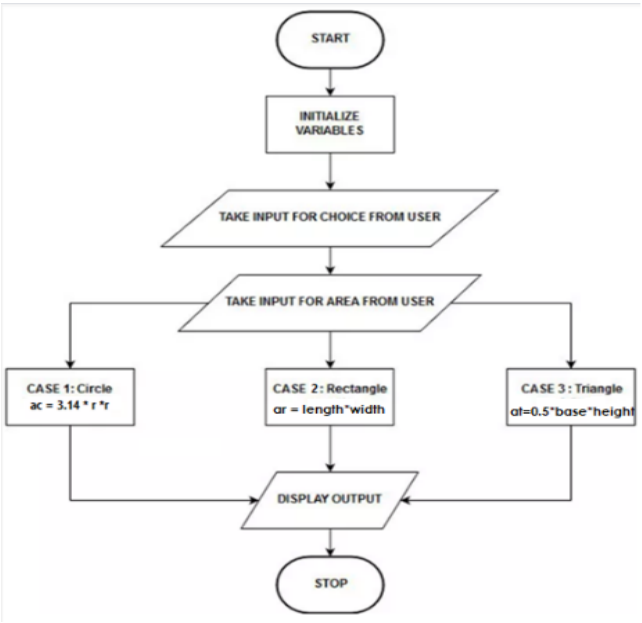
Step 1: Start
Step2: read integer number say (day)
Step3: Switch (day)

```
{
   Case 1 :
            Print “Saturday“
            break;
   Case 2 :
            Print “Sunday“
            break;
   Case 3 :
            Print “Monday“
            break;
    Case 4 :
          Print “Tuesday“
            break;
    Case 5 :
            Print “Wednesday“
            break;
Case 6 :
            Print “Thursday“
            break;
   Case 7 :
            Print “Friday“
            break;
  Default :
          Print “Invalid day“
          break;
  }
```

Step4: stop

Example 15

Algorithm to calculate the area of A rectangle or circle or triangle by taking the user's choice.



Step 1: Start
Step 2: Initialize variables
Step 3: Take input for choice and then for area variables from the user
Step 4: Case 1: ac = 3.14*r*r
break;
Case 2: ar = length*width
break;
Case 3: at =0.5*base*height
break;
Step 5: Display output according to case
Step 6: Stop

Example 16

Write algorithm that read a word English vocabulary from user as string and print its definition, at case of word not found print "The vocabulary is not found in the database".

Note: solve using if statement then try to solve using select case statement

word vocabulary	Definition
Pillow	Is soft or hard thing that you put your head on it.
Pillowcase	Is a cover for the pillow
Dresser	Is a piece of furniture that you put your clothes into

Algorithm

Step 1: Start
Step 2: Initialize variable (word) as string
Step 2: Read word
Step 3: if (word == "Pillow")
Print (Is soft or hard thing that you put your head on it)
else if (word == "Pillowcase")
Print (Is a cover for the pillow)
else if (word == "Dresser")
Print (Is a piece of furniture that you put your clothes into)

else
Print (The vocabulary is not found in the database)
Step 4: stop

Problem solving with loops

A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages.

Types of Loop Structures

1. **Automatic Counter Looping (ACL)**
   - For– Counter Loop
2. **Conditioned Looping (CL)**
   - While Loop
   - Do-While Loop
   - Repeat-Until Loop

For loop

- A FOR loop is a loop that repeats a specified number of times. The loop uses a counter to tell it how many times to run the same sequence of activities.
- **Syntax**

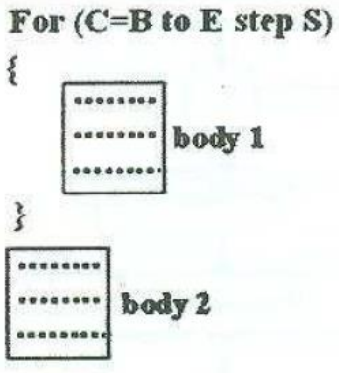
LOOP: COUNTER = BEGIN TO END STEP S

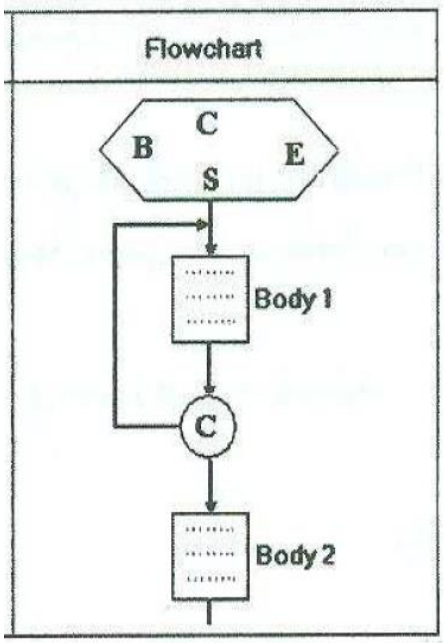
INSTRUCTION

INSTRUCTION

…… . . .

LOOP-END: COUNTE



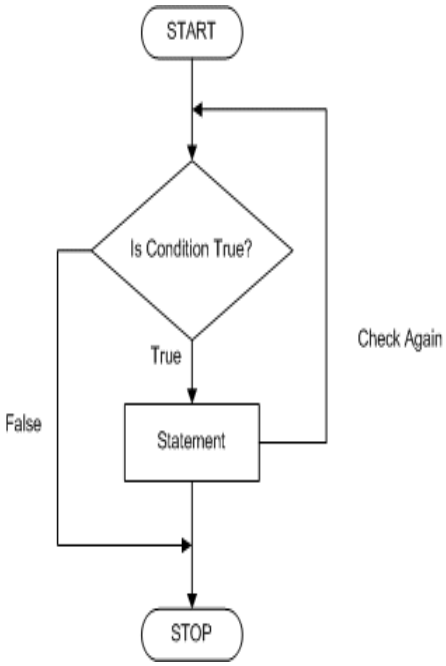


While Loop

This type of loop tells the computer that while the condition is TRUE, repeat all instructions between the WHILE and the WHILE END.

Syntax:

WHILE condition
    Instructions
END-WHILE



The statement is executed as long as the condition is True. The loop terminates when the condition is False.

Repeat/Until Loop

- This type of loop tells the computer to repeat the set of instructions between the REPEAT and until, until a condition is TRUE.
- Syntax

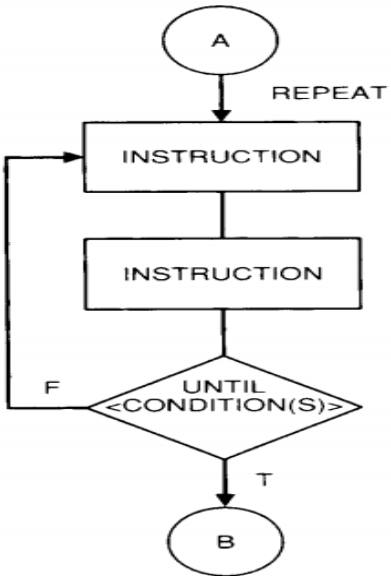
Repeat

Instruction

Instruction

. . .

Until <condition>



Do-While Loop

- Like a while loop, a do-while loop is a loop that repeats while some condition is satisfied.
- Unlike a while loop, a do-while loop tests its condition at the end of the loop.
- This means that its sequence of activities always runs at least once.

➨ **Syntax**

Do
Statement
WHILE (condition)

***Repeat until* loop, continues to run *until* the control expression is true (and then terminates).**

Do while loop runs *while* the control expression is true (and terminates once the expression becomes false).

Example 17

Create the algorithm and the flowchart to Print Hello World 10 times.

Algorithm (using While loop)

Step 1: Start

Step 2: Initialize count = 0

Step 3: While (count < 10)

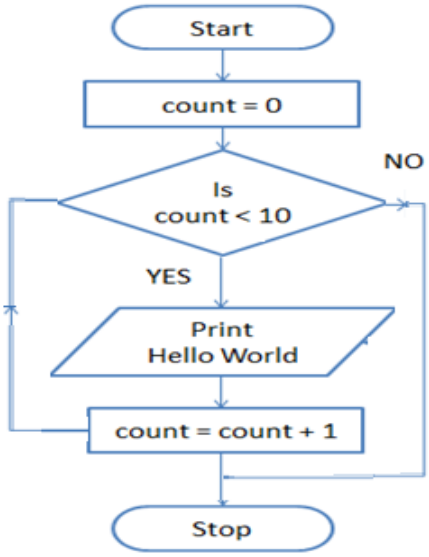
Print (Hello World)

Increment count by 1

While- end

Step 4: stop

Flowchart



Algorithm (using Repeat until)

Step 1: Initialize count = 0

Step 2: Repeat

Print (Hello World)

count = count +1

Until (count = 10)

Step 3: stop

Algorithm (using do- while)

Step 1: Initialize count = 0

Step 2: do

Print (Hello World)

count = count +1

While (count < 10)

Step 3: stop

Algorithm (using For loop)

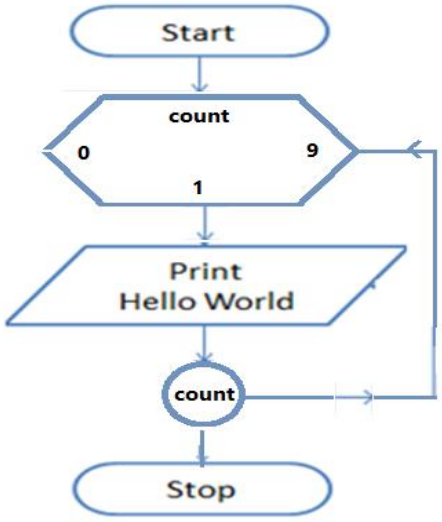
Step 1: Loop: (count = 0 to 9)

Step 2 : Print (Hello World)

Step 3: Loop-end

Step 4: Stop

Flowchart



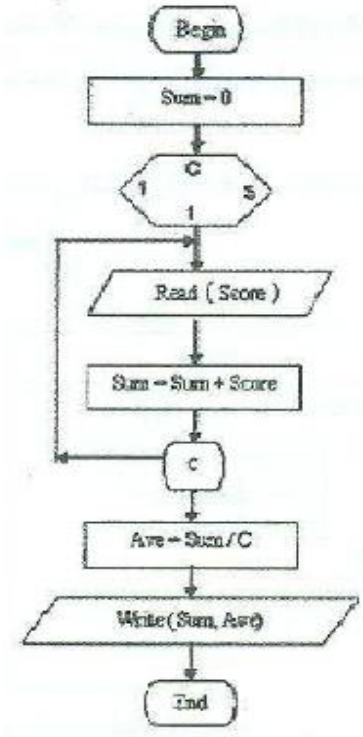
Example 18

مثال:

أوجد المخطط الانسيابي (Flowchart) بالإضافة إلى كود الشفرة (Pseudo code) لخوارزم يقوم بقراءة درجات طالب في خمس مواد دراسية ثم يقوم بحساب وطباعة مجموع درجات الطالب وكذلك متوسط درجاته

Pseudo code

Variable C, Sum=0, Score: integer
Variable Ave : real
Begin
Loop: (C = 1 to 5)
    Read (Score)
    Sum = Sum + Score
Loop-end: C
Ave = Sum / C
Write (Sum, Ave)
End



Result of the algorithm

Input : Score	50	70	90	60	80
Output : Sum, Ave	350 70				

Task

Find Even numbers between 1 to 50.

Algorithm

Step-1 Start

Step-2 I = 1

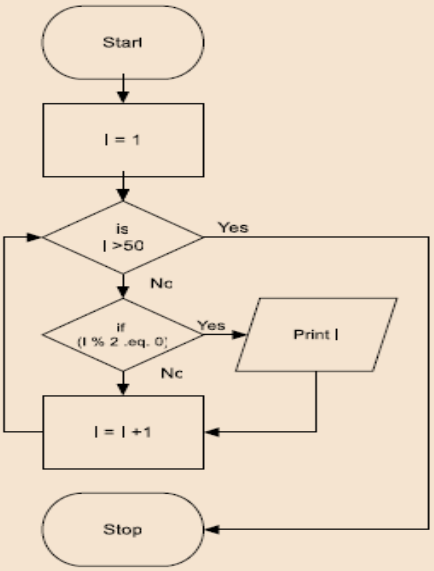
Step-3 IF (I >50) THEN
GO TO Step-7
ENDIF

Step-4 IF ((I % 2) =0) THEN
Display I
ENDIF

Step-5 I = I + 1

Step-6 GO TO Step--3

Step-7 Stop



Task

حساب المعدل التراكمي GPA للطالب يتم بالتالي:

إذا كان التقدير (grade = A) زاد GPA بمقدار 0.9

إذا كان التقدير (grade = B) زاد GPA بمقدار 0.7

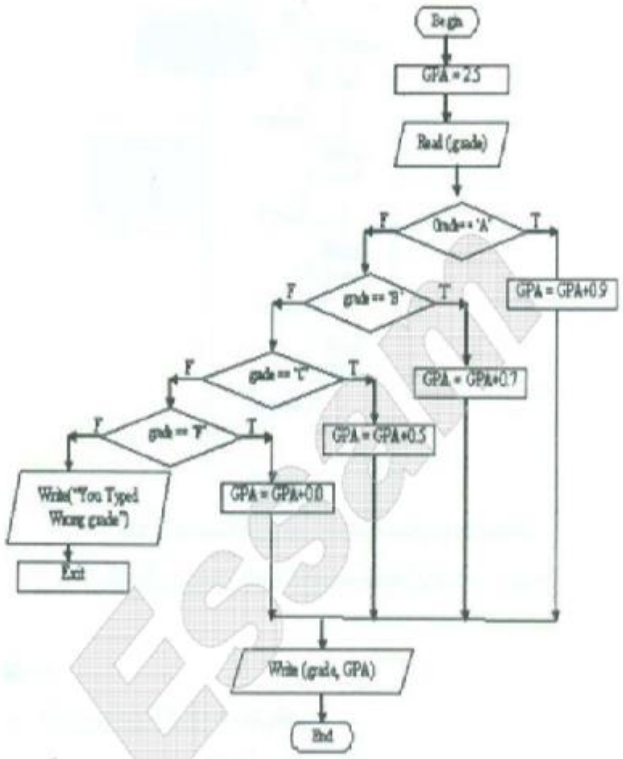
إذا كان التقدير (grade = C) زاد GPA بمقدار 0.5

إذا كان التقدير (grade = F) زاد GPA بمقدار 0.0

أوجد المخطط الانسيابي (flowchart) بالإضافة إلى كود الشفرة (Pseudo code) لخوارزم يقوم بحساب المعدل التراكمي GPA للطالب في ماده معينه. إذا تم إدخال تقدير (GPA) خاطئ فان الخوارزم ينتهي مع طباعة رسالة "You typed wrong grade" . الخوارزم يطبع قيمه كل من grade and GPA . يمكن اعتبار وجود قيمه أوليه (GPA = 2.5)

Pseudo code

```
Variable GPA=2.5 : real
Variable grade: character
Begin
Read (grade)
Switch (grade)
        case: "A"
                GPA = GPA + 0.9
        Break
        case: "B"
                GPA = GPA + 0.7
        Break
case: "C"
                GPA = GPA + 0.5
        Break
        case: "F"
                GPA = GPA + 0.0
        Break
        Default
                Write ("You Typed Wrong Grade ")
                Exit
Switch-end
Write (grade, GPA)
End
```



Example 19

باستخدام While structure أوجد Pseudo code لخوارزم يقوم بطباعة الأعداد من 1 إلى 100

```
Variable Num=1 : integer
Begin
While (Num <=100)
        Write (Num)
        Num = Num + 1
While-end
End
```

Example 20

Algorithm & Flowchart to find Odd numbers between 1 to 100.

ALGORITHM:

Step 1: Start

Step 2: Declare variable N as integer type

Step 3: Set N = 1

Step 4: while (N <= 100)

Step 5: if (N % 2 != 0) then

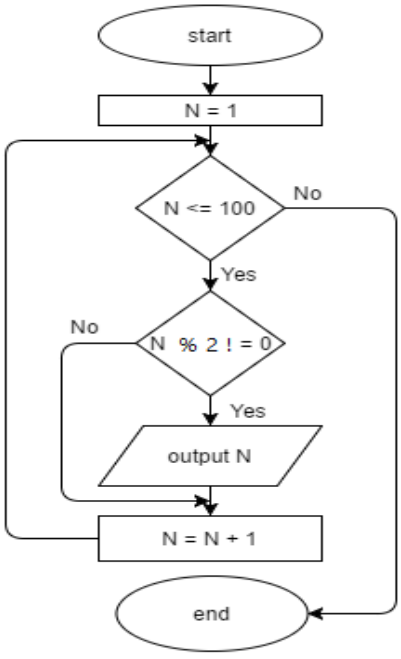
Step 6: print N

Step 7: End if

Step 8: N = N + 1

Step 9: End while

Step 10: Stop



Example 21

Algorithm & Flowchart to find sum of series 1+2+3+…..+N

Algorithm

Step-1 Start

Step-2 Input Value of N

Step-3 Initialize SUM = 0 , i = 1

Step-4 while (i <= N)

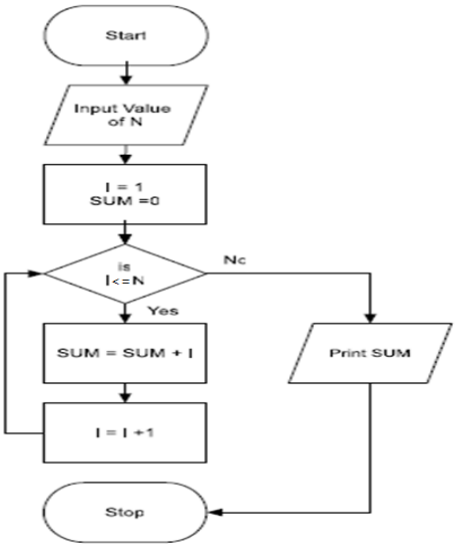
Step-5 SUM = SUM + i

Step-6 i = i + 1

Step-7 End while

Step-8 Display value of SUM

Step-9 Stop



Example 22

Algorithm & Flowchart to find Factorial of number n (n!=1x2x3x…n)

Algorithm (Using While loop)

Step-1 Start

Step-2 Read number N

Step-3 FACT = 1 , i =1

Step-4 WHILE (i <= N)

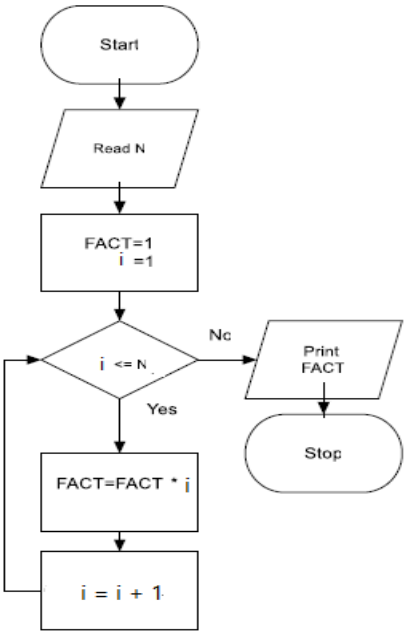
FACT = FACT * i

i = i + 1

End While

Step-8 Display FACT

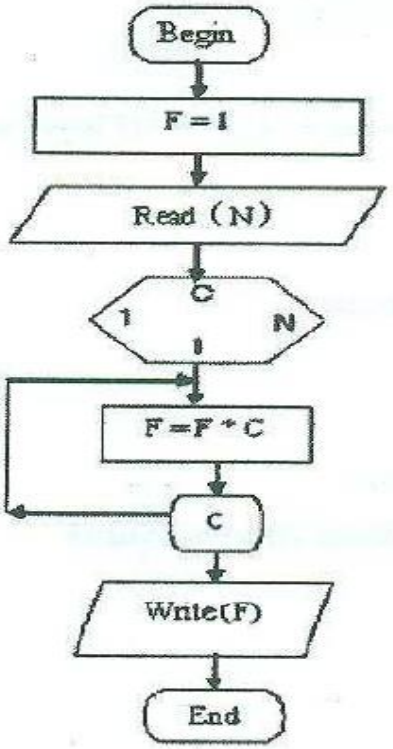
Step-9 Stop



Using For Loop

Pseudo code

```
Variable C, F=1, N: integer
Begin
Read (N)
Loop :( C = 1 to N)
        F = F * C
Loop-end: C
Write (F)
End
```



Example 23

Write an algorithm and draw a Flowchart to read in two numbers, x and n, and then compute the sum of this geometric progression:

$$1+x+x^2+x^3+.....+x^n$$

For example: if n is 3 and x is 5, then the program computes 1+5+25+125.

Algorithm

Step-1 Start

Step-2 Read numbers N, X

Step-3 SUM = 1, TERM =1, i = 1

Step-4 WHILE (i <= N)

TERM = TERM * x

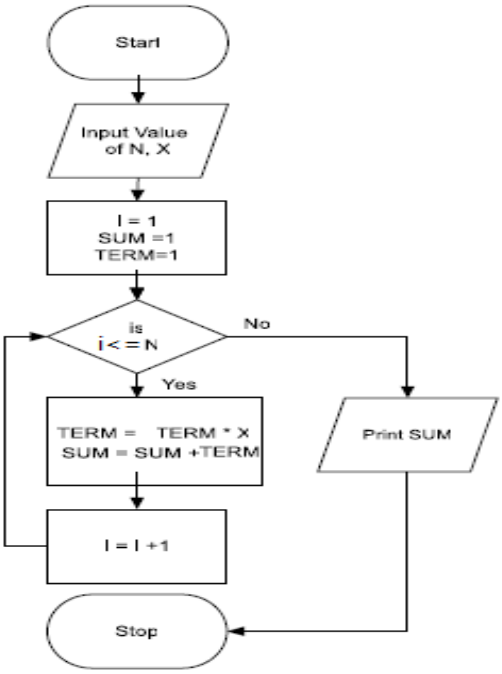
SUM = SUM + TERM

i = i + 1

End While

Step-8 Display SUM

Step-9 Stop



Example 24

Algorithm & Flowchart to print multiplication Table of a number

Algorithm (Using While loop)

Step-1 Start

Step-2 Input Value of NUM

Step-3 i =1

Step-4 While (i <= 12)

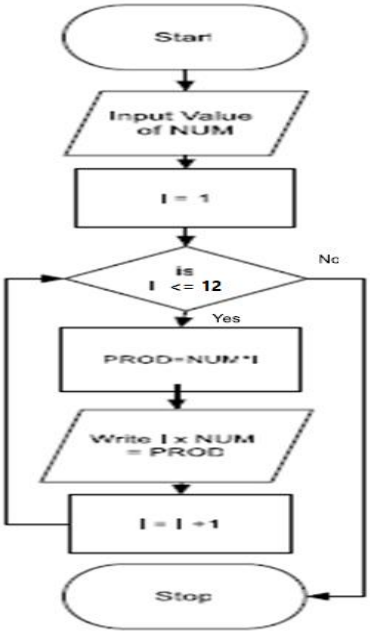
PROD= NUM * i

Write i "x" NUM "=" PROD

i = i + 1

End While

Step-5 Stop



Algorithm (Using For loop)

Step-1 Start

Step-2 Input Value of NUM

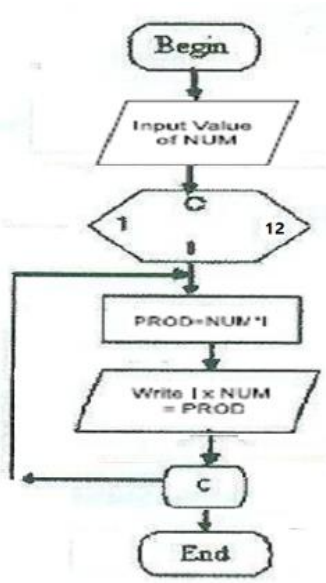
Step-3 For (i = 1 to 12)

PROD= NUM * i

Write i "x" NUM "=" PROD

End For

Step-4 Stop



Algorithm & Flowchart to print multiplication Table (using nested loop)

Pseudo code

Variable C1, C2, Prod : integer

Begin

Loop:(C1=1 to 10)

Loop:(C2 = C1 to 10)

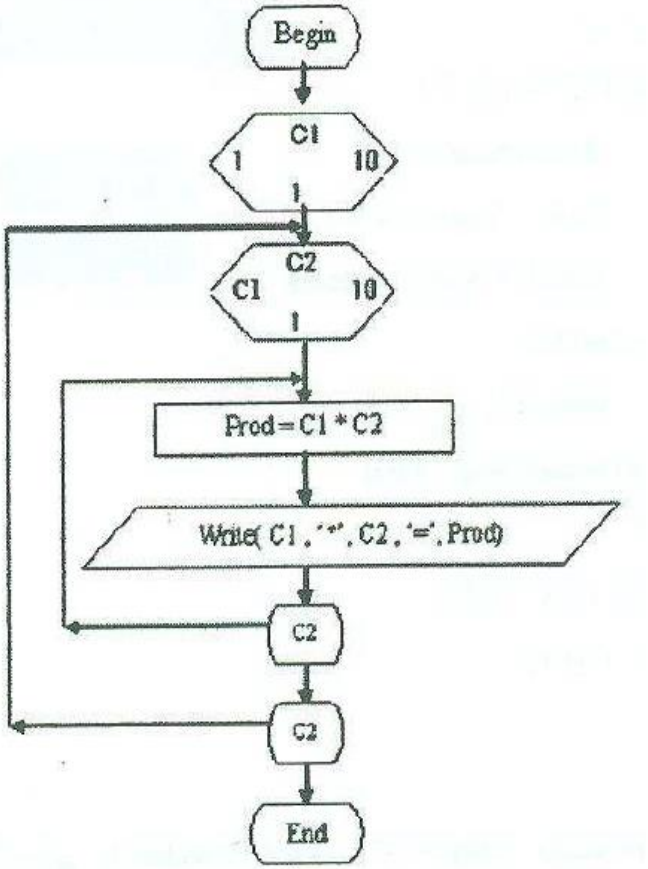
Prod = C1 * C2

Write (C1 , '*', C2, '=', Prod)

Loop-end:C2

Loop-end:C1

End



Example 25

Algorithm & Flowchart to generate Fibonacci series 0,1,1,2,3,5…,n

Algorithm

Step-1 Start

Step-2 Initialize the variables, next, A=0, B=1, Count =2

Step-2 Read number N

Step-3 Print (A, B)

Step-4 While (Count < N)

next = A + B

print (next)

A = B

B = next

Count = Count + 1

End While

Step-5 Stop