Al-Ma'moun University College
Department of Engineering of Medical instrumentation Techniques
Second Class

Chapter Two

Algorithm and Flowchart

M.SC Halah Adnan

Algorithms and Flowcharts

Introduction

The program is a set of instruction gave to the computer to execute successive operations leads to solve specific problem. In general to solve any problem in computer we must follow these steps:

1. Analyze the problem.
2. Write an Algorithm.
3. Draw flowchart.
4. Convert the flowchart to program.
5. Run the program and test the solution.

The first step is depends on how the programmer is thinking to solve the problem which may be different from one person to another.

The second and third steps are how to translate this idea to the paper this can be explained in this chapter. The fourth step is to convert this idea from paper to computer by using any programming language you.

Also you can see the result after executing the written program after running it.

1. Algorithm

In general, the algorithm can be defined as: "combination of phrases and events that can be arranged as steps to solve a specific problem". That can be done by understanding this problem whether it mathematic or logic before convert it to flowchart.

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

2. How to write Algorithms:

Step 1: Start.

Step 2: 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 length and rectangle width. The area of a rectangle = LENGTH × WIDTH. We can define two variables for rectangle length and rectangle width as LENGTH and WIDTH.

Step 3: To calculate the area of a rectangle, we multiply the LENGTH by the WIDTH. Output will be the value stored in variable AREA. If the input variables described a rectangle with a LENGTH = 2 and a WIDTH = 3, the algorithm would output the value of AREA =6.

Step 4: Print the output.

Step 5: End.

Thus an algorithm is:

```
1. Start
2. Input LENGTH, WIDTH
3. Find AREA = LENGTH * WIDTH
4. Print AREA
5. End
```

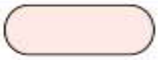
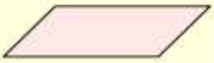
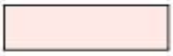
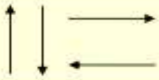
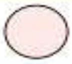
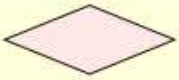
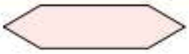
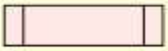
Example: Write an algorithm to read five numbers then find and print their summation.

Algorithm is:

```
1. Start
2. Read L, M, N, O, P
3. Find Sum (S) , S =L+M+N+O+P
4. Print S
5. End
```

3. Flowchart:

Flowchart is diagrammatic /Graphical representation of sequence of steps to solve a problem. Flowcharts consist of a shapes connected by a straight lines. The following table shows these shapes and their operations.

Shape	Operation
	Start or End
	Input / Output data • Read or Input • Print
	Processing / Storing • Addition(+), Subtraction(-), Multiplication(*), Division(/), Expontiation(^), ... • Store a value (Put)
	Flow Lines
	Connection
	Decision • If statement • Question (?)
	Looping / Counters
	Subroutine

4. The Advantage of using algorithms and flowcharts:

1. To show the mathematical logic used to solve problems.
2. To show how the data processing is done.
3. Helps the programmer to write his program.
4. Divides the big problem to smaller parts.
5. To avoid the errors that occurred during writing the program.
6. It is a middle step between problem difficulty and its conversion to suitable program.
7. Easy to convert it to any programming language.

5. Type of Algorithms and flowchart:

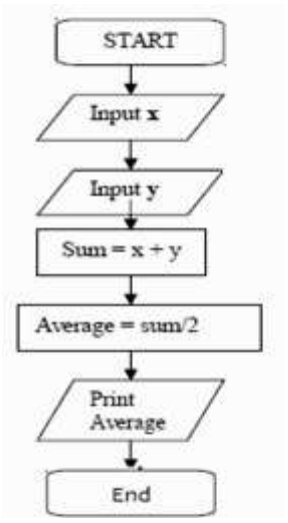
The algorithm and flowchart, classification to the three types of *control structures*. They are:

1. Sequence.
2. Branching (Selection).
3. Loop (Repeating).

These three control structures are sufficient for all purposes.

1. **The sequence** is exemplified by sequence of statements place one after the other – the one above or before another gets executed first. In flowcharts, sequence of statements is usually contained in the rectangular process box.

Example: Write an algorithm and draw the flowchart to find the average of two numbers.

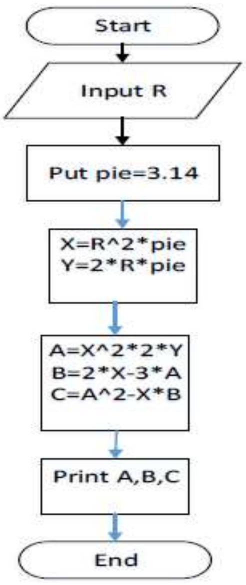
Algorithm:	Flowchart:
1. Start 2. Input x, Input y 3. Sum = x + y 4. Average = Sum /2 5. Print Average 6. End	 START Input x Input y Sum = x + y Average = sum/2 Print Average End

Example : Write an algorithm and draw a flowchart to find the value of A, B, C from the following equations:

$$A = X^2+2Y \ , \ B = 2X-3A \ , \ C = A^2-XB$$

Where X is the circle area and Y is the circumference. Input the radius (R) and print the value of A, B and C.

Algorithm:	Flowchart:

1. Start
2. Input Radius (R)
3. pie = 3.14
4. Area (X), X= R^2*pie
5. Circumference(Y),Y= 2*R*pie
6. A= X^2+2*Y
7. B=2*X-3*A
8. C=A^2-X*B
9. Print A, B, C
10. End
	 Start
	Input R
	Put pie=3.14
	X=R^2*pie
	Y=2*R*pie
	A=X^2*2*Y
	B=2*X-3*A
	C=A^2-X*B
	Print A,B,C
	End

2. **The branch** refers to a binary decision based on some condition. If the condition is true, one of the two branches is explored; if the condition is false, the other alternative is taken. This is usually represented by the **'if-then'** & **'Switch Statement'** construct in programs. In flowcharts, this is represented by the diamond-shaped decision box.

2.1 'IF-ELSE' Statement

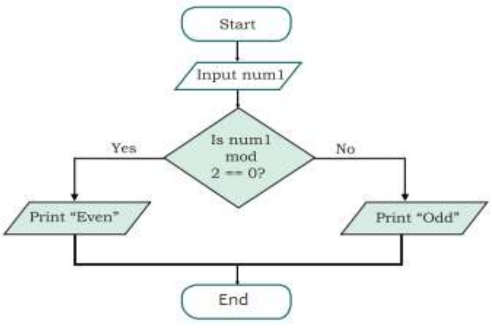
The if-else statement is used to execute the code if condition is true or false. It is also called two-way selection statement.

Example: write an algorithm and draw a flowchart to check number is odd or even?

Algorithm:

```
1. Start
2. INPUT number
3. IF number mod 2 == 0   THEN
              PRINT "Number is Even"
       ELSE
              PRINT "Number is Odd"
4. End
```

Flowchart:

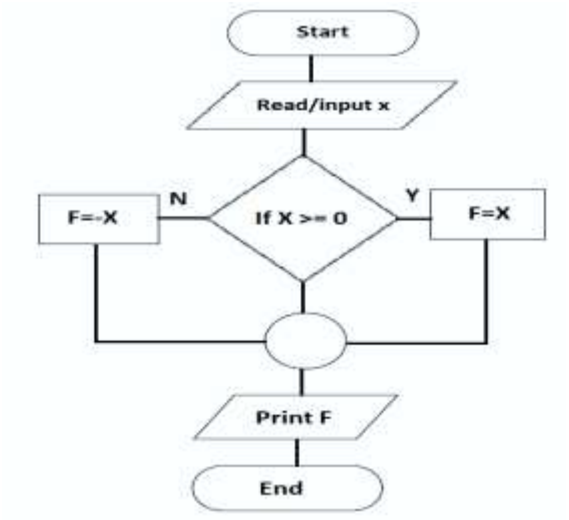


Example: Write algorithm and draw flowchart to find the result of equation:

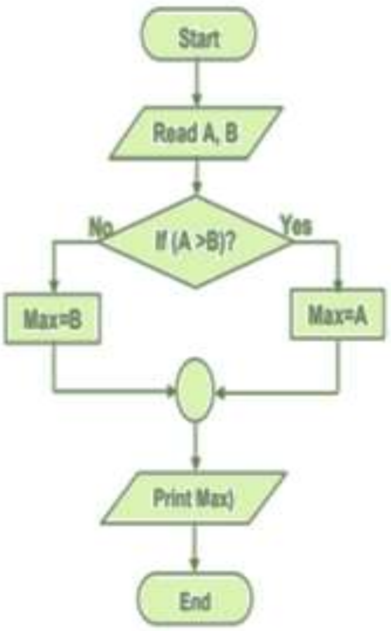
$$f(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

```
Algorithm:
  1. Start
  2. Read/input x
  3. If X >=0 then
                 F=X
          Else
                 F=-X
     End if
  4. Print F
  5. End
```

Flowchart:



Example: Write an algorithm and draw a flowchart to find and print the maximum number between two numbers.

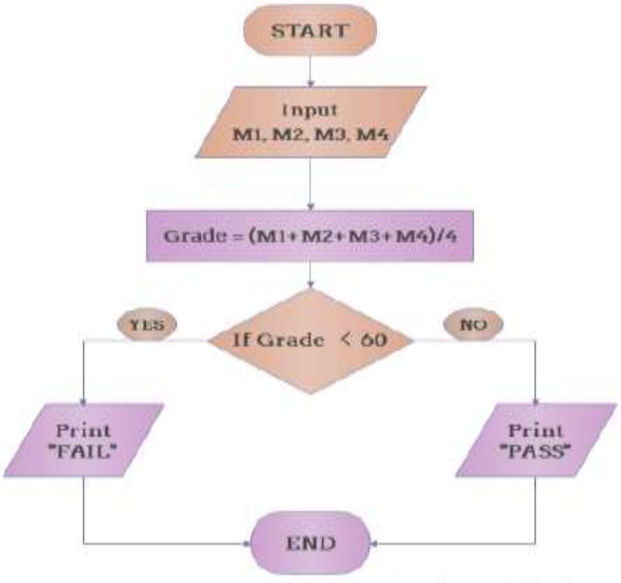
Algorithm:	Flowchart:
1. Start 2. Read A , B 3. If (A > B) then maximum(Max=A) Else maximum(Max=B) 4. Print Max 5. End	 Start Read A, B No If (A >B)? Yes Max=B Max=A Print Max) End

Example: 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:

```
1. Input M1, M2, M3, M4
2. GRADE = (M1+M2+M3+M4)/4
3. if (GRADE < 50) then
                          Print "FAIL"
        Else
             Print "PASS"
     End if
4. End
```

Flowchart:



2.2 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.

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

```
1. Start
2. Read number (day)
3. 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;
4. End
```

3. **The loop** allows a statement or a sequence of statements to be repeatedly executed based on some loop condition. It is represented by the 'while' and 'for' constructs in most programming languages, for unbounded loops and bounded loops respectively. (Unbounded loops refer to those whose number of iterations depends on the eventuality that the termination condition is satisfied; bounded loops refer to those whose number of iterations is known before-hand.) In the flowcharts, a back arrow hints the presence of a loop. A trip around the loop is known as iteration. You must ensure that the condition for the termination of the looping must be satisfied after some finite number of iterations, otherwise it ends up as an infinite loop, a common mistake made by inexperienced programmers.

Example: Write an Algorithm to find Factorial of number N (N!=1x2x3x…N) (Using While , for loop)

while

```
1. Start
2. Read number N
3. FACT=1: i=1
4. WHILE (i <= N)
     DO
        FACT=FACT*i
        i=i+1
     End
5. Print FACT
6. End
```

for

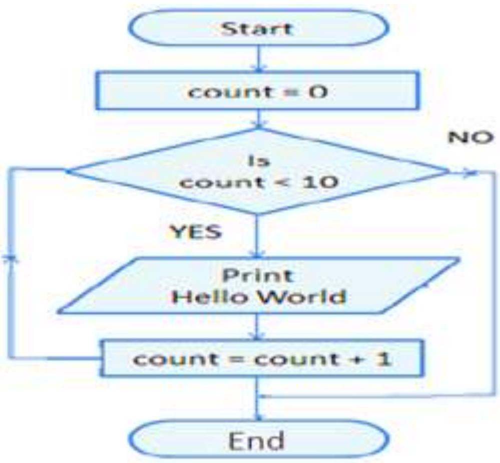
```
1. Start
2. Read number N
3. FACT=1
4. for i= 1 to n
          FACT=FACT*i
   Next i
5. Print FACT
6. End
```

Example: Create the algorithm and the flowchart to Print "Hello World" 10 times. (Using While loop)

Algorithm:

```
1. Start
2. count = 0
3. While (count < 10) do
       Print (Hello World)
       count=count+1
   End (While)
4. End
```

Flowchart:

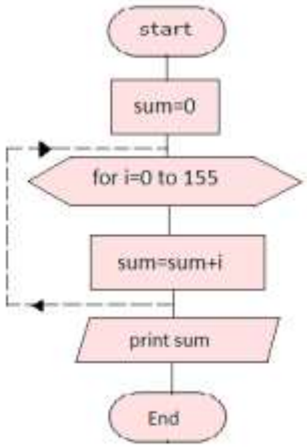


Example: Write an algorithm and draw a flowchart to find the summation of numbers from 0 to 150. (Using for loop)

Algorithm:

```
1. Start
2. sum=0
3. for i=0 to 150
         sum=sum+i
   next i
4. print sum
5. End
```

Flowchart:

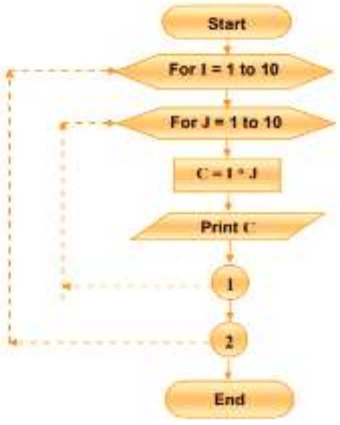


Example Write an algorithm and draw a flowchart to find an print the multiplication table from 1 to 10.

Algorithm:

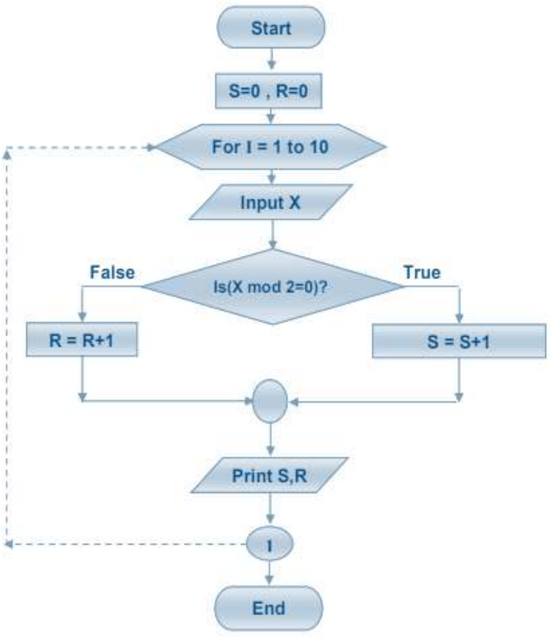
```
1. Start
2. For I=1 to 10
3. For J=1 to 10
4. Find C, C= I * J
5. Print C
6. Next J
7. Next I
8. End
```

Flowchart:



Exercises

1. Write an algorithm and draw a flow chart to calculate 2^4 .
2. Write an algorithm and draw a flowchart to check number is positive or negative.
3. Algorithm & Flowchart to find the smallest of two numbers.
4. Write an algorithm and draw a flowchart to Find Even numbers between 1 to 50.
5. Discuss the flowing flowchart and show its purpose and final results.


Start
S=0 , R=0
For I = 1 to 10
Input X
False
Is(X mod 2=0)?
True
R = R+1
S = S+1
Print S,R
I
End