Chapter 1

Algorithm and Flow Charts

Concept of Programming

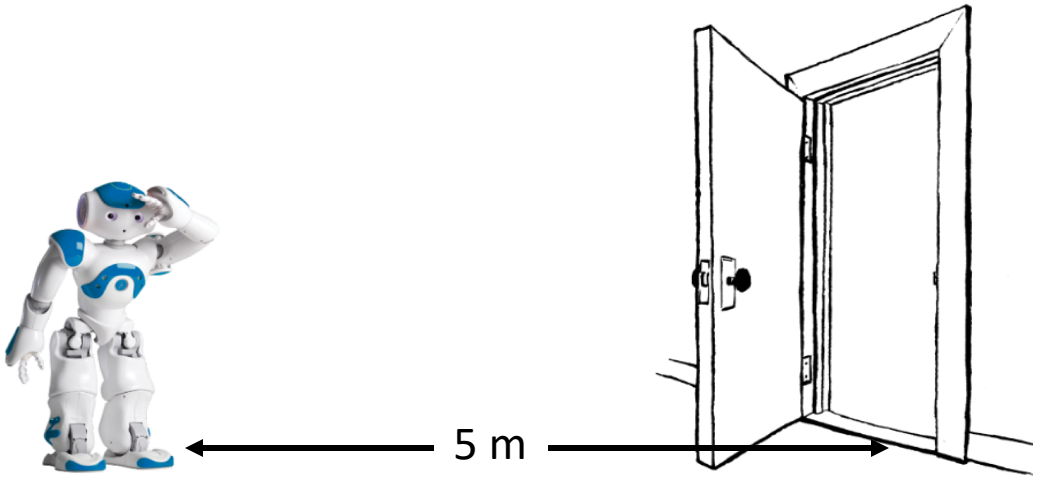
Before we learn how to write a program, we will first learn the concept/the idea of programming. There are 3 States in Developing a program. As good programmer these stages must be followed.

Stages of Developing a Program

1. **Designing the Program:** Decide how to do it. We also call this stage as developing an algorithm.
2. **Implementation Stage of the Algorithm:** In this stage we start writing the program.
3. **Check/Test Stage:** We check/test the program if it is doing what we want by using arbitrary and different cases.

Concept of Programming (Example Algorithm)

Algorithm for a robot to get out of the classroom. Consider a robot placed in a classroom close to the room. Design an algorithm for a program which guides the robot to get out of room. You may use instructions like **start, stop, walk, if the door open, open the door** and **close the door.**



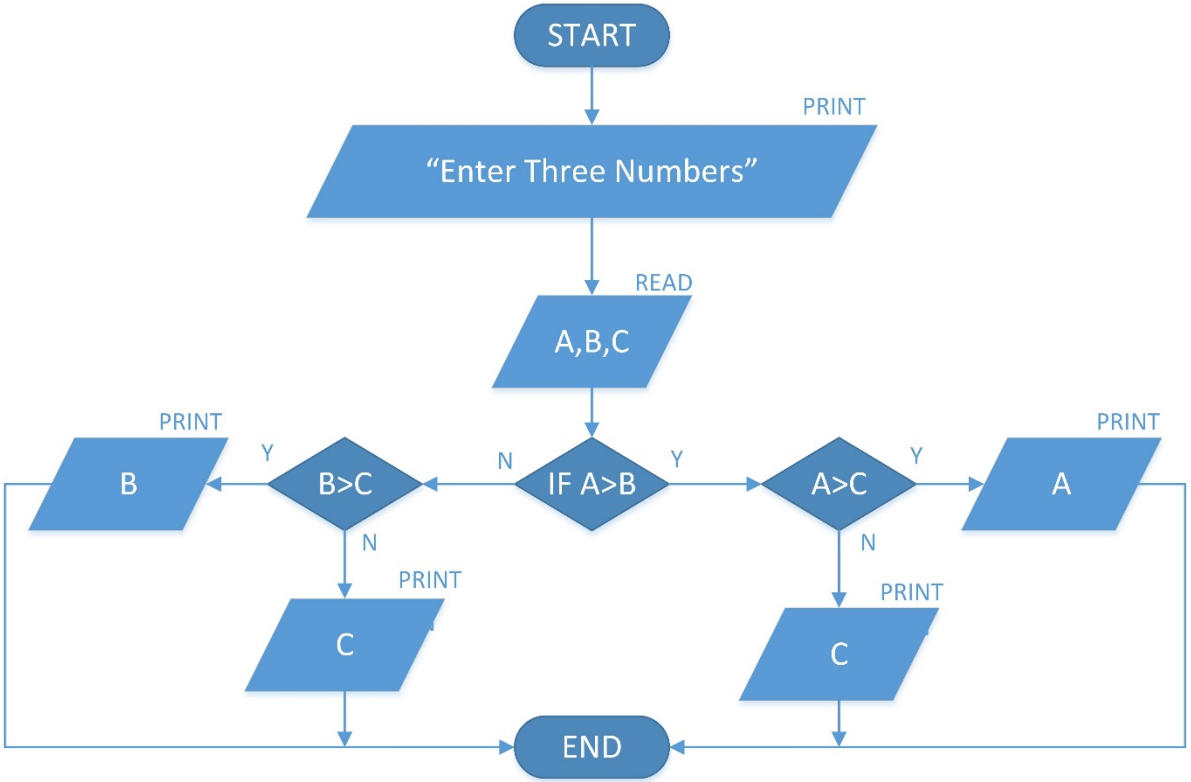
Algorithm

1. Start
2. Walk 5m
3. If the Door is open Walk out, Close the door. Stop
4. Open the Door Walk out, Close the door. Stop

Concept of Programming (Flow Charts)

In the first stage of programming, we design our program. As beginner, it is easier to express the program by using flow-charts. Flow Charts are good way of showing a program visually. After getting experience, you don't need to use the flow charts to begin with.

Below is a flow-chart for a program that prints the largest of three numbers.



Elements in Flow Charts

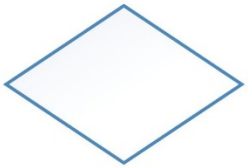
Most commonly used elements in flow charts and some others are shown below.

Commonly Used Elements

Start / End

Computation / Assignment

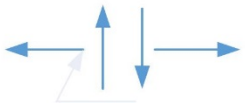
Input / Output Operations



Decision Making and Braching

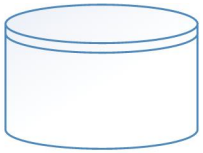
Connector or Joining of Two Part in a program.

Loops / Repetitions



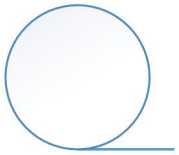
Flow Lines – Shows the direction to proceed.

Rarely Used Elements



Magnetic Disk (Secondary Memory)

Output to Printer



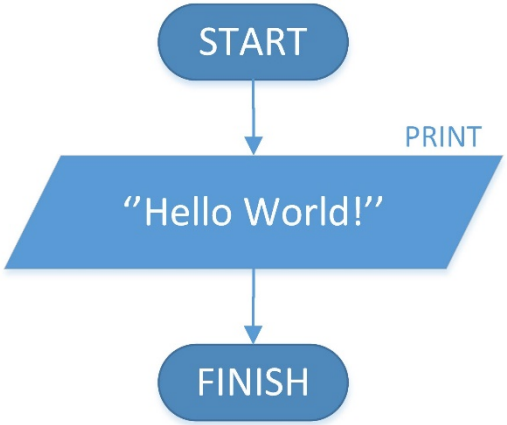
Magnetic Tape (Floppy)

Example 1: Printing (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program prints the “Hello World!” to the screen.**

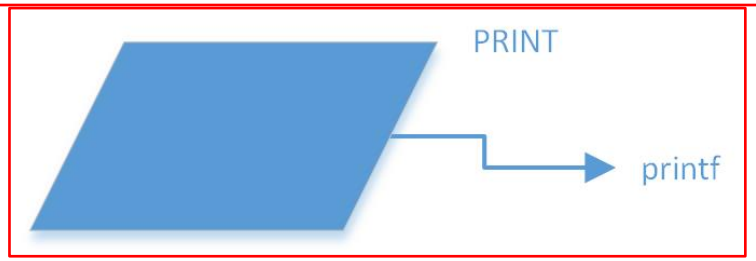
Algorithm

1. Start
2. Print “Hello World!”
3. End



This is the simplest program one can think. When we start writing program, as we will see later there will be a command/statement corresponding each element, so that we will be easily write the program based on the flow-charts.

For example, printing command in C programming language is **printf**



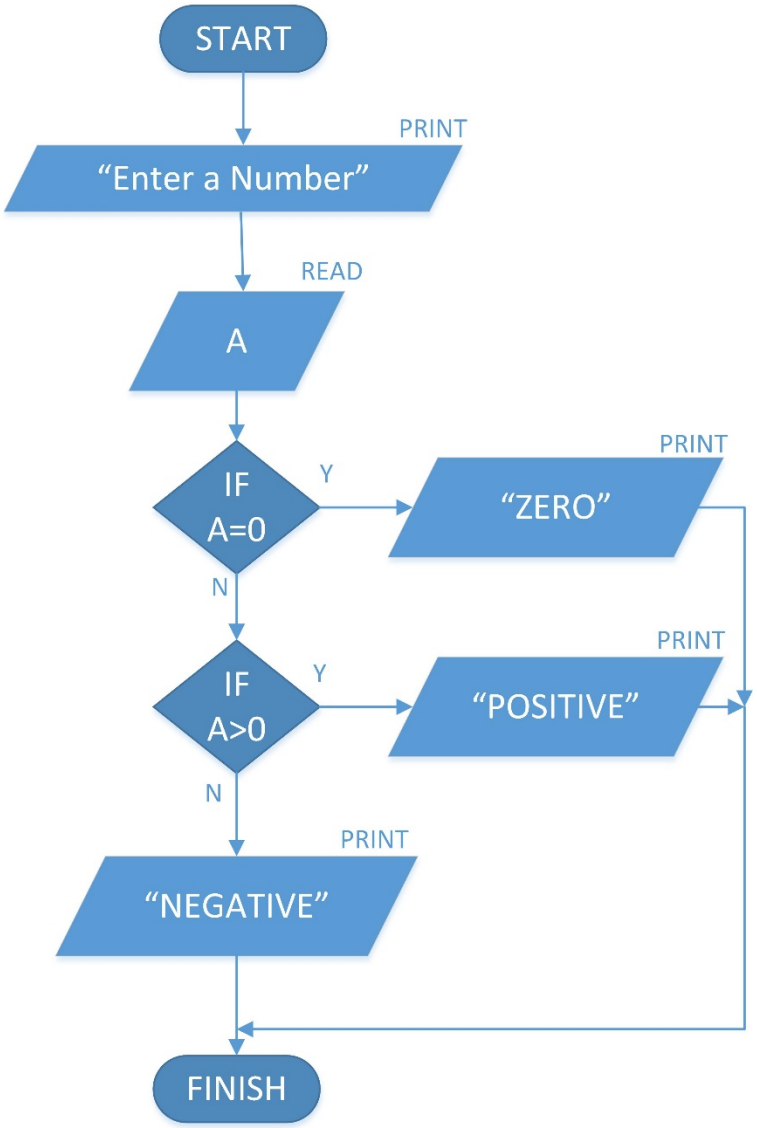
Example 2: ….. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program determines if a given real number is “zero”, “positive”, or “negative” and write this information to the screen. **

Algorithm

1. Start
2. Prompt user to enter a number
3. Read a number and store it in A
4. If A= 0 print “ZERO” then DONE!
5. If A > 0 print “POSITIVE” then DONE!
6. print “NEGATIVE” then DONE!
7. Finish

Now try testing you program by using
10 , 0 and -10



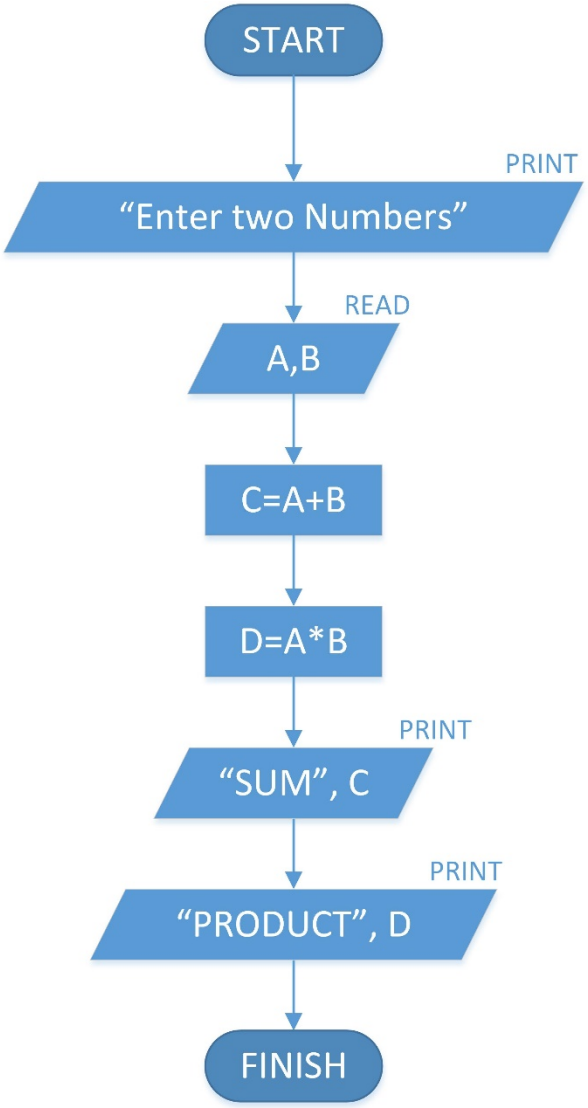
Example 3: ….. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program takes two numbers from keyboard then writes the sum and product of these numbers to the screen. **

Algorithm

1. Start
2. Prompt user to enter two numbers
3. Read two numbers and store them in A, B
4. Define C = A+B
5. Define D = A*B
6. Write C
7. Write D
8. Finish

Now try testing you program by using 3 and 5

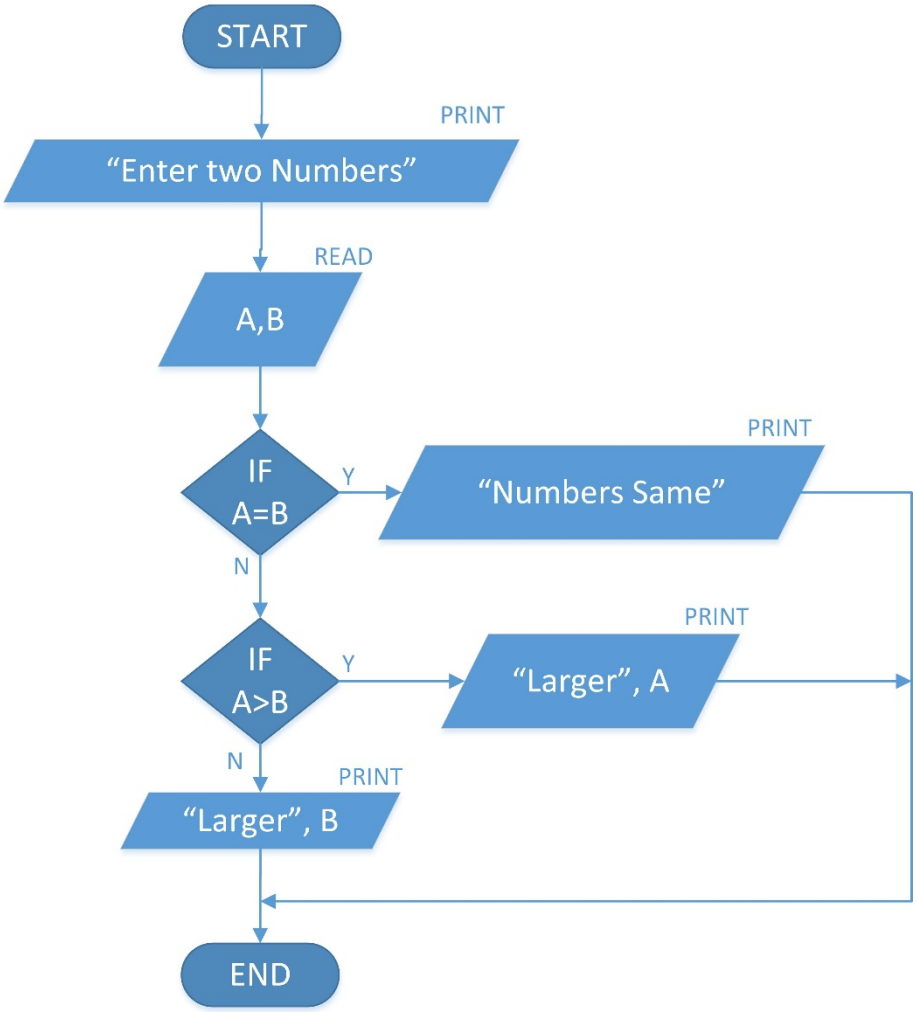


Example 4: ….. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program takes two numbers from keyboard then writes the larger one to the scree. If the numbers are the same then the program prints, “Numbers Same” to the screen. **

Algorithm

1. Start
2. Read two numbers and store them in A, B
3. If A=B then print “Numbers Same” DONE!
4. If A>B then print “Larger”,A . DONE!
5. Print “Larger”,B
6. Finish



Now try testing you program by using

7 and 11

15 and 6

5 and 5

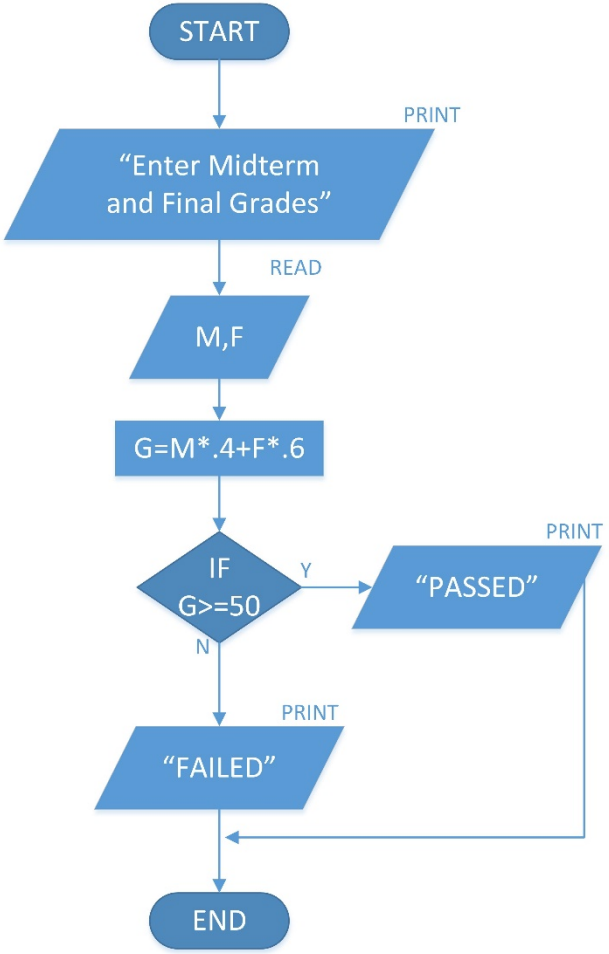
Example 5: ….. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program reads the midterm and final exam results of a student then calculates the final grades as 40% of the midterm and 60% of final exam. Then writes to the screen “PASSED” if the grade is equals or greater than 50 or “FAILED” if grade is less than 50. **

Algorithm

1. Start
2. Read two numbers and store them in M, F
3. Assign/Calculate G = M*.4+F*.6
4. If G ≥ 50 then print “PASSED” , DONE!
5. Print “FAILED”
6. STOP/FINISH/END

Now try testing you program by using
40 and 60
20 and 50



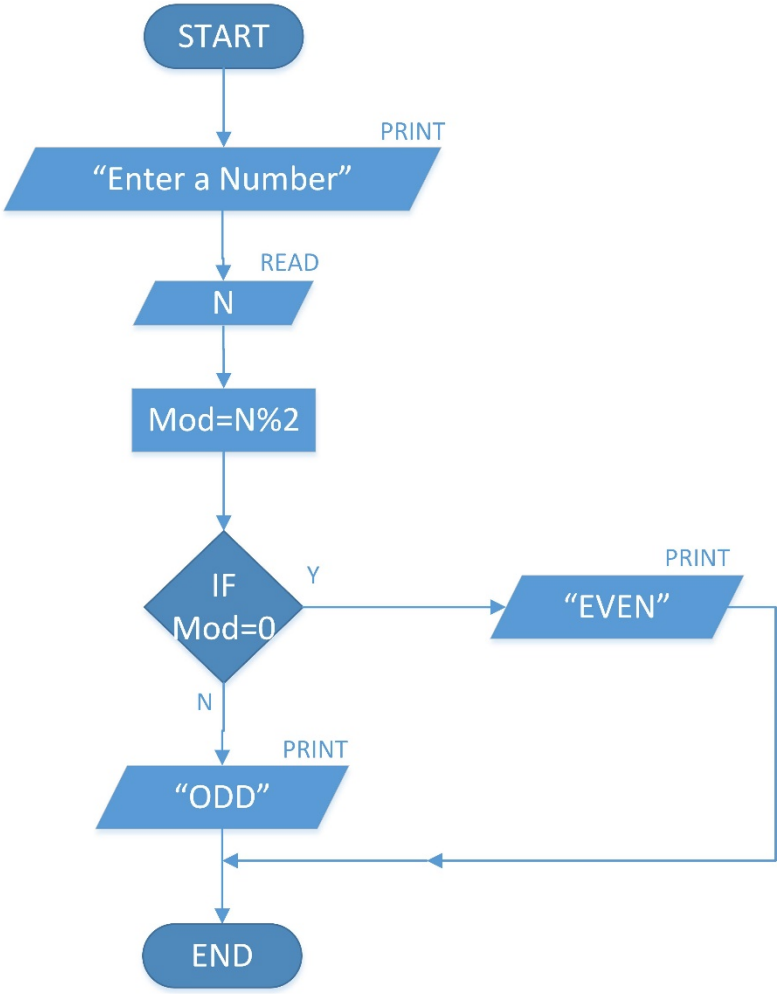
Example 6: …. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program inputs a number from keyboard then prints to screen “EVEN” or “ODD” if the number is even or odd respectively. Not: You may use % modulus / remainder operator. **

Algorithm

1. Start
2. Prompt user to enter a number
3. Read the number N
4. Assign mod = N % 2
5. If mod = 0 then print “EVEN” END!
6. Print “ODD”
7. STOP/FINISH/END

Test your program with inputs 6 and 11.

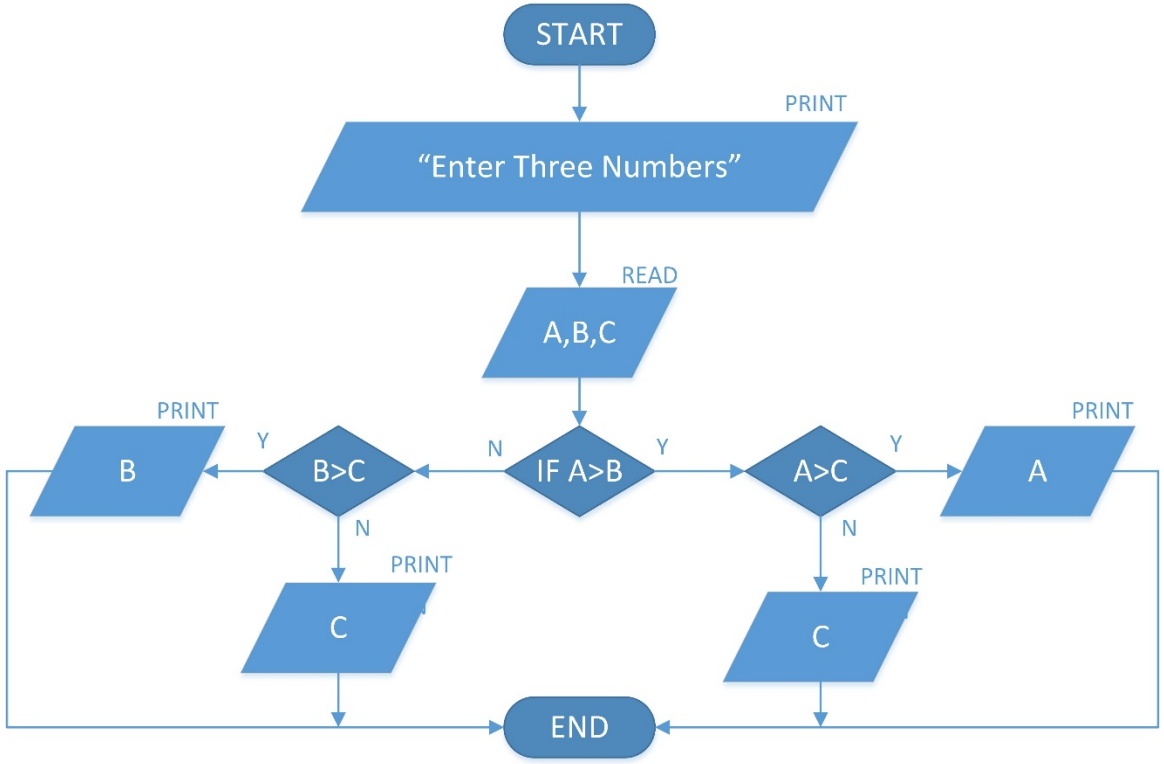


Example 7: ….. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program inputs 3 numbers from the keyboard then prints the largest one to the screen. **

Algorithm

1. Start
2. Prompt user to enter three numbers
3. Read three numbers and store them in A,B,C
4. Compare the first two numbers, A and B
5. Compare the largest of A and B with C then print the largest to the screen
6. STOP/FINISH/END



Now try testing you program by using

5 , 7, and 2

11 , 3, and 21

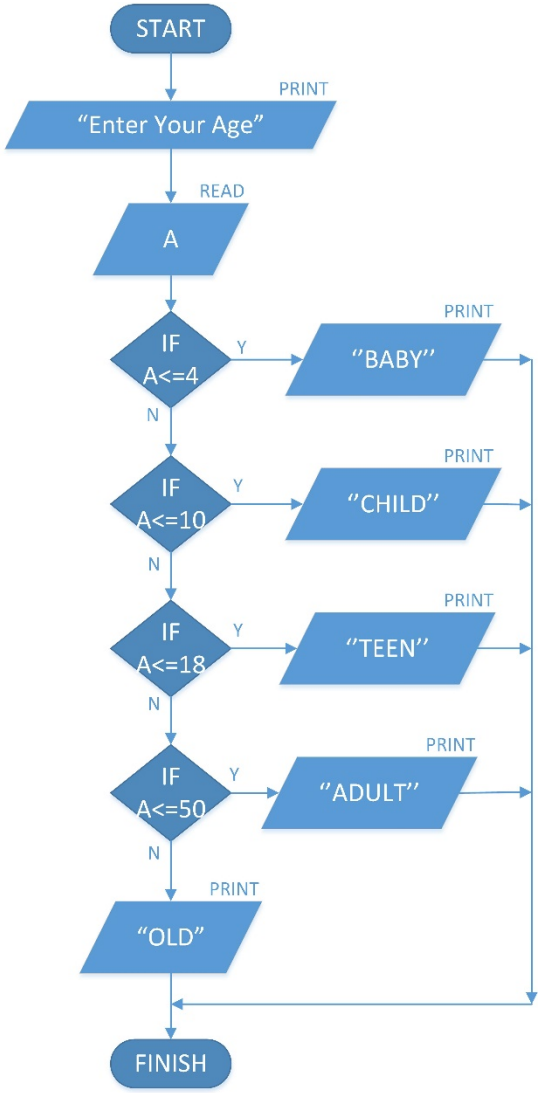
Example 8: ….. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program inputs the age the user then based on the age of the person, the program prints the age status of the person to the screen. **

Algorithm

1. Start
2. Prompt user to enter his/her age.
3. Read the entered number and store it in A variable.
4. If A ≤ 4 print “BABY” then END!
5. If A ≤ 10 print “CHILD” then END!
6. If A ≤ 18 print “TEEN” then END!
7. If A ≤ 50 print “ADULT” then END!
8. print “OLD”
9. STOP/FINISH/END

Don’t forget to test your program.
Try with 10, 25, and 59

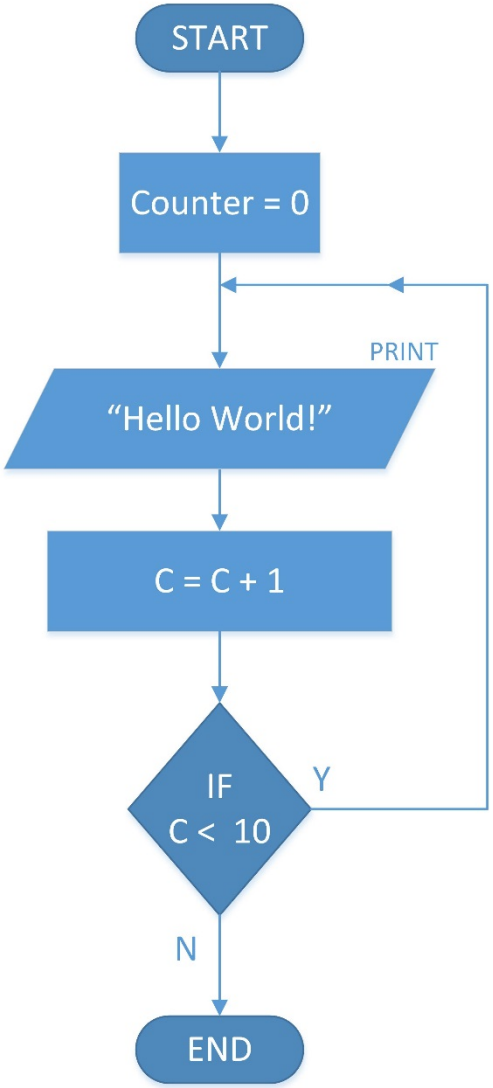


Example 9: ….. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program prints “Hello World!” to screen 10 times. **

Algorithm

1. Start
2. Assign C=0 (C is counter to count number of prints)
3. Print “Hello World!”
4. Assign C = C +1 (increasing the counter)
5. If C < 10 go to step 3.
6. STOP/FINISH/END



Example 10: (Algorithm and Flow-Chart)

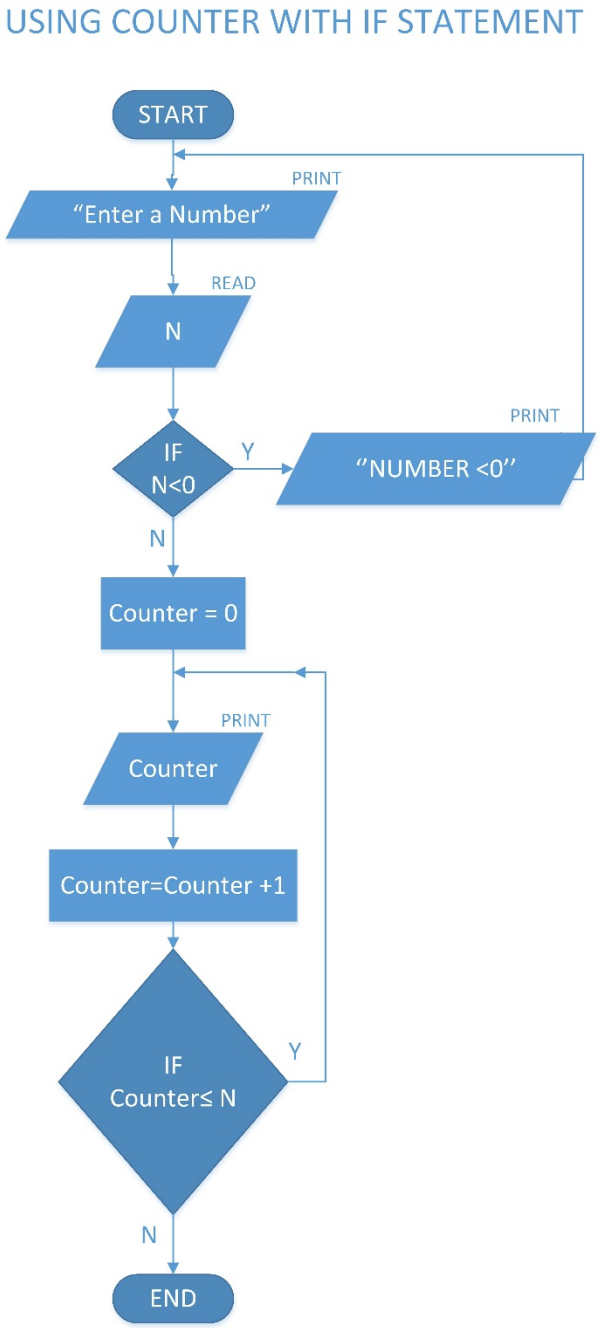
Write the algorithm and draw flow-chart for a program. The program prints out the integer numbers from 0 to a desired positive integer number. **

Algorithm

1. Start
2. Prompt user to enter a number.
3. Read the entered number and store it in N variable.
4. If N < 0 go to step 2.
5. Define/Assign Counter = 0
6. Print the Counter
7. Increase Counter by +1
8. If Counter ≤ N go to step 6.
9. STOP/FINISH/END

Test your program with -5, 0, and 7

One may also do this by using loop / repetition algorithm with step of +1.

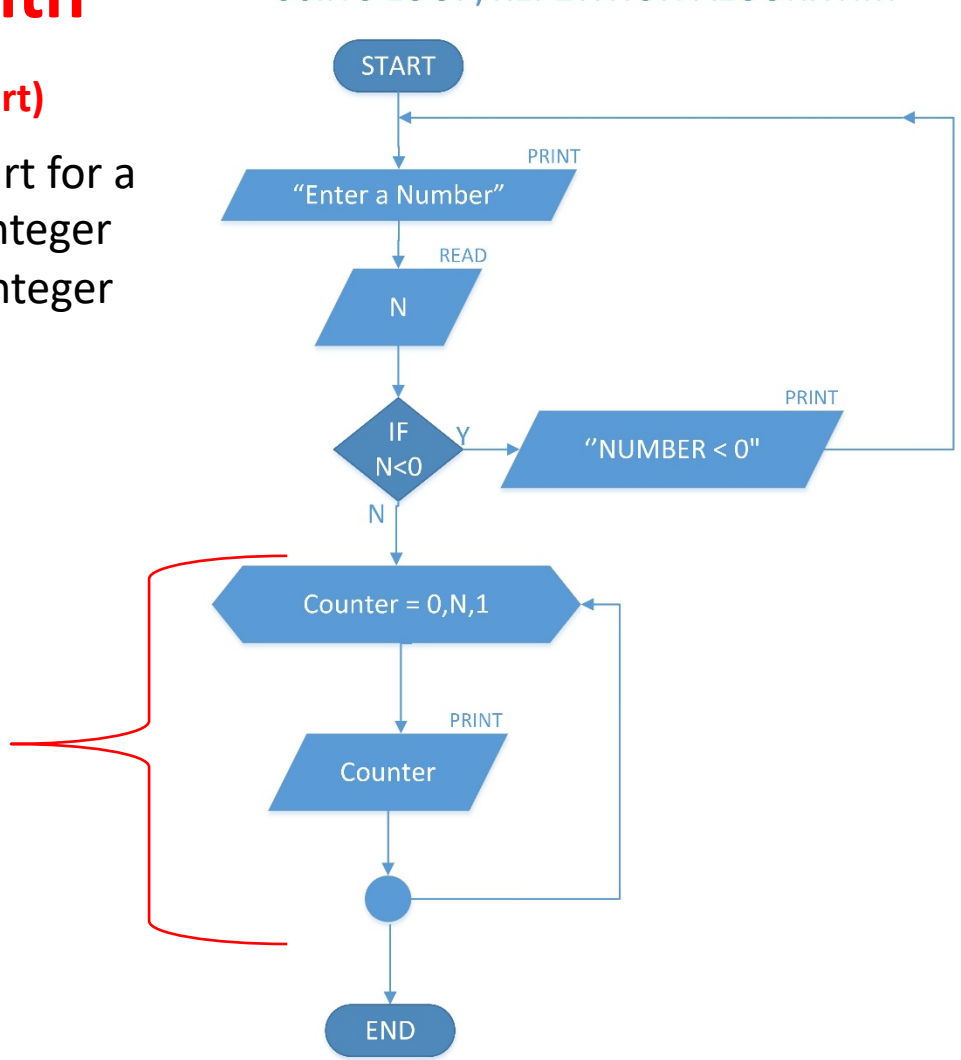


Example 10: Revision with Repetition / Loop (Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program prints out the integer numbers from 0 to a desired positive integer number. *

We may also do this by using loop / repetition algorithm with step of +1.

Loop / Repetition where
Counter goes from 0 to N step of +1
Repetition is N+1 times.

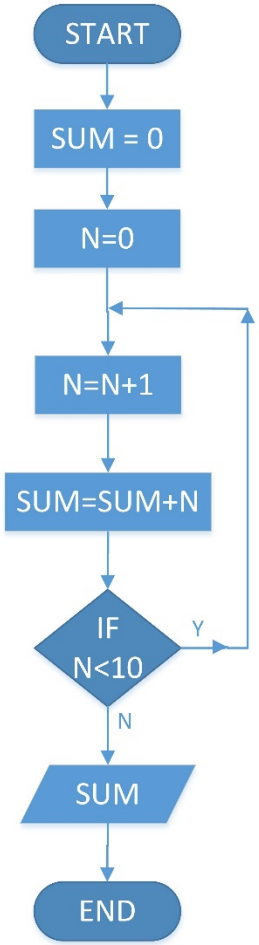


Example 11: …. (Algorithm and Flow-Chart)

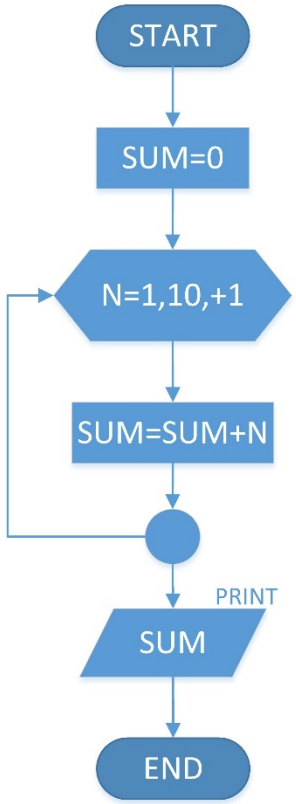
Explain what does the following program with flow-chart does? By using a loop/repetition procedure redraw another flow-chart that does the same. **

*It prints the sum of the integers from 1 to 10.
* We can do this by using repetition step of +1

USING COUNTER WITH IF STATEMENT



USING LOOP/REPETITION ALGORITHM

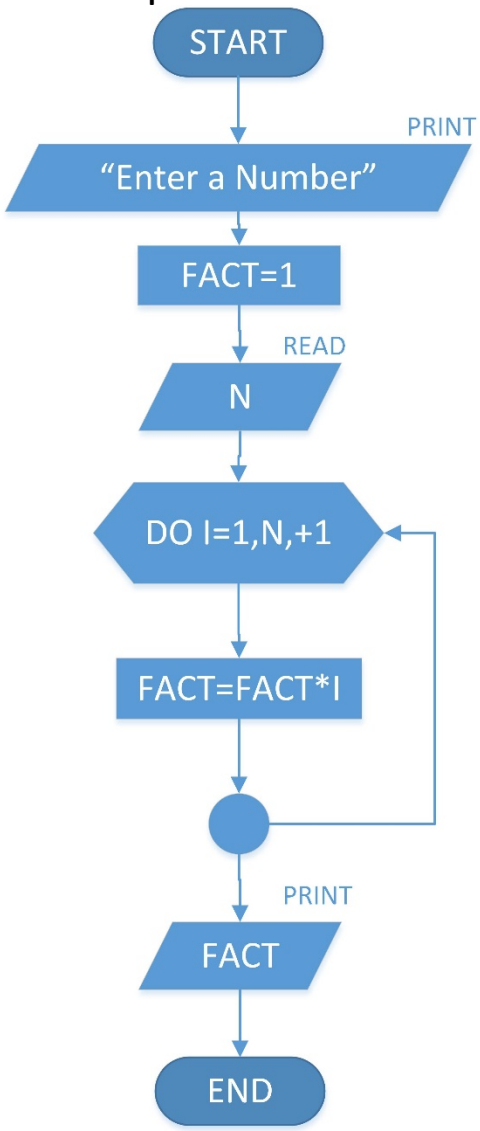


Example 12: …. (Algorithm and Flow-Chart)

Write the algorithm and draw flow-chart for a program. The program calculates the factorial ($\mathrm{n!} = 1.2.3 \ldots n-1.n$) of a desired positive number and prints it to the screen **.

Algorithm

1. Start
2. Prompt user to enter a positive integer number.
3. Read the number N
4. Assign fact=1
5. Set a repetition/loop with counter I starting from 1 to N, inside the repetition, multiply fact with I.
6. Print the fact
7. STOP/FINISH/END



Test your program with input 5.

Example 13: …. (Flow-Chart)

Draw flow-chart for a program. The program determines the sum of the even numbers from 0 to a desired number. **