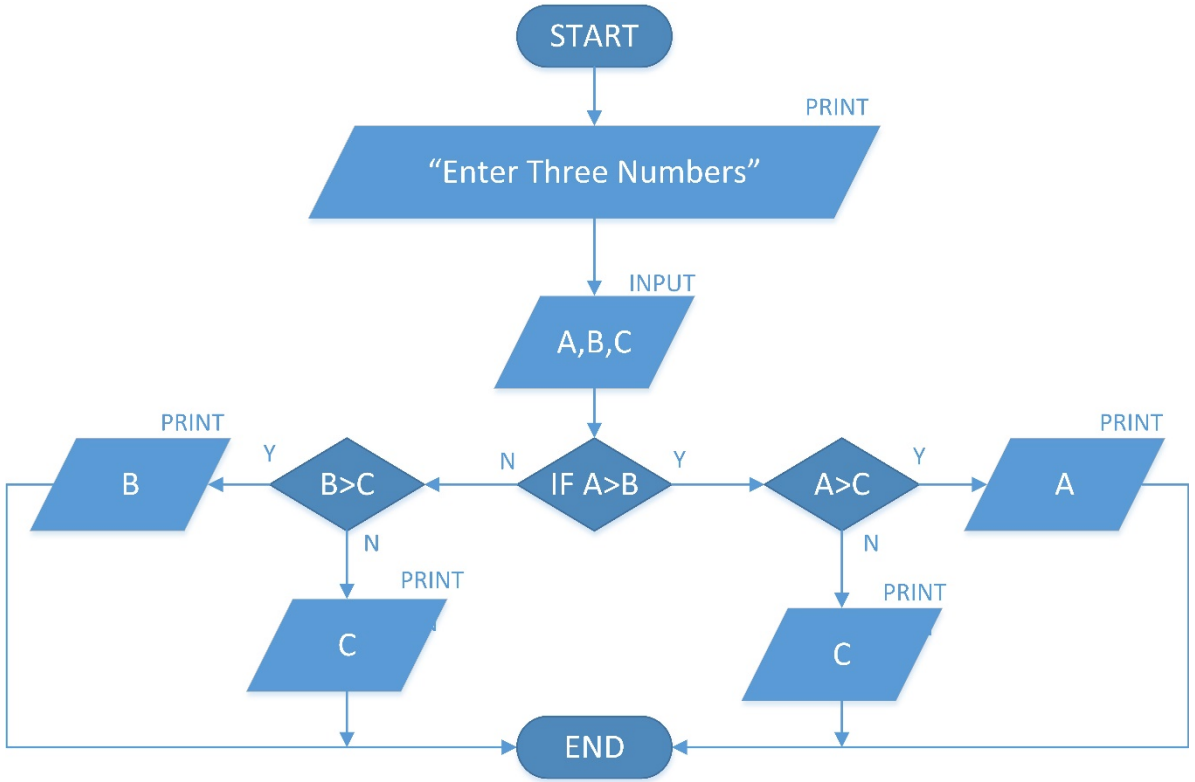
Concept of Programming (Flow Charts)

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

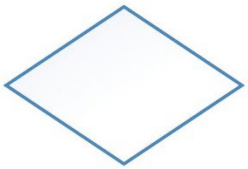
Most commonly used elements in flow charts and some others are shown below.

Commonly Used Elements

Start / End

Computation / Assignment

Input / Output Operations



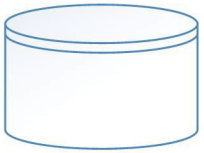
Decision Making and Braching

Connector or Joining of Two Part in a program.

Loops / Repetitions

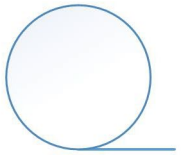
Flow Lines – Shows the direction to proceed.

Rarely Used Elements



Magnetic Disk (Secondary Memory)

Output to Printer



Magnetic Tape (Floppy)

Example 1: Printing Hello World

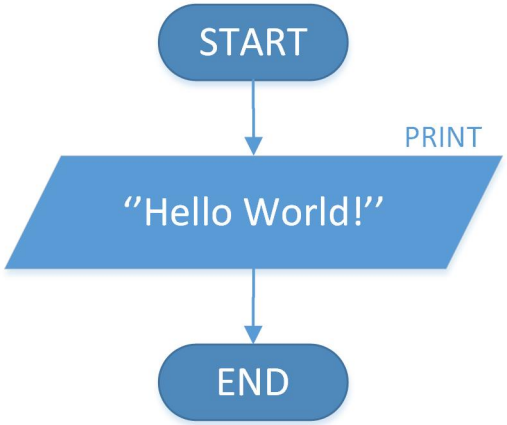
Write the algorithm, pseudocode and draw flow-chart for a program. The program prints the “Hello World!” to the screen.**

Algorithm

Print out the sting.

Pseudocode

Print “Hello World!”



Example 2: Zero – Positive – Negative Prog.

Write the algorithm, pseudocode 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. Get the number from user.
2. Check if the number is + , 0 or –
3. Print out the result

Pseudocode

Prompt user to enter a number

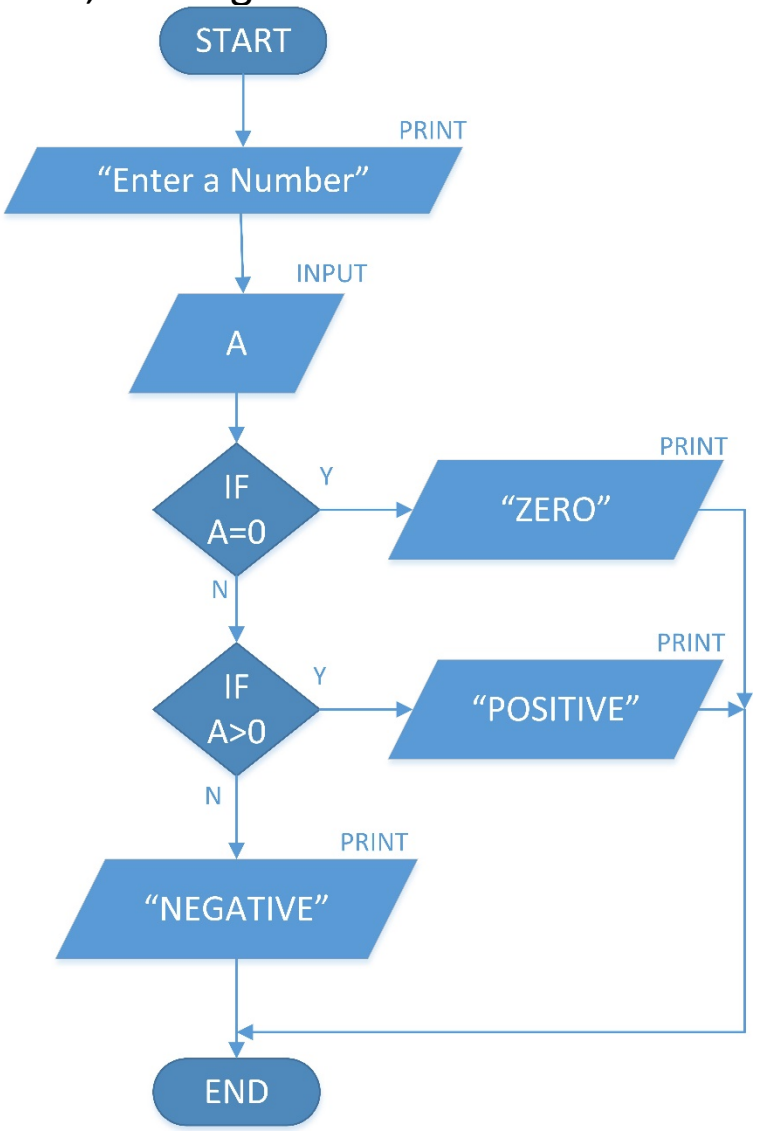
Input the number and store it in A

If A= 0 print out “ZERO” then END!

If A > 0 print “POSITIVE” then END!

Print out “NEGATIVE” then END!

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



Example 3: Adding and Multiplying Two Numbers

Write the algorithm, pseudocode 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. Get two numbers from user
2. Add the numbers
3. Multiply the numbers
4. Print out the results

Pseudocode

Prompt user to enter two numbers

Input two numbers and store them in A, B

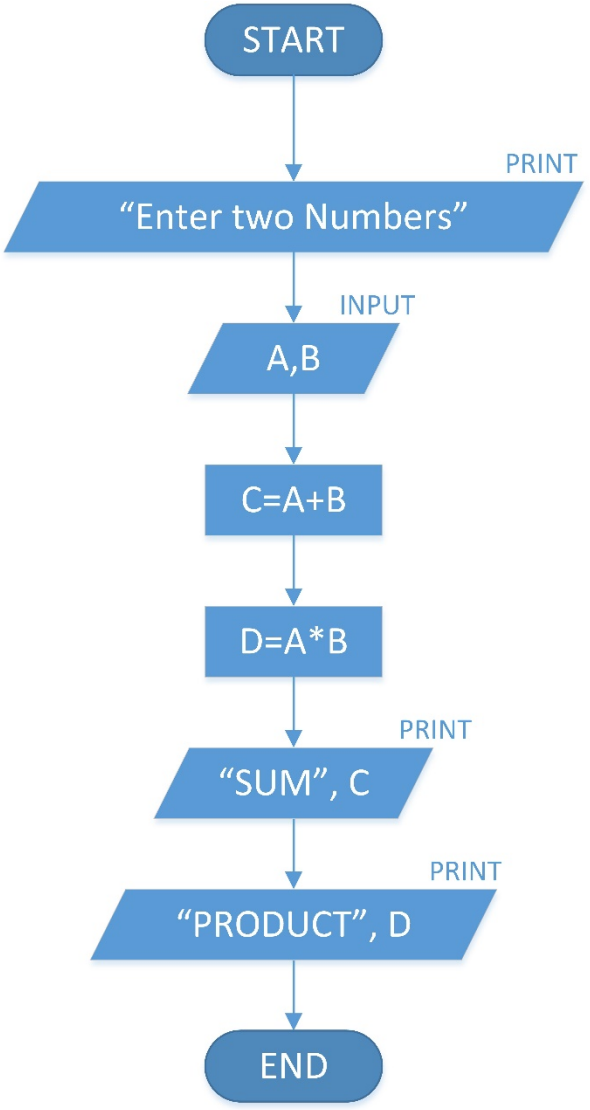
Define C = A+B

Define D = A*B

Print C

Print D

Now try testing you program by using
3 and 5



Example 4: Comparing Two Numbers and Finding Larger

Write the algorithm, pseudocode 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. Get two numbers from user
2. Check if they are same then print out SAME
3. Check which one is larger
4. Print out the larger number

Pseudocode

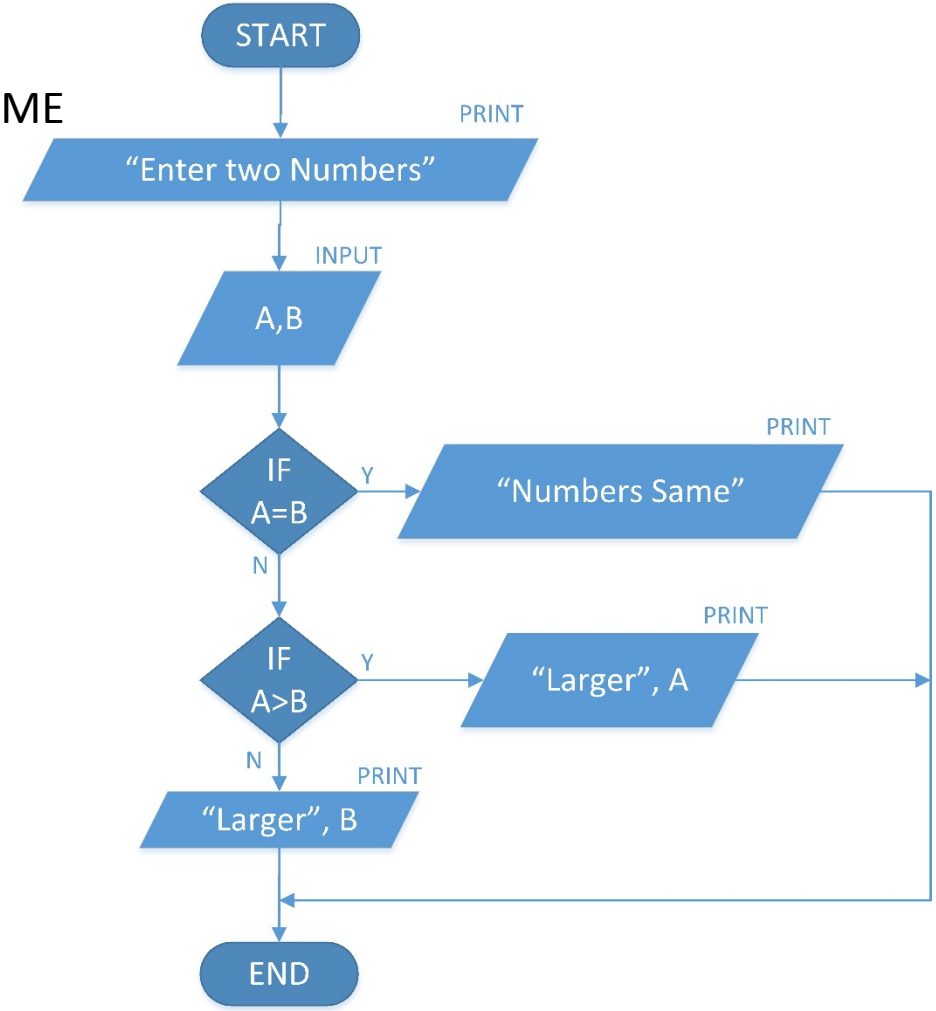
Input two numbers and store them in A, B

If A=B then print “Numbers Same” END!

If A>B then print “Larger”,A . END!

Print “Larger”,B

Now try testing you program by using
7 and 11
15 and 6
5 and 5



Example 5: Grade Calculator – Fail/Pass Program

Write the algorithm, pseudocode 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. Get Midterm and Final Grades from user
2. Calculate the grade as 40% of Mid. and 60% of Fin.
3. If result is bigger than 50 print out “PASSED”, END!
4. Print out “FAILED”

Pseudocode

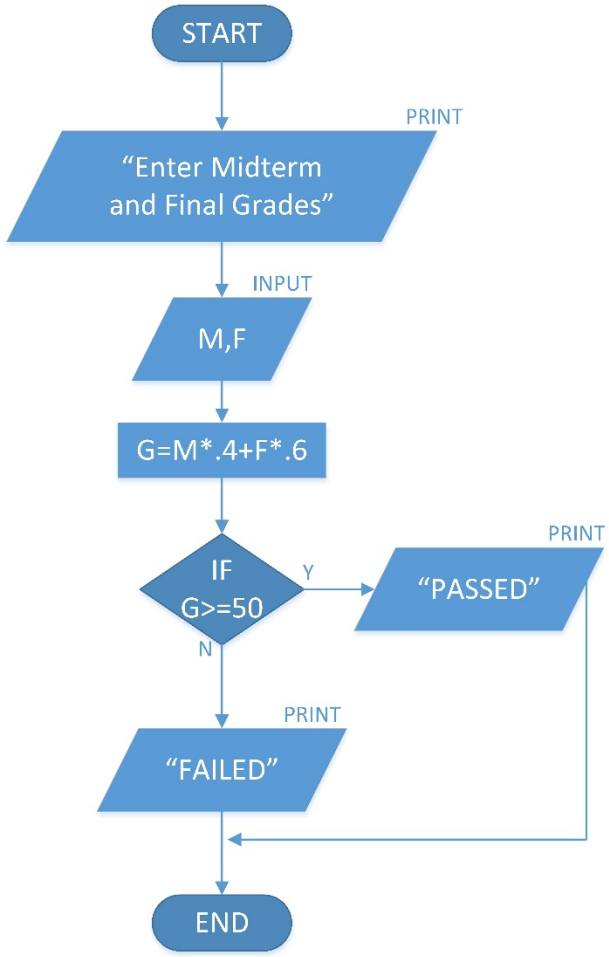
Input two numbers and store them in M, F

Assign/Calculate G = M*.4+F*.6

If G ≥ 50 then print “PASSED” , DONE!

Print “FAILED”

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



Example 6: Quadratic Eq. Solver Program

Write the pseudocode and draw flow-chart for a program. The program reads the coefficients of a quadratic equation in the form $ax^2 + bx + c = 0$ from the keyboard then determines the solutions and print the solution/s to the screen. If there is no real solution, program should print "No Real Solution" to the screen. **

Pseudocode

Prompt user to enter the coefficients.

Input the coefficients and store in a,b,c

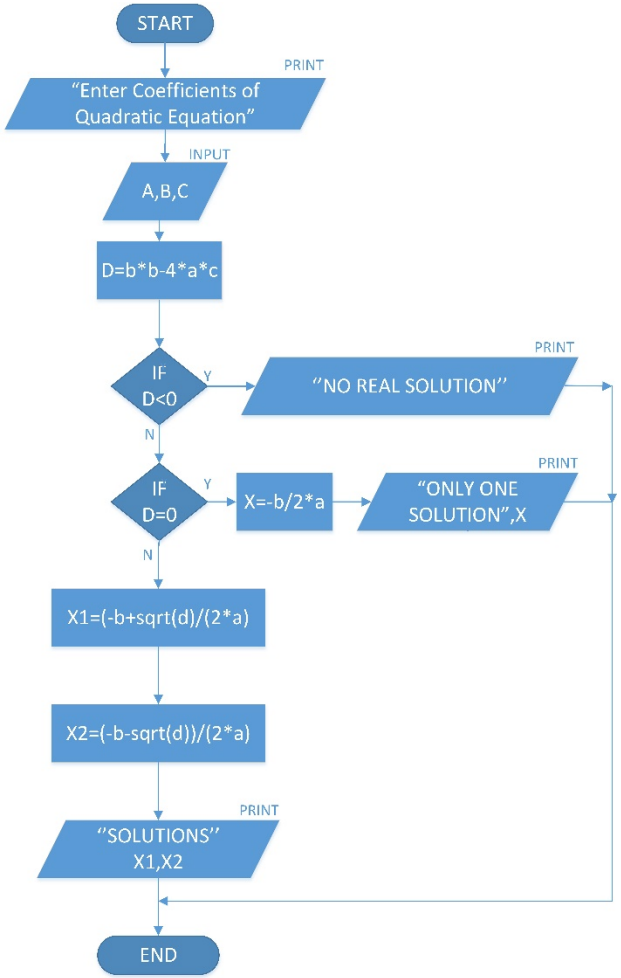
Assign d=$b^2 - 4ac$

If d < 0 print "No Real Solution" END!

if d = 0 assign x = -b / 2a print "One Solution", x

Assign $x1 = \frac{-b-\sqrt{d}}{2a}$ and $x2 = \frac{-b+\sqrt{d}}{2a}$

print "Two Solution", x1,x2



Don't forget to test your program with different values.

Example 7: Even or Odd Program

Write the pseudocode 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. **

Pseudocode

Prompt user to enter a number

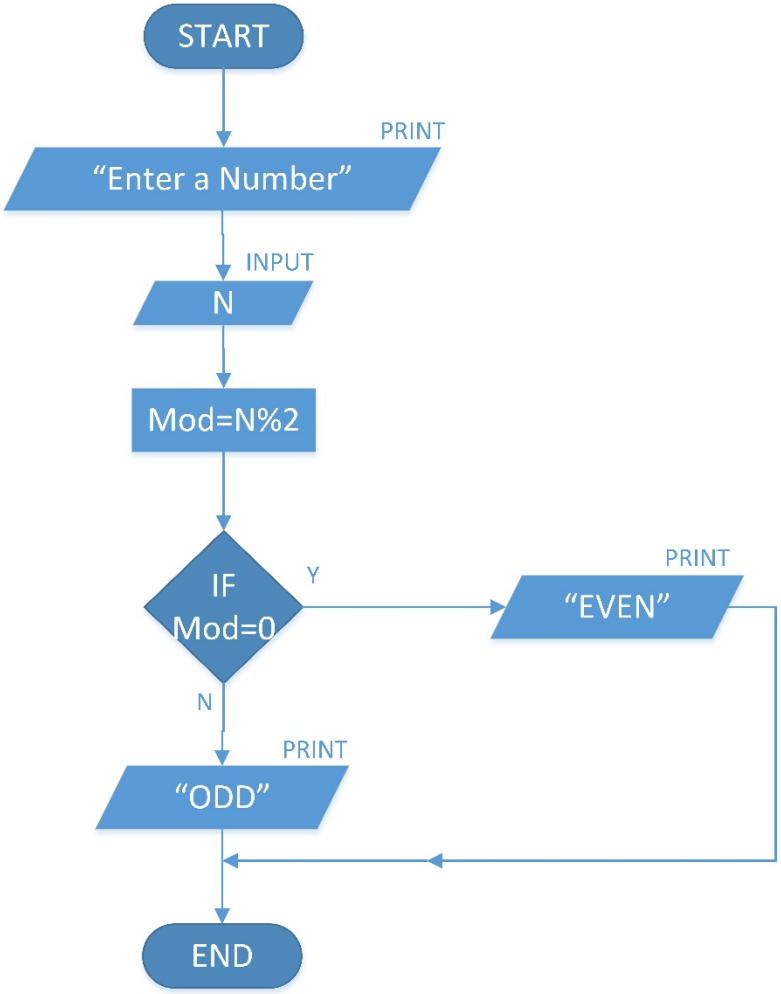
Input the number and store in N

Assign mod = N % 2

If mod = 0 then print “EVEN” END!

Print “ODD”

Test your program with inputs 6 and 11.

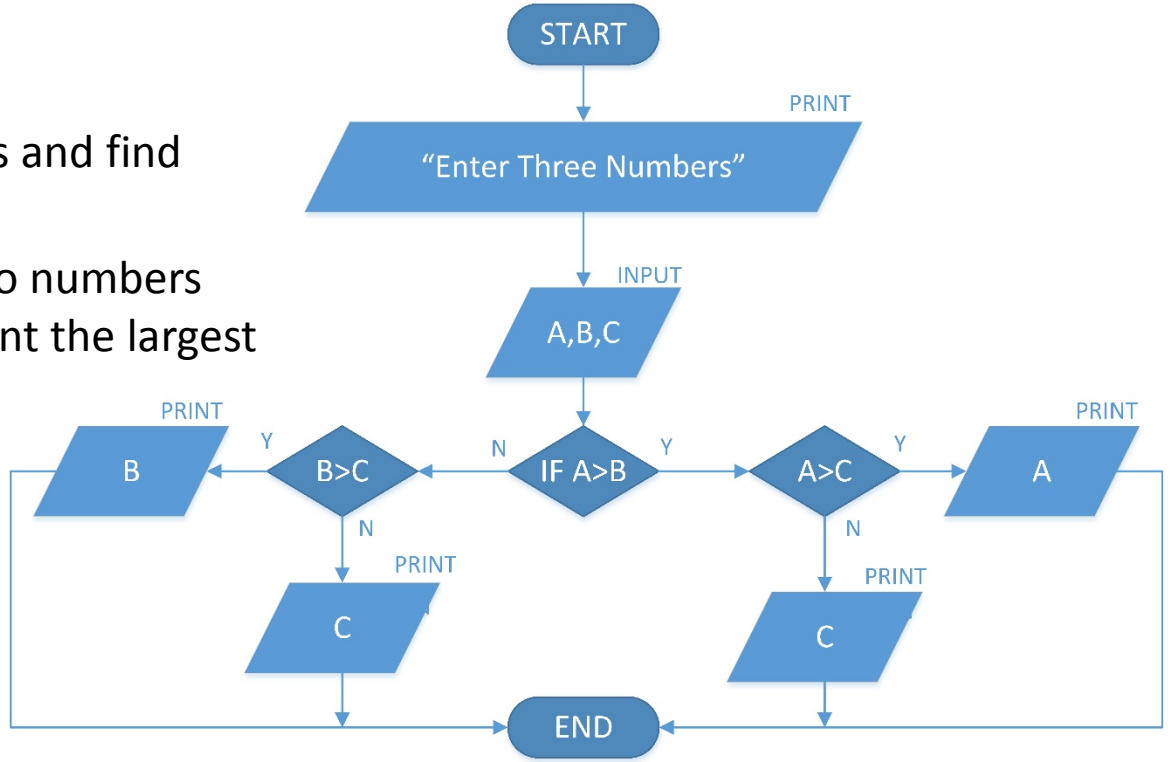


Example 8: Finding Greatest of Three Numbers

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. Get three numbers from user
2. Compare the first two numbers and find largest one of two
3. Compare the largest of first two numbers with the third number then print the largest to the screen



Now try testing you program by using
5 , 7, and 2
11 , 3, and 21

Example 9: Age Status Program

Write the pseudocode 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. **

Pseudocode

Prompt user to enter his/her age.

Input the entered number and store it in A

If A ≤ 4 print “BABY” then END!

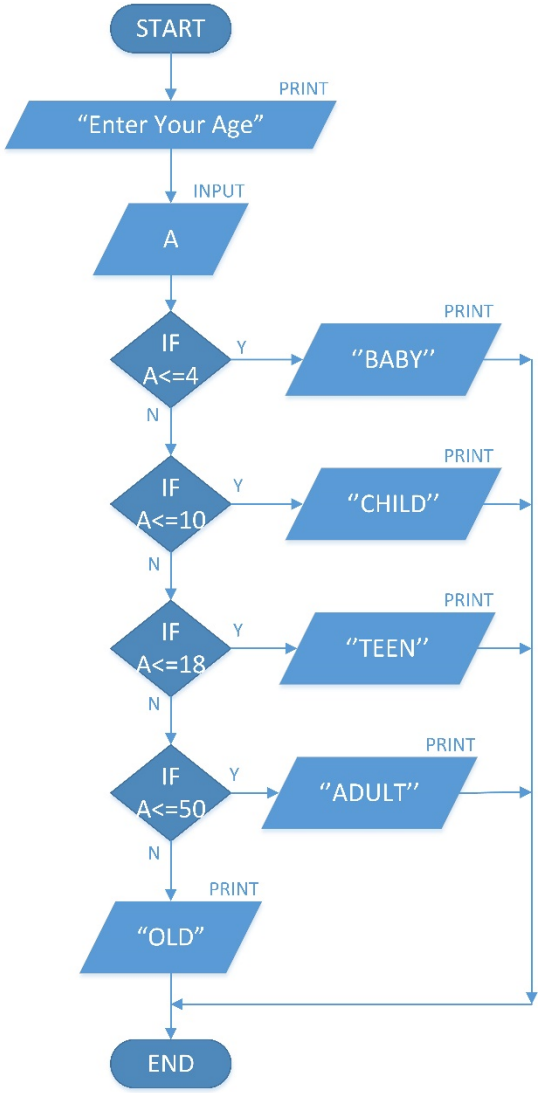
If A ≤ 10 print “CHILD” then END!

If A ≤ 18 print “TEEN” then END!

If A ≤ 50 print “ADULT” then END!

print “OLD”

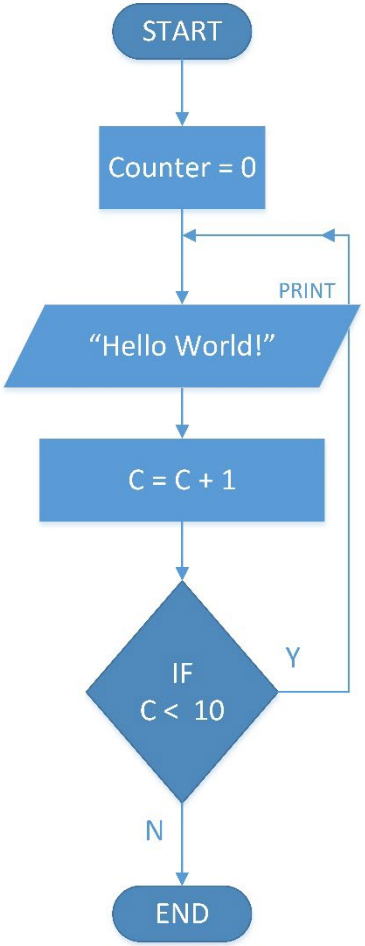
Don’t forget to test your program.
Try with 10, 25, and 59



Example 10: Print “Hello World!” 10 Times

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

USING COUNTER WITH IF STATEMENT



Pseudocode

Assign/Set C=0 *(C is counter to count number of prints)*

Print “Hello World!”

Assign C = C +1 *(increasing the counter)*

If C < 10 go to step 2.

Example 11: Print Integers from 0 to N

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

Psedocode

Prompt user to enter a number.

Input the entered number and store it in N

If N < 0 go to step 1.

Define/Assign Counter = 0

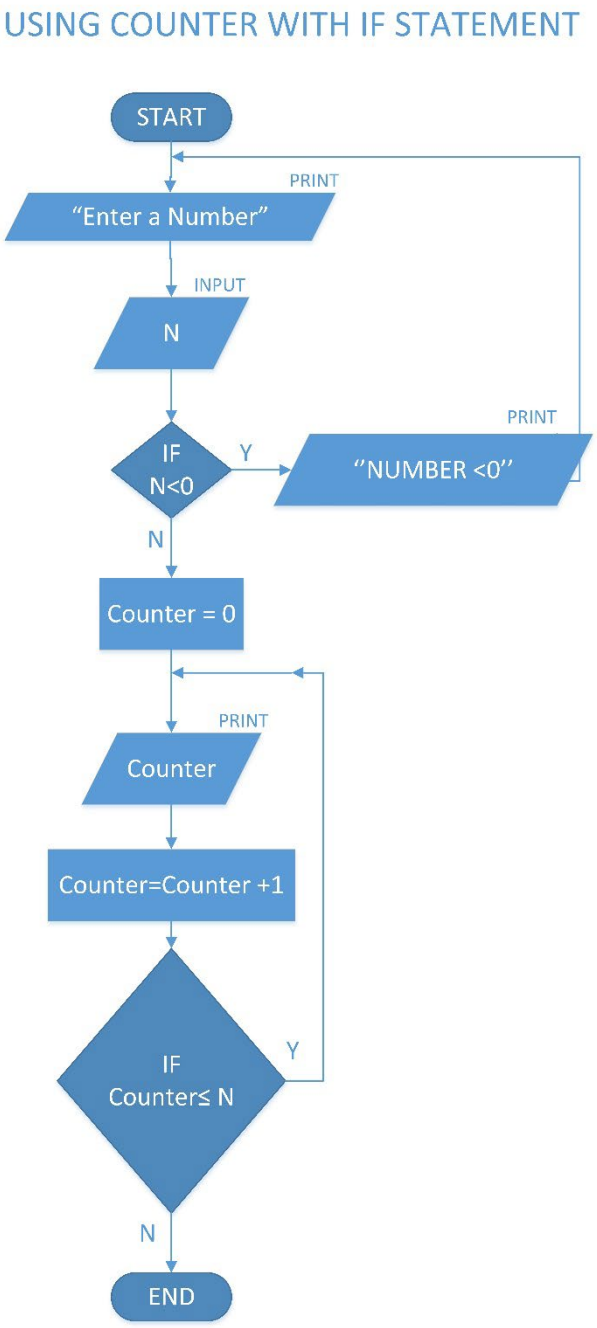
Print the Counter

Increase Counter by +1

If Counter ≤ N go to step 5.

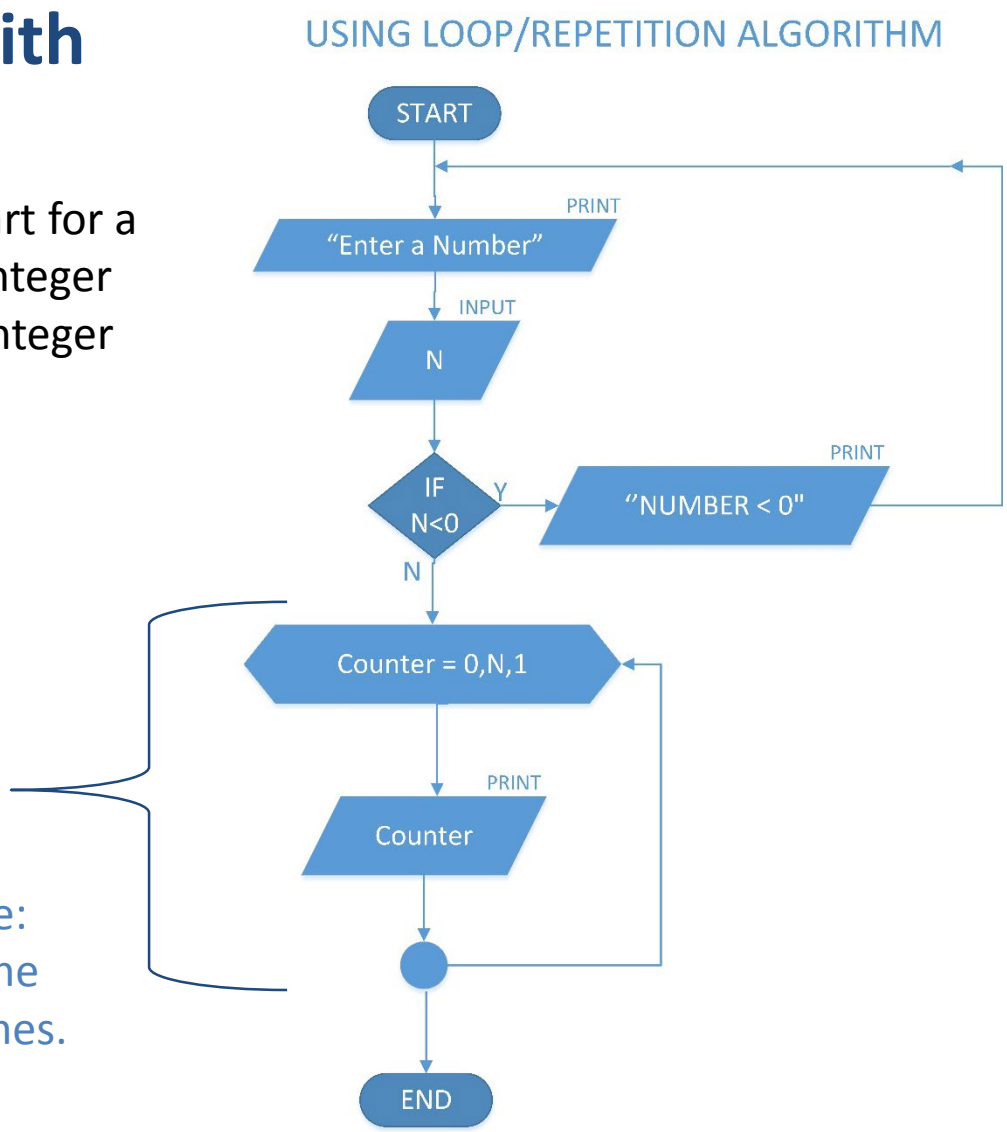
Test your program with -5, 0, and 7

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



Example 11: Revision with Repetition / Loop

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. *



Here we use a loop / a repetition algorithm with step of +1.

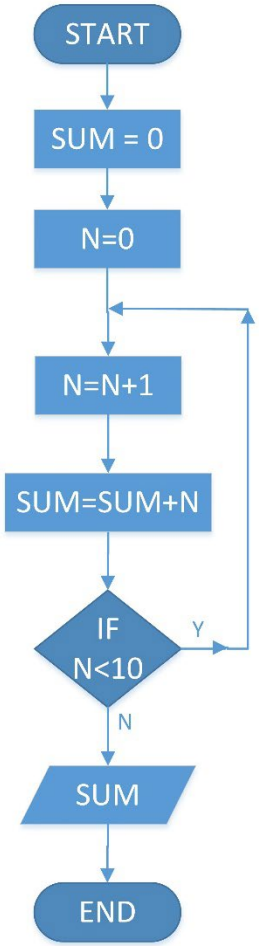
Please note that Loop / Repetition here: Counter goes from 0 to N step of +1. The part inside repetition repeated N+1 times.

Example 12: Checking Understanding of Flow-Chart

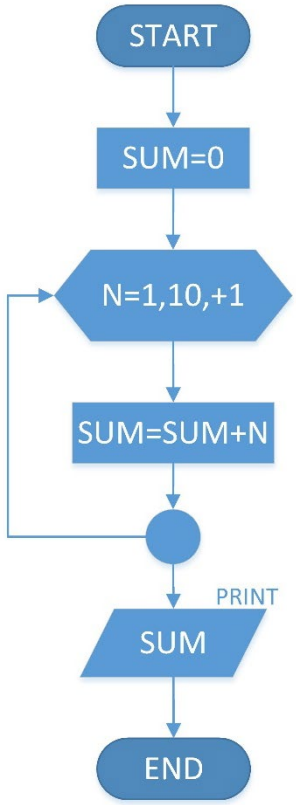
Explain what does the following program with flow-chart does? By using a loop/repetition procedure redraw another flow-chart that does the same. **

*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

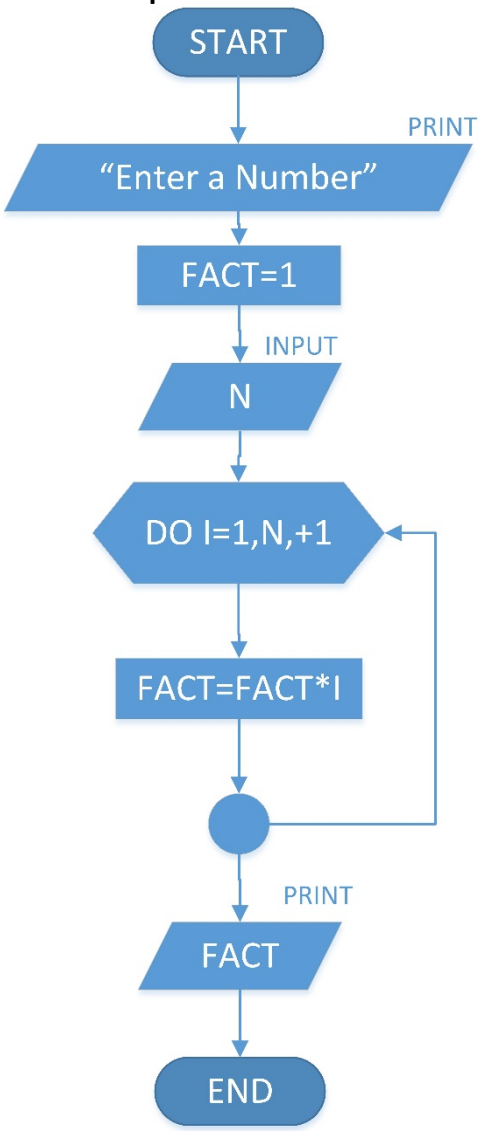


Example 13: Factorial Program

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. Take a number from the user
2. Assign fact=1
3. Set a repetition/loop with counter I starting from 1 to entered number
4. Inside the repetition, multiply fact with I values.
5. Print out the result



Test your program with input 5.

Example 14: Sum of Even Numbers from 0 to N

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