Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on Boundary-Value Analysis Robust, execute the test cases and discuss the results.

**REQUIREMENTS**

**R1.** The system should accept 3 positive integer numbers (a, b, c) which represents 3 sides of the triangle.

**R2.** Upper Limit for the size of any side is 10. If not print the message given input

is **not in a range(Invalid Input)**

**R3.** Based on the input should determine if a triangle can be formed or not.

If not print the message“doesn’t form a triangle” for the given input

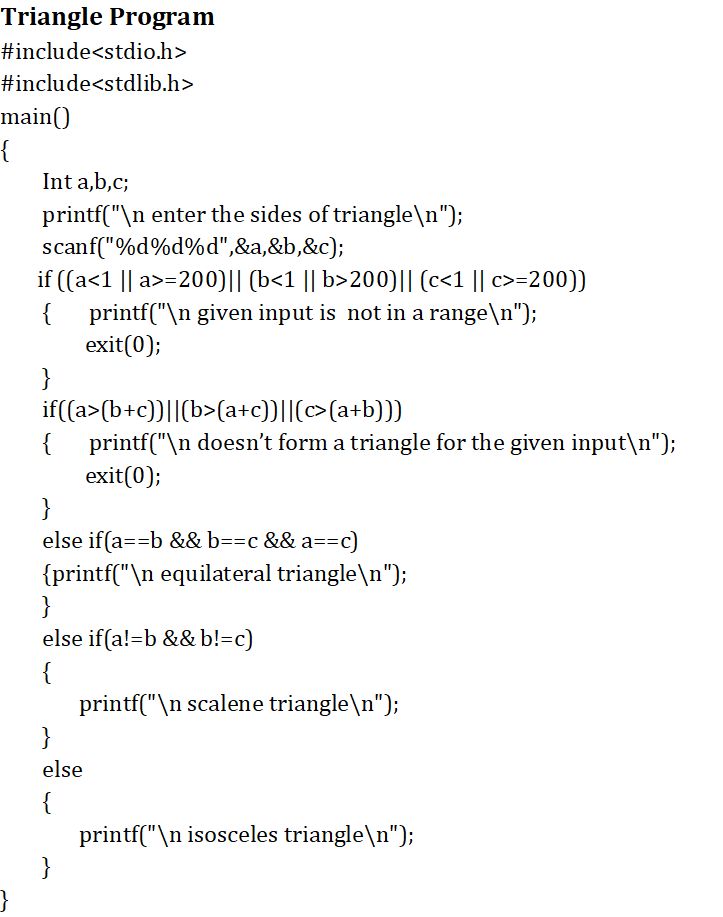
**R4.** If therequirement R2 & R3 is satisfiedthen the system should determine the

type of the triangle, which can be

* + - Equilateral (i.e. all the three sides are equal)
    - Isosceles (i.e Two sides are equal)
    - Scalene (i.e All the three sides are unequal)

**Steps to be followed for the above lab program.**

1. Write a program for the above problem statement
2. Prepare the test Input for the program
3. Write the test-cases in the lab test-case Template
4. Execute the test-cases
5. Compare the Expected output and Actual output and write the status in the test-case table.
6. Show few failed test cases by following the mutation testing.

**Write a program as per the requirement** 

min- min min+ nom max- max max+

***a |------|--------|-----------|---------|-----|------|***

**-1 1 2 4 9 10 11**

***b |------|--------|-----------|---------|-----|------|***

**-1 1 2 4 9 10 11**

***c |------|--------|-----------|---------|-----|------|***

**-1 1 2 5 9 10 11**

Follow the lab Test case template and consider every input as a different test case.

Test-case Table for Triangle Program using Boundary Value Analysis

