CS 455: Software Testing Techniques
Due Date: April 17, 2024
Note: This assignment can be done in groups of two. Grade will be awarded on the basis of a viva conducted in the class on the deadline date. The report must be shown in softcopy form to the course instructor at the time of viva.

The triangle program is a well known program for testing which appeared in Myers book on testing in 1979. It is has been used by many other books and is a good benchmark for other ideas about testing. The problem description is as under:

"*A program reads three integers which represent the three sides of a triangle. The program outputs a message whether a particular triangle is a scalene, isosceles or equilateral*".

From elementary geometry it can be recalled that:
1. A triangle is a polygon with three sides.
2. The vertices of a triangle must not be in a straight line.
3. An equilateral triangle has three sides of equal length.
4. An isosceles triangle has two sides of equal length.
5. A scalene triangle has three sides of different lengths.

A pseudo code that implements the logic of the program is given below

Algorithm 1

```
Type Kind = {scalene, isosceles, equilateral, badside,
notriagnle}//data type definition
method TriangleTest(s1, s2, s3 : Integer) : Kind
begin
    if s1 <=0 s2 <= 0 or s3 <= 0
    then return badside
    else
        if s1 + s2 <= s3 or s2 + s3 <= s1 or s1 + s3 <= s2
        then return  notriangle
        else
            if s1 == s2 & s2 == s3
            then
                return equilateral
            else
                if s1 == s2 or s2 == s3 or s1 == s3
                then
                    return isosceles
                else
                    return scalene
end
```

Exercise Questions

1) You will program the above pseudo code in any Java.
2) You will extract the CFG of your with the help of the AutoTester tool available on (http://autotester.bitsol.tech/, you may need an older version of Java like Java 8 to run the jar file on your machine) and WebApps available for Graph Coverage at (https://cs.gmu.edu:8443/offutt/coverage/GraphCoverage), (https://cs.gmu.edu:8443/offutt/coverage/DFGraphCoverage) and logic coverage available at (https://cs.gmu.edu:8443/offutt/coverage/LogicCoverage).
3) In the final part of this exercise, you will find out what test requirements /test paths corresponding to a given coverage criteria are supported / provided by these tool/WebApps and answer the following questions below:
    1. Does the given tool supports node coverage (NC) criteria. Give the test cases generated by each tool for this criteria. Are these minimized?
    2. Does the given tool supports edge coverage (EC) criteria. Give the test cases generated by each tool for this criteria. Are these minimized?
    3. Does the given tool supports predicate coverage (PC) criteria. (non-distributive predicate coverage is sufficient here). Give the test cases generated by the tool for this criteria? Are these minimized?
    4. Does the given tool supports clause coverage (CC). Give the test cases generated by the tool for this criteria. Are these minimized?
    5. Does the given tool supports restricted active clause coverage (RACC) also known as MCDC. Give the test cases generated by the tool for this criteria? Are these minimized?
4) For the questions 1-5 above in part (3) do you think that the test cases generated are executable? If not how can you make them executable on your code?
5) Execute the test cases after making them executable (if they are not) on your code and analyze your code behavior corresponding to executing these test cases. Provide a brief report of your analysis what you found or what you did not find or what you should have found by executing these test cases on your code. Your analysis must be critical to give you a good grade on this assignment.

Tool Assessment Exercise

For each of the coverage models above (NC, EC, PC, CC, RACC) and their corresponding test cases generated by the tool(s) answer the following 14 questions for each tool in the form of a table. For each requirement fulfilled give that tool one point. Maximum points are 14. In the end choose the tool which gives the highest points.

1) Do you have a test case that represents a valid scalene triangle?
2) Do you have a test case that represents a valid isosceles triangle?
3) Do you have a test case that represents a valid equilateral triangle?
4) Do you have three test case that represent valid isosceles triangle that is you have tried all three permutations of two equal sides?
5) Do you have a test case in which one side has a zero value?
6) Do you have a test case in which one side has a negative value?
7) Do you have a test case with three integers such that the sum of two is equal to the third?
8) Do you have at least three test cases for question 7 above such that you have tried all permutations of sum of lengths of two sides equal to the length of third side?
9) Do you have a test case with three integers greater than zero such that the sum of two is less than the third?
10) Do you have at least three test cases for question 9 such that you have tried all three permutations?
11) Do you have a test case in which all sides are zero?
12) Do you have a test case with non-integer values?
13) Do you have a test case with wrong number of values (two or less or four or more)
14) For each test case did you specify the expected output along with the input value?

Good Luck :)