**Assignment Description:**

Sometimes you will be given a program that someone else has written, and you will be asked to fix, update and enhance that program.   In this assignment you will start with an existing implementation of the classify triangle program that will be given to you.   You will also be given a starter test program that tests the classify triangle program, but those tests are not complete.

In order to determine if the program is correctly implemented, you will need to update the set of test cases in the test program.  You will need to update the test program until you feel that your tests adequately test all of the conditions.   Then you should run the complete set of tests against the original triangle program to see how correct the triangle program is.    Capture and then report on those results in a formal test report described below.   For this first part you should not make any changes to the classify triangle program.  You should only change the test program.

Based on the results of your initial tests, you will then update the classify triangle program to fix all defects.  Continue to run the test cases as you fix defects until all of the defects have been fixed.   Run one final execution of the test program and capture and then report on those results in a formal test report described below.

**Author:** Olof Persson

**Summary:** After creating the initial tests which would test as much as was possible, the first run through of the tests created mostly failed test results. After code inspections, somethings were changed in the code, then the failed tests were analyzed to change more of the code until there was 100% pass rate of the test cases. Along the way during the test analysis, it was realized that one of the test cases was incorrectly made so it was fixed to reflect the correct result and then the test passed.

**I pledge my honor that I will be abide by the Stevens Honor System.**

**Details Results:**

**Initial Test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| testInvalidInputA | 201,201,201 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputB | 0,0,0 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputC | -1,-1,-1 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputD | True, -1, 201 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputE | [1],1,2 | ‘InvalidInput’ | TypeError | Fail |
| testInvalidInputF | ‘a’, ‘b’, ‘c’ | ‘InvalidInput’ | TypeError | Fail |
| testNotTriangleA | 1,1,3 | ‘NotTriangle’ | ‘InvalidInput’ | Fail |
| testNotTriangleB | 5,2,1 | ‘NotTriangle’ | ‘InvalidInput’ | Fail |
| testNotTriangleC | 2,9,1 | ‘NotTriangle’ | ‘InvalidInput’ | Fail |
| testRightTriangleA | 3,4,5 | ‘Right’ | ‘InvalidInput’ | Fail |
| testRightTriangleB | 5,12,13 | ‘Right’ | ‘InvalidInput’ | Fail |
| testRightTriangleC | 7,24,25 | ‘Right’ | ‘InvalidInput’ | Fail |
| testIsocelesTriangleA | 3,3,4 | ‘Isoceles’ | ‘InvalidInput’ | Fail |
| testIsocelesTriangleB | 150,40,150 | ‘Isoceles’ | ‘InvalidInput’ | Fail |
| testIsocelesTriangleC | 100,7,7 | ‘Isoceles’ | ‘InvalidInput’ | Fail |
| testScaleneTriangleA | 3,4,6 | ‘Scalene’ | ‘InvalidInput’ | Fail |
| testScaleneTriangleB | 5,12,14 | ‘Scalene’ | ‘InvalidInput’ | Fail |
| testScaleneTriangleC | 7,24,26 | ‘Scalene’ | ‘InvalidInput’ | Fail |
| testEquilateralTriangleA | 1,1,1 | ‘Equilateral’ | ‘InvalidInput’ | Fail |
| testEquilateralTriangleB | 100,100,100 | ‘Equilateral’ | ‘InvalidInput’ | Fail |
| testEquilateralTriangleC | 200,200,200 | ‘Equilateral’ | ‘InvalidInput’ | Fail |

**Iterative Updating Process**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Run 1** | **Test Run 2** | **Test Run 3** | **Test Run 4** | **Test Run 5** |
| **Tests Planned** | **21** | **21** | **21** | **21** | **21** |
| **Tests Executed** | **21** | **21** | **21** | **21** | **21** |
| **Tests Passed** | **4** | **6** | **16** | **19** | **21** |
| **Defects Found** | **17** | **15** | **5** | **2** | **0** |
| **Defects Fixed** | **0** | **2** | **10** | **3** | **2** |

**After Updating Code**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| testInvalidInputA | 201,201,201 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputB | 0,0,0 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputC | -1,-1,-1 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputD | True, -1, 201 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputE | [1],1,2 | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testInvalidInputF | ‘a’, ‘b’, ‘c’ | ‘InvalidInput’ | ‘InvalidInput’ | Pass |
| testNotTriangleA | 1,1,3 | ‘NotTriangle’ | ‘NotTriangle’ | Pass |
| testNotTriangleB | 5,2,1 | ‘NotTriangle’ | ‘NotTriangle’ | Pass |
| testNotTriangleC | 2,9,1 | ‘NotTriangle’ | ‘NotTriangle’ | Pass |
| testRightTriangleA | 3,4,5 | ‘Right’ | ‘Right’ | Pass |
| testRightTriangleB | 5,12,13 | ‘Right’ | ‘Right’ | Pass |
| testRightTriangleC | 7,24,25 | ‘Right’ | ‘Right’ | Pass |
| testIsocelesTriangleA | 3,3,4 | ‘Isoceles’ | ‘Isoceles’ | Pass |
| testIsocelesTriangleB | 150,40,150 | ‘Isoceles’ | ‘Isoceles’ | Pass |
| testIsocelesTriangleC | 100,7,7 | ‘Isoceles’ | ‘Isoceles’ | Pass |
| testScaleneTriangleA | 3,4,6 | ‘Scalene’ | ‘Scalene’ | Pass |
| testScaleneTriangleB | 5,12,14 | ‘Scalene’ | ‘Scalene’ | Pass |
| testScaleneTriangleC | 7,24,26 | ‘Scalene’ | ‘Scalene’ | Pass |
| testEquilateralTriangleA | 1,1,1 | ‘Equilateral’ | ‘Equilateral’ | Pass |
| testEquilateralTriangleB | 100,100,100 | ‘Equilateral’ | ‘Equilateral’ | Pass |
| testEquilateralTriangleC | 200,200,200 | ‘Equilateral’ | ‘Equilateral’ | Pass |