1. 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.

2. Author: Herb Zieger

3. Summary:

Github Repo: <https://github.com/zildj/Triangle567>

Test-driven debugging is a very effective way of fixing erroneous code. As I fixed defects in the code and ran the tests, more defects became apparent. However, writing tests as you write code would be a more effective way of error-checking than writing all the code, then all the tests in my opinion.

4. Honor pledge:

I pledge my honor that I have abided by the Stevens Honor System. -*Herb Zieger*

5. Detailed results:

(on following pages)

**Initial test results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Results | Actual Results | Pass or Fail |
| testRightTriangleA | (3,4,5) | Right | InvalidInput | Fail |
| testRightTriangleB | (5,3,4) | Right | InvalidInput | Fail |
| testRightTriangleC | (13, 12, 5) | Right | InvalidInput | Fail |
| testRightTriangleD | (8, 6, 10) | Right | InvalidInput | Fail |
| testEquilateralTriangleA | (1,1,1) | Equilateral | InvalidInput | Fail |
| testEquilateralTriangleB | (10,10,10) | Equilateral | InvalidInput | Fail |
| testIsocelesTriangleA | (3, 3, 2) | Isoceles | InvalidInput | Fail |
| testIsocelesTriangleB | (4, 6, 6) | Isoceles | InvalidInput | Fail |
| testIsocelesTriangleC | (6, 4, 6) | Isoceles | InvalidInput | Fail |
| testIsocelesTriangleD | (6, 6, 4) | Isoceles | InvalidInput | Fail |
| testScaleneTriangleA | (10, 15, 12) | Scalene | InvalidInput | Fail |
| testInvalidInputA | (-1, -1, -1) | InvalidInput | InvalidInput | Pass |
| testInvalidInputB | (201, 201, 201) | InvalidInput | InvalidInput | Pass |
| testInvalidInputC | ("200", "0", "0") | InvalidInput | TypeError | Fail |
| testNotATriangleA | (5, 1, 1) | NotATriangle | InvalidInput | Fail |
| testNotATriangleB | (1, 5, 1) | NotATriangle | InvalidInput | Fail |
| testNotATriangleC | (1, 1, 5) | NotATriangle | InvalidInput | Fail |

**Test Run Matrix:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test Run 1 | Test Run 2 | Test Run 3 | Test Run 4 |
| Tests Planned | 17 | 17 | 17 | 17 |
| Tests Executed | 17 | 17 | 17 | 17 |
| Tests Passed | 2 | 6 | 10 | 17 |
| Defects Found | 2 | 1 | 3 | 0 |
| Defects Fixed | 0 | 2 | 1 | 3 |

**Final test results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Results | Actual Results | Pass or Fail |
| testRightTriangleA | (3,4,5) | Right | Right | Pass |
| testRightTriangleB | (5,3,4) | Right | Right | Pass |
| testRightTriangleC | (13, 12, 5) | Right | Right | Pass |
| testRightTriangleD | (8, 6, 10) | Right | Right | Pass |
| testEquilateralTriangleA | (1,1,1) | Equilateral | Equilateral | Pass |
| testEquilateralTriangleB | (10,10,10) | Equilateral | Equilateral | Pass |
| testIsocelesTriangleA | (3, 3, 2) | Isoceles | Isoceles | Pass |
| testIsocelesTriangleB | (4, 6, 6) | Isoceles | Isoceles | Pass |
| testIsocelesTriangleC | (6, 4, 6) | Isoceles | Isoceles | Pass |
| testIsocelesTriangleD | (6, 6, 4) | Isoceles | Isoceles | Pass |
| testScaleneTriangleA | (10, 15, 12) | Scalene | Scalene | Pass |
| testInvalidInputA | (-1, -1, -1) | InvalidInput | InvalidInput | Pass |
| testInvalidInputB | (201, 201, 201) | InvalidInput | InvalidInput | Pass |
| testInvalidInputC | ("200", "0", "0") | InvalidInput | InvalidInput | Pass |
| testNotATriangleA | (5, 1, 1) | NotATriangle | NotATriangle | Pass |
| testNotATriangleB | (1, 5, 1) | NotATriangle | NotATriangle | Pass |
| testNotATriangleC | (1, 1, 5) | NotATriangle | NotATriangle | Pass |