HW 02a - Testing a legacy program and reporting on testing results

1)Description of assignment:

From time to time, you may need to improve, update, or fix programs written by others. You will begin this project with a pre-existing version of the classify triangle software that you will be given. Not only that, but you will also receive a beginning test program that partially tests the classification triangulation program. These are the two files: Triangle.py and Test Triangle.py The triangle categorization software is implemented as a starting version in Triangle.py.

Triangle.py classify Triangle () function is tested using a set of unit test cases found in Test Triangle.py

You must alter the list of test cases in the test program in order to assess whether the software is correctly implemented. Until you believe that your tests appropriately test all of the conditions, you will need to update the test program. Then, to determine how accurate the original triangle program is, you should run every test against it. Observe the results, and then document them in the official test report that is described below. You shouldn't alter the categorize triangle program at all for this first stage. Simply put, changing the test software is enough. The classify triangle program will then be updated to address all issues based on the outcomes of your initial tests.Run the test cases again and again until all the errors have been corrected. Run the test program one last time, record the results, and then report on them in the official test report outlined below.

Keep in mind that you shouldn't just substitute your reasoning from Assignment 1 with the existing reason. In most cases, test teams are forced to make corrections to code that has already been sent to them rather than having the option of starting from fresh.

2)Author: Jaiminkumar Bhupeshkumar Desai

GitHub: https://github.com/jaiminbd13/jdesai567/tree/main/HW_2a

3)SUMMARY

Initial result

Testcase ID	Input	Expected Result	Actual result	Pass/Fail
test Right Triangle A	3,4,5	Right Triangle	Invalid Input	Fail
test Right Triangle B	5,3,4	Right Triangle	Invalid Input	Fail
test Equilateral Triangles	1,1,1	Equilateral	Invalid Input	Fail

Final result

Test Case ID	Input	Expected Result	Actual Result	Pass/Fail
test Equilateral Triangle 1	2,2,2	Equilateral	Equilateral	Pass
test Equilateral Triangles 2	1,1,1	Equilateral	Equilateral	Pass
Test Right Triangle 3	3,4,5	Right	Right	Pass
Test Right Triangle 4	5,3,4	Right	Right	Pass
test Scalene Triangle 5	3,6,7	Scalene	Scalene	Pass
test Scalene Triangle 6	11,14,23	Scalene	Scalene	Pass
test Nota Triangle 7	1,1,20	Not A Triangle	Not A Triangle	Pass
test Nota Triangle 8	5,5,25	Not A Triangle	Not A Triangle	Pass
Test Invalid Input 9	-5,-9,-1	Invalid Input	Invalid Input	Pass
Test Invalid Input 10	0,0,0	Invalid Input	Invalid Input	Pass
Test Invalid Input11	1.5,2.5,3.5	Invalid Input	Invalid Input	Pass
Test Isosceles Triangle 12	4,4,5	Isosceles	Isosceles	Pass
Test Isosceles Triangle 13	6,9,6	Isosceles	Isosceles	Pass

4)Matrix

	Test Run 1	Test Run 2	Test Run 3
Tests Planned	**3**	**12**	**13**
Tests Executed	**3**	**12**	**13**
Tests Passed	**0**	**12**	**13**
Defects Found	**3**	**0**	**0**
Defects Fixed	**0**	**0**	**0**

5) Honor pledge

"On my honor, I certify that I did not give or accept any unapproved help for this work. I further swear that, unless I specifically referenced the source, I did not copy any content from a book, article, the Internet, or any other source.