CSCS 351: Software Quality Assurance

Assignment 1

Name: Muhammad Muzammil

Roll Number: 231485432

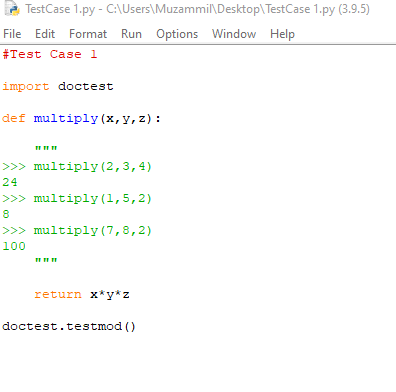
Section: A

Teacher: Saad Bin Saleem

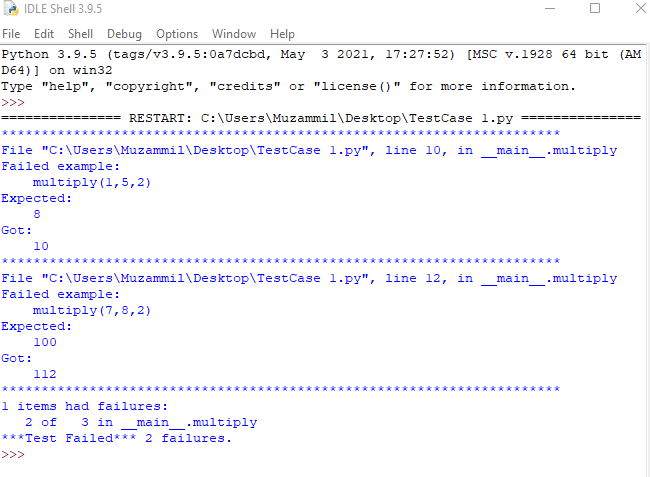
**Test 1:**

This test case uses unit testing to check for failures. It tests for 3 data sets and tell us how many are failures. Doctest is used in this code.

This is the code for the test case.

****

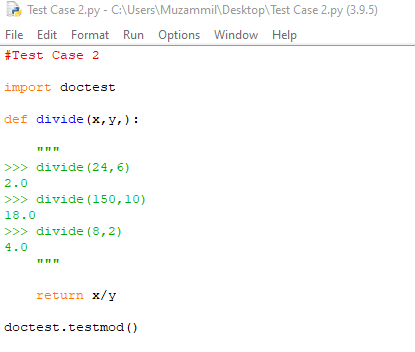
This is the result which tells us there are 2 failures. 2 data sets are not giving the right results.

****

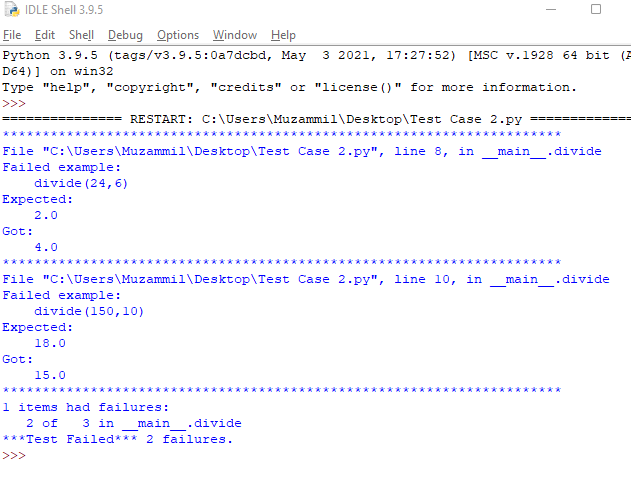
**Test 2:**

This test case uses unit testing to check for failures. It tests for 3 data sets and tell us how many are failures. Doctest is used in this code.

This is the code for the test case.

****

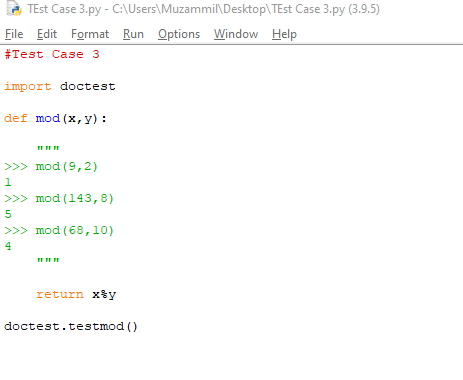
This is the result which tells us there are 2 failures. 2 data sets are not giving the right results.

****

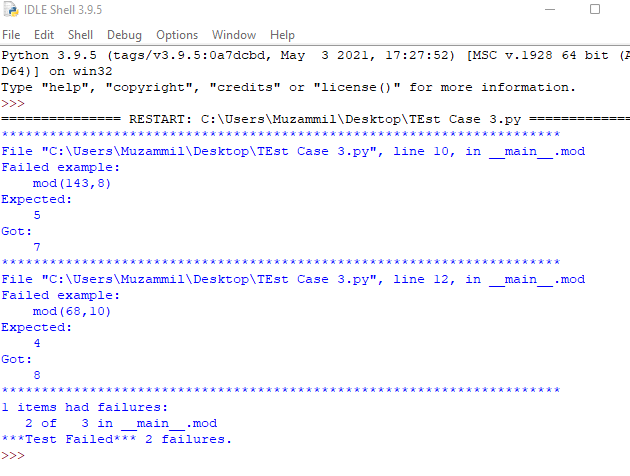
**Test 3:**

This test case uses unit testing to check for failures. It tests for 3 data sets and tell us how many are failures. Doctest is used in this code.

This is the code for the test case.

****

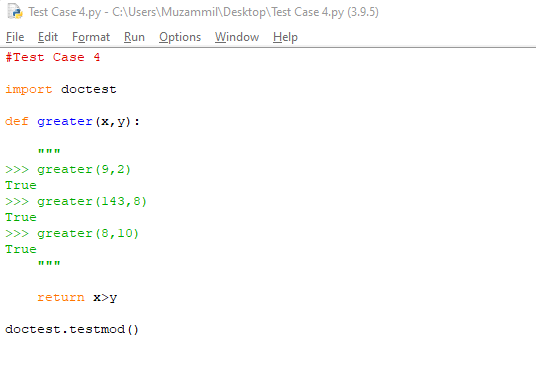
This is the result which tells us there are 2 failures. 2 data sets are not giving the right results.

****

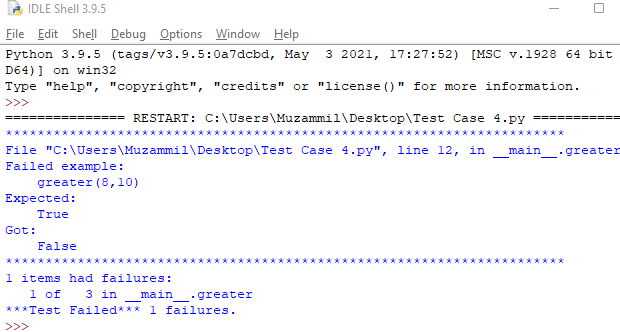
**Test 4:**

This test case uses unit testing to check for failures. It tests for 3 data sets and tell us how many are failures. Doctest is used in this code.

This is the code for the test case.

****

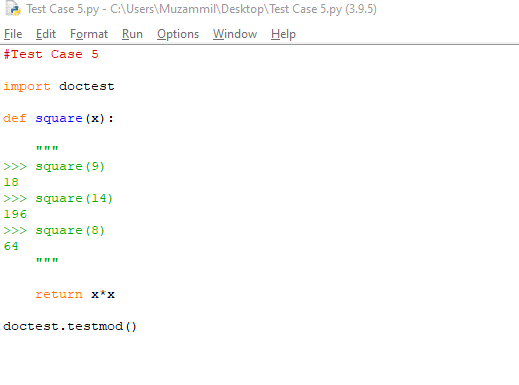
This is the result which tells us there are 1 failure1. 1 data set is not giving the right result.

****

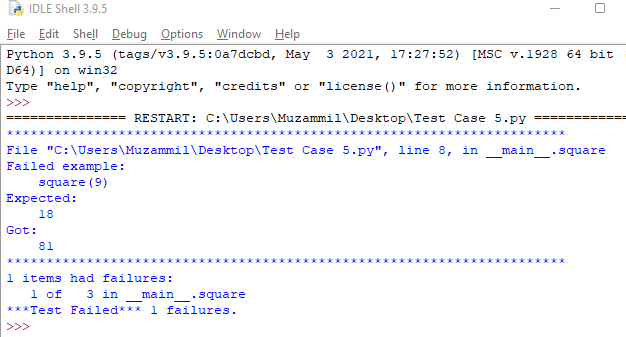
**Test 5:**

This test case uses unit testing to check for failures. It tests for 3 data sets and tell us how many are failures. Doctest is used in this code.

This is the code for the test case.

****

This is the result which tells us there are 1 failure. 1 data set is not giving the right result.

****

**Test Suite:**

**#Test Case 1**

import doctest

def multiply(x,y,z):

"""

>>> multiply(2,3,4)

24

>>> multiply(1,5,2)

8

>>> multiply(7,8,2)

100

"""

return x\*y\*z

doctest.testmod()

**#Test Case 2**

import doctest

def divide(x,y,):

"""

>>> divide(24,6)

2.0

>>> divide(150,10)

18.0

>>> divide(8,2)

4.0

"""

return x/y

doctest.testmod()

**#Test Case 3**

import doctest

def mod(x,y):

"""

>>> mod(9,2)

1

>>> mod(143,8)

5

>>> mod(68,10)

4

"""

return x%y

doctest.testmod()

**#Test Case 4**

import doctest

def greater(x,y):

"""

>>> greater(9,2)

True

>>> greater(143,8)

True

>>> greater(8,10)

True

"""

return x>y

doctest.testmod()

**#Test Case 5**

import doctest

def square(x):

"""

>>> square(9)

18

>>> square(14)

196

>>> square(8)

64

"""

return x\*x

doctest.testmod()

**This is the final result of all 5 tests.**

