Task 1. Create a new Python file named test\_math\_operations.py. Write at least three new test functions in this file to test basic arithmetic operations (e.g., multiplication, division, modulo).Ensure at least one test is designed to fail .Run only the tests in test\_math\_operations.py. 2. Modify your conftest.py file to add a new fixture called sample\_string. This fixture should yield a string like "Pytest is awesome!". Create a new test file named test\_string\_fixture.py. Write two test functions in test\_string\_fixture.py that use the sample\_string fixture:  One test should check if the string contains a specific substring (e.g., "awesome").  Another test should check the length of the string. Run these new tests and observe the fixture's setup and teardown messages (remember the -s flag). 3. In test\_math\_operations.py (from Task 1), ensure you have a failing test.Use PyCharm's debugging features:  Set a breakpoint on the assert line of the failing test.  Right-click the green play icon next to the failing test and select Debug 'test\_...'.  Step through the code and inspect variables to understand why the assertion is failing.  Fix the test so it passes.

Ans:

Test\_math\_operation.py:

|  |
| --- |
| import pytest  def test\_multiplication():  assert 5 \* 3 == 15, "Multiplication of 5 and 3 should equal 15"  def test\_division():  assert 10 / 2 == 5, "Division of 10 by 2 should equal 5"  def test\_modulo\_failing():  assert 10 % 3 == 2, "Modulo of 10 by 3 should equal 2" |

Test\_string\_fixture.py:

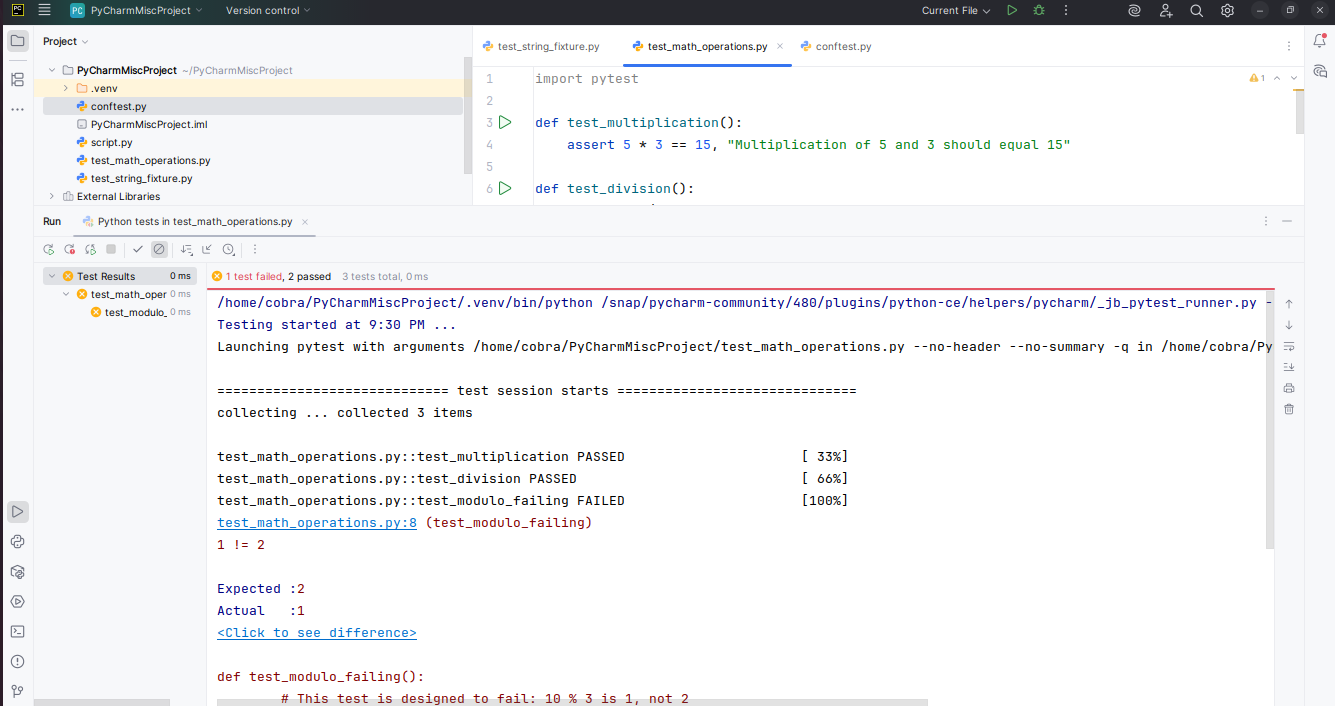
|  |
| --- |
| import pytest def test\_string\_contains\_substring(sample\_string):  assert "awesome" in sample\_string, "String should contain 'awesome'" def test\_string\_length(sample\_string):  assert len(sample\_string) == 17, "String length should be 17" |

Conftest.py:

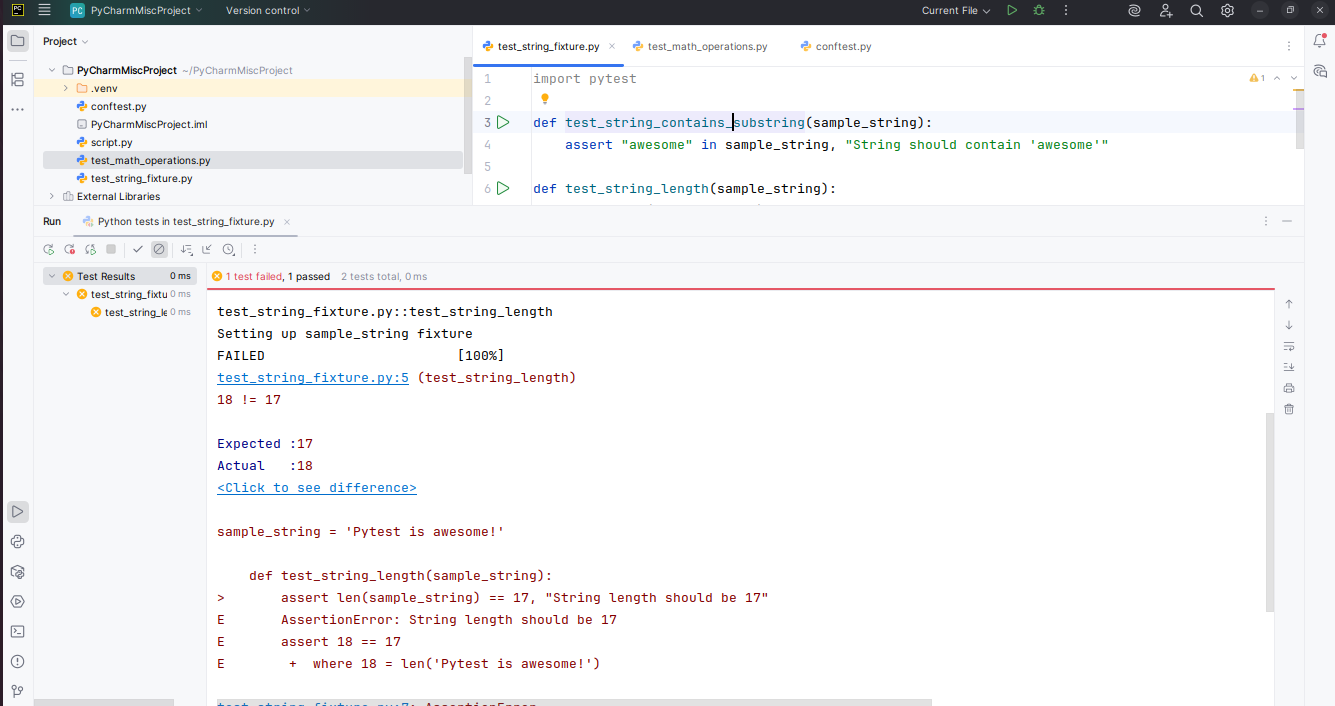
|  |
| --- |
| import pytest @pytest.fixture def sample\_string():  print("\nSetting up sample\_string fixture")  yield "Pytest is awesome!"  print("\nTearing down sample\_string fixture") |

OUTPUT:

A)Test\_math\_operation.py:



B)Test\_string\_fixture.py:



C)Conftest.py:

