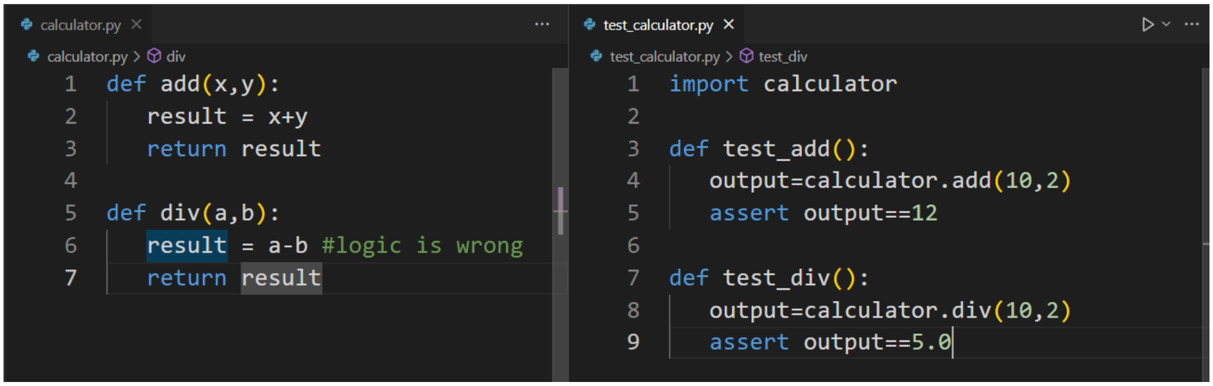
1. Install pytest
    a. Open cmd and install pytest as >pip install -U pytest
    b. Check version >pytest --version
2. creating come calculator related functions and testing it
    a. create calculator.py and add following code
    b. create testing file for pytest
        i. remember, for pyetst, your file name must be either start or ends with test like
            1. test_calculator.py or calculator_test.py

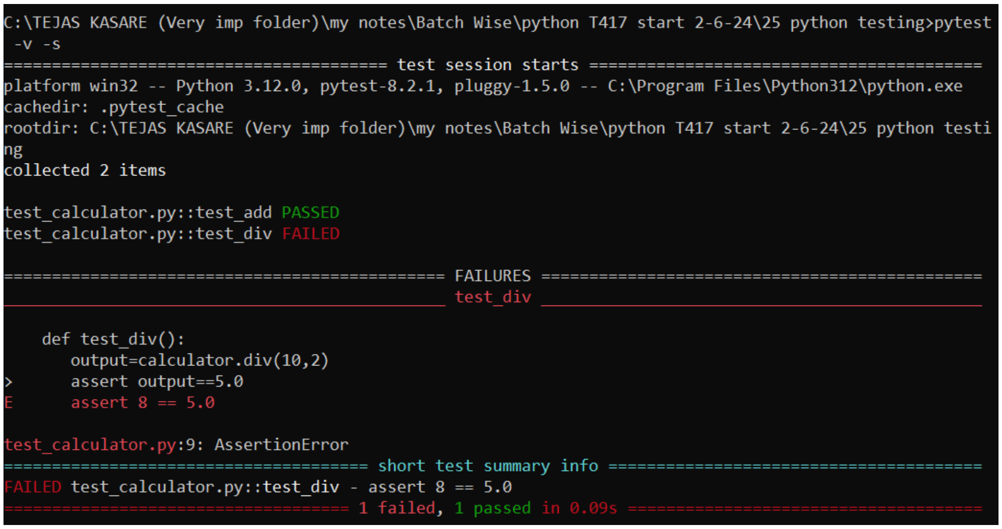


```python
def add(x,y):
    result = x+y
    return result

def div(a,b):
    result = a-b #logic is wrong
    return result
```

```python
import calculator

def test_add():
    output=calculator.add(10,2)
    assert output==12

def test_div():
    output=calculator.div(10,2)
    assert output==5.0
```

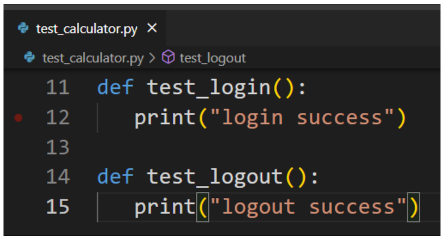
    c. running test file :
        i. open cmd on folder location and run any of the following command

>pytest
>pytest -v (v for verbose)
>pytest -v -s (s to show message that we write in print)



```
C:\TEJAS KASARE (Very imp folder)\my notes\Batch Wise\python T417 start 2-6-24\25 python testing>pytest
 -v -s
======================================= test session starts ========================================
platform win32 -- Python 3.12.0, pytest-8.2.1, pluggy-1.5.0 -- C:\Program Files\Python312\python.exe
cachedir: .pytest_cache
rootdir: C:\TEJAS KASARE (Very imp folder)\my notes\Batch Wise\python T417 start 2-6-24\25 python testi
ng
collected 2 items

test_calculator.py::test_add PASSED
test_calculator.py::test_div FAILED

============================================= FAILURES =============================================
_____________________________________________ test_div _____________________________________________

    def test_div():
       output=calculator.div(10,2)
>      assert output==5.0
E      assert 8 == 5.0

test_calculator.py:9: AssertionError
===================================== short test summary info ======================================
FAILED test_calculator.py::test_div - assert 8 == 5.0
==================================== 1 failed, 1 passed in 0.09s ===================================
```

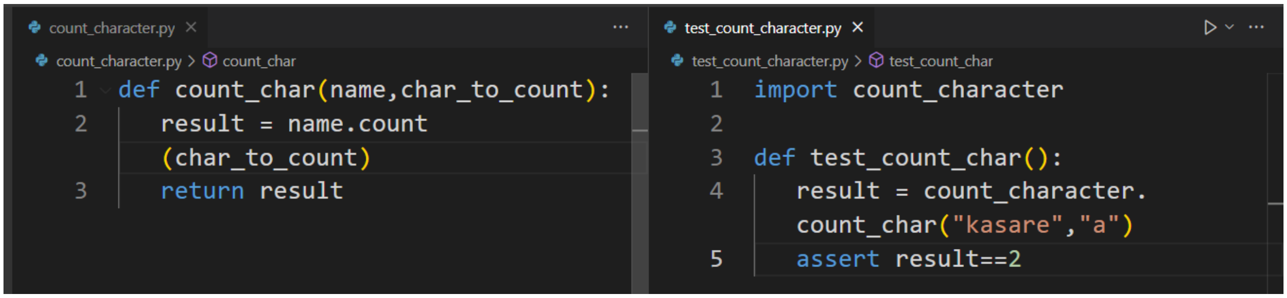
    d. running test file with keywords
        i. create 2 more functions in test_calculator.py and run it as



```python
def test_login():
    print("login success")

def test_logout():
    print("logout success")
```

>pytest –vsk log
K for keyword and log is name of that keyword

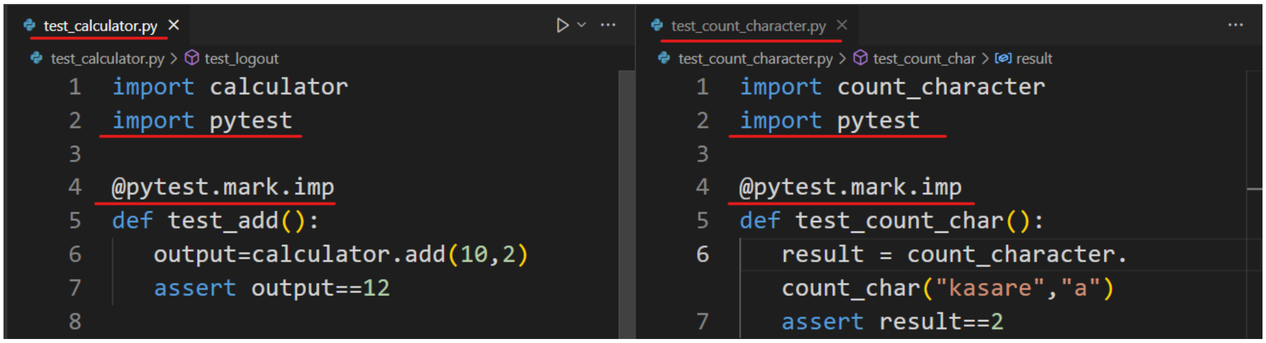
3. running test file in group by using custom markers
    a. create one more file with test file


count_character.py
count_character.py > count_char
test_count_character.py
test_count_character.py > test_count_char

```python
def count_char(name,char_to_count):
    result = name.count(char_to_count)
    return result
```

```python
import count_character

def test_count_char():
    result = count_character.count_char("kasare","a")
    assert result==2
```

    b. add custom markers on some functions in test_calcultor.py and test_count_character.py


test_calculator.py
test_calculator.py > test_logout
test_count_character.py
test_count_character.py > test_count_char > result

```python
import calculator
import pytest

@pytest.mark.imp
def test_add():
    output=calculator.add(10,2)
    assert output==12

```

```python
import count_character
import pytest

@pytest.mark.imp
def test_count_char():
    result = count_character.count_char("kasare","a")
    assert result==2
```

    c. run test as >pytest -vsm imp
    m for marker and imp is the name of marker

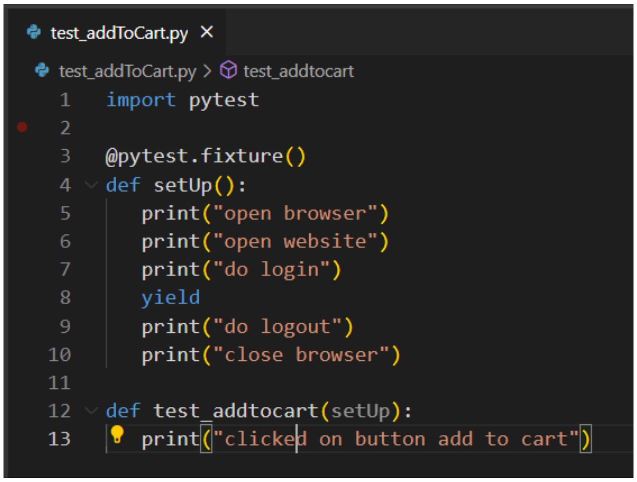
4. The Complete List Of Python Assert Statements :
    a. Click below link to get most common assert in pytest
    https://understandingdata.com/posts/list-of-python-assert-statements-for-unit-tests/

5. fixture
    a. consider the following flow to test add to cart button :

        i. open browser --------------------------
        ii. open website -------------------------- Arrange
        iii. do login ----------------------------------
        iv. click on add to cart button ------------ Action
        v. verify product is added to the cart-- Assert
        vi. do logout --------------------------------- Cleanup
        vii. close browser ---------------------------

    Arrange and cleanup together called as fixture

    b. create test_addToCart.py and add following code



```python
import pytest

@pytest.fixture()
def setUp():
    print("open browser")
    print("open website")
    print("do login")
    yield
    print("do logout")
    print("close browser")

def test_addtocart(setUp):
    print("clicked on button add to cart")
```

    c. run it as > pytest test_addToCart.py –vs

6. DONE !!!!!!