

## Unit 6.2 Graded Assignment:

Syed Muhammad Raqim Ali Shah (2303.KHI.DEG.008)

Maaz Javaid Siddique (2303.KHI.DEG.004)

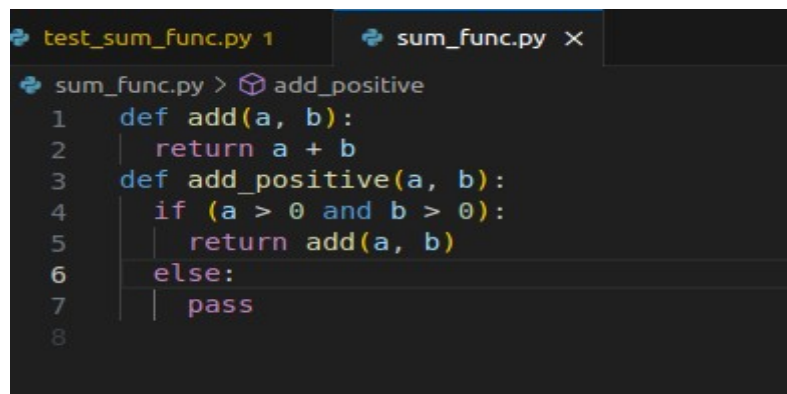
Qadeer Hussain (2303.KHI.DEG.006)

### Daily Assignment :

Implement tests for  
tasks/6\_writing\_robust\_readable\_and\_maintainable\_code/day\_2\_testing/assignment  
my\_proj project (there is a tests/test\_sum\_func.py file ready, missing the tests  
only). Aim for 100% code coverage, branch coverage and condition coverage.  
Run the tests with pytest. Then run the tests with a flag producing coverage  
reports (you'll need to install pytest-cov at <https://pypi.org/project/pytest-cov/>).

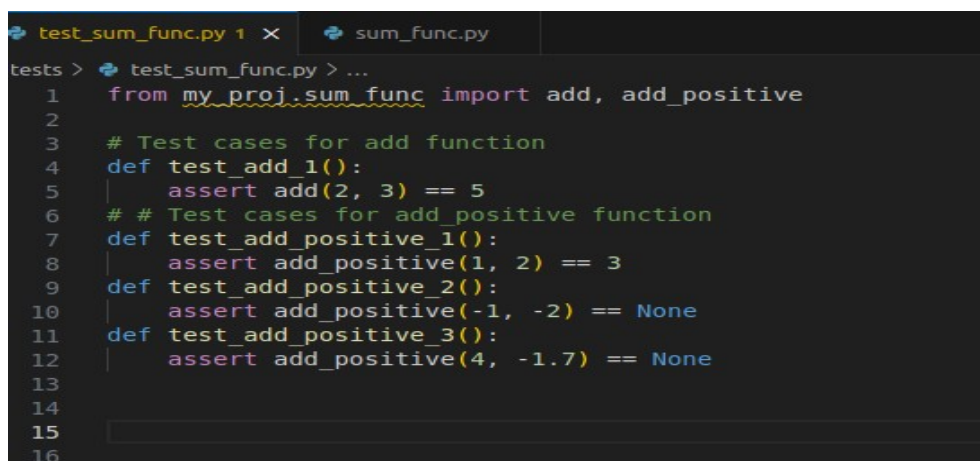
### Answer:

sum\_func.py

A screenshot of a code editor showing the sum\_func.py file. The code defines two functions: add(a, b) which returns a + b, and add\_positive(a, b) which returns add(a, b) if both a and b are greater than 0, and None otherwise. The code is as follows:

```
1 def add(a, b):
2     return a + b
3 def add_positive(a, b):
4     if (a > 0 and b > 0):
5         return add(a, b)
6     else:
7         pass
8
```

test\_sum\_func.py

A screenshot of a code editor showing the test\_sum\_func.py file. The code imports add and add\_positive from my\_proj.sum\_func and defines four test functions: test\_add\_1(), test\_add\_positive\_1(), test\_add\_positive\_2(), and test\_add\_positive\_3(). The code is as follows:

```
1 from my_proj.sum_func import add, add_positive
2
3 # Test cases for add function
4 def test_add_1():
5     assert add(2, 3) == 5
6
7 # Test cases for add_positive function
8 def test_add_positive_1():
9     assert add_positive(1, 2) == 3
10 def test_add_positive_2():
11     assert add_positive(-1, -2) == None
12 def test_add_positive_3():
13     assert add_positive(4, -1.7) == None
14
15
16
```

The command “pytest .” is used to run tests using the pytest framework.

This command “pytest --cov=my\_proj tests/” is used to run tests with code coverage analysis using the pytest-cov plugin

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
• (base) qadeerhussain@all-MS-7D35:~/Qadeer/Training/module_6/6.2_day_2_testing/6.2_Assignment/my_proj$ pytest .
===== test session starts =====
platform linux -- Python 3.10.9, pytest-7.3.1, pluggy-1.0.0
rootdir: /home/qadeerhussain/Qadeer/Training/module_6/6.2_day_2_testing/6.2_Assignment/my_proj
plugins: dash-2.9.3, cov-4.0.0, anyio-3.6.2
collected 4 items

tests/test_sum_func.py ....

[100%]

===== 4 passed in 0.02s =====
• (base) qadeerhussain@all-MS-7D35:~/Qadeer/Training/module_6/6.2_day_2_testing/6.2_Assignment/my_proj$ pytest --cov=my_proj tests/
===== test session starts =====
platform linux -- Python 3.10.9, pytest-7.3.1, pluggy-1.0.0
rootdir: /home/qadeerhussain/Qadeer/Training/module_6/6.2_day_2_testing/6.2_Assignment/my_proj
plugins: dash-2.9.3, cov-4.0.0, anyio-3.6.2
collected 4 items

tests/test_sum_func.py ....

[100%]

----- coverage: platform linux, python 3.10.9-final-0 -----
Name                               Stmts   Miss  Cover
-----
__init__.py                         0      0   100%
sum_func.py                         6      0   100%
tests/__init__.py                   0      0   100%
tests/test_sum_func.py              9      0   100%
TOTAL                             15      0   100%

===== 4 passed in 0.05s =====
○ (base) qadeerhussain@all-MS-7D35:~/Qadeer/Training/module_6/6.2_day_2_testing/6.2_Assignment/my_proj$
```

Based on the output, it appears that all the tests in the test\_sum\_func.py file have passed successfully, and the coverage report shows 100% coverage for the relevant files in the project.