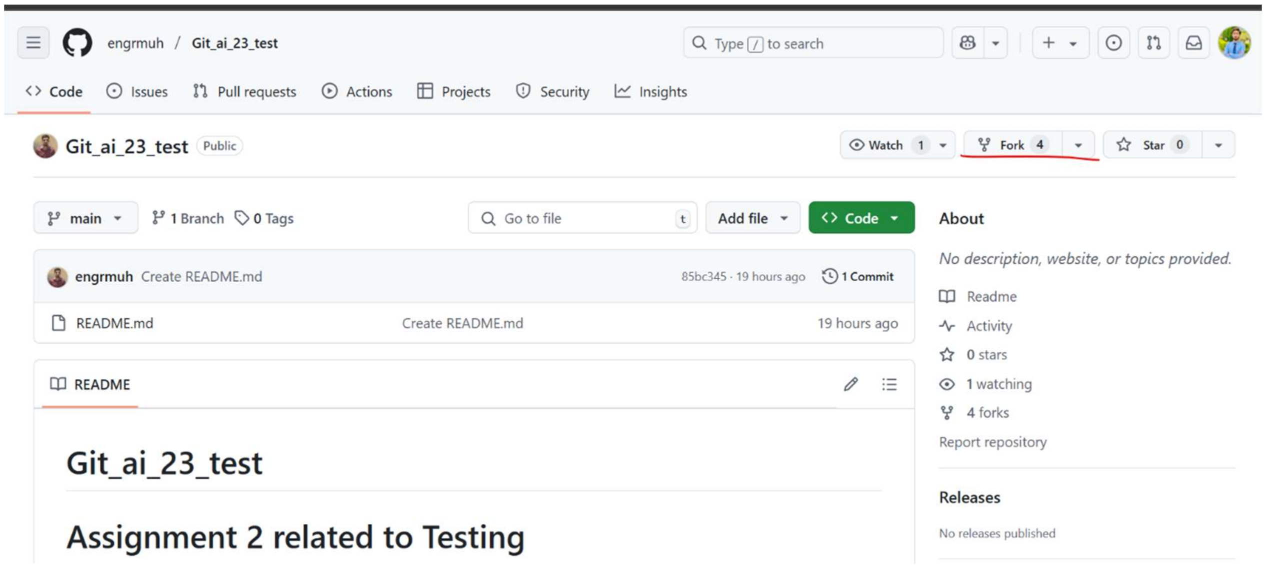

engrmuh / Git_ai_23_test

Code · Issues · Pull requests · Actions · Projects · Security · Insights

Git_ai_23_test Public

Watch 1 · Fork 4 · Star 0

main · 1 Branch · 0 Tags · Go to file · Add file · Code

engrmuh Create README.md 85bc345 · 19 hours ago · 1 Commit

README.md Create README.md 19 hours ago

README

Git_ai_23_test

Assignment 2 related to Testing

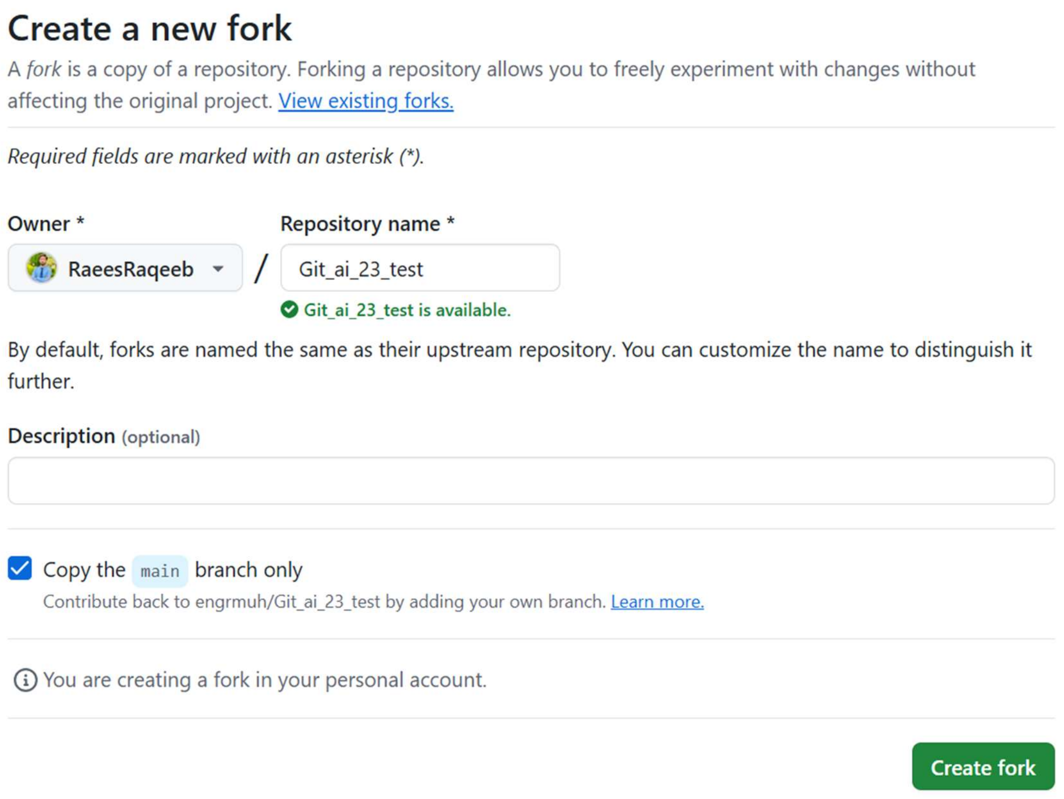
About

No description, website, or topics provided.

Readme · Activity · 0 stars · 1 watching · 4 forks · Report repository

Releases

No releases published


Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.

Required fields are marked with an asterisk (*).

Owner * RaeesRaqeeb / Repository name * Git_ai_23_test

Git_ai_23_test is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

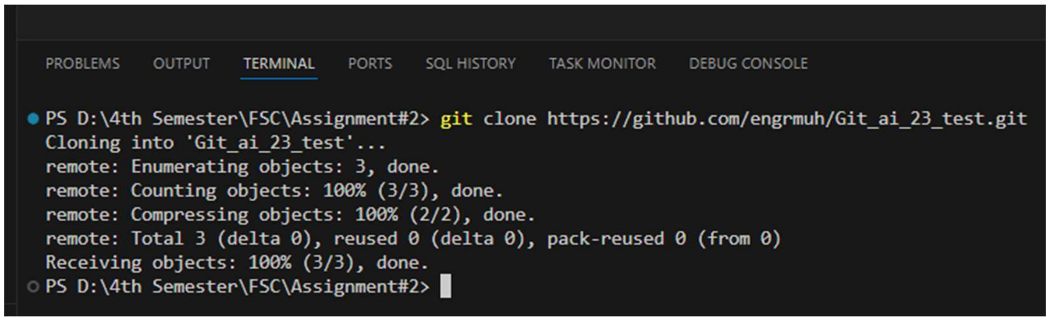
Description (optional)

[✓] Copy the `main` branch only

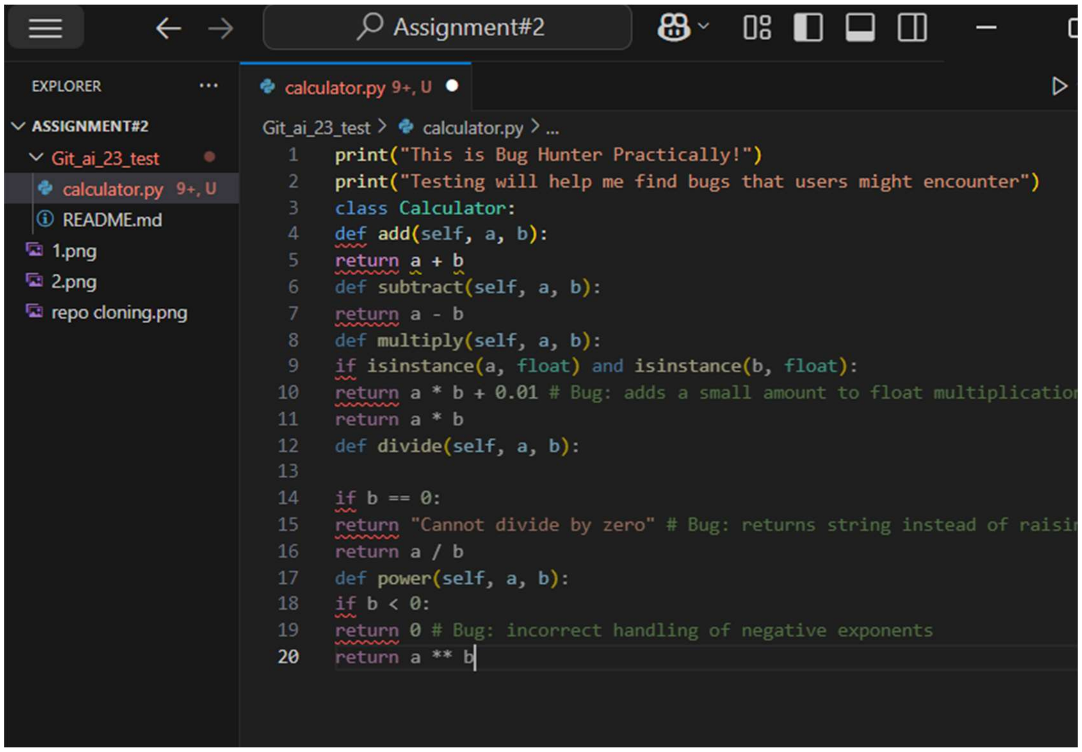
Contribute back to engrmuh/Git_ai_23_test by adding your own branch. Learn more.

You are creating a fork in your personal account.

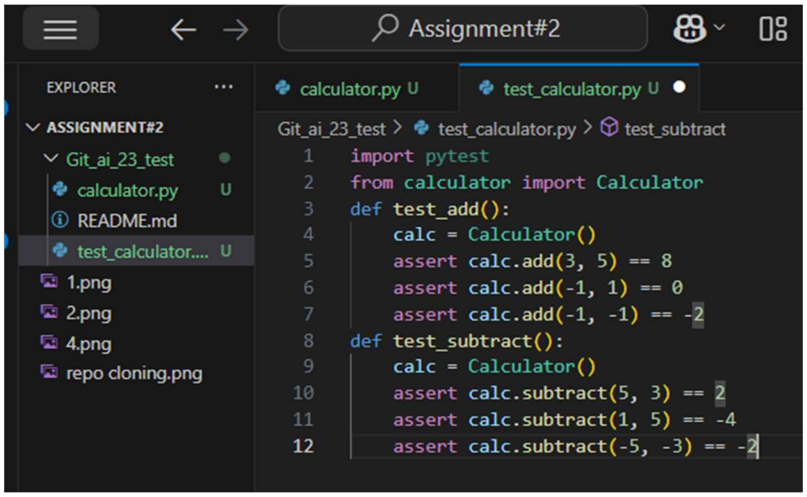
Create fork



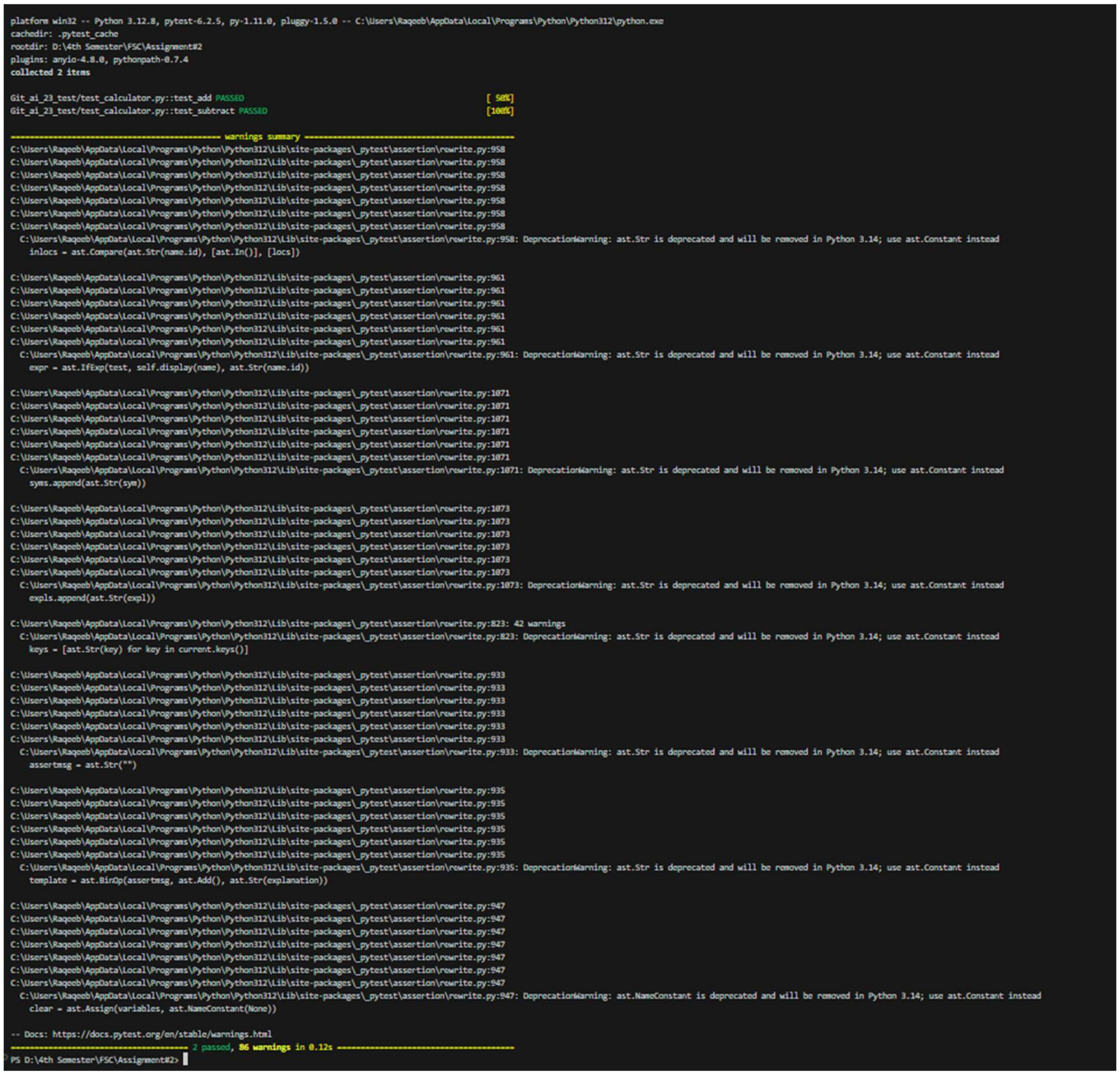
```
PS D:\4th Semester\FSC\Assignment#2> git clone https://github.com/engrmuh/Git_ai_23_test.git
Cloning into 'Git_ai_23_test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
PS D:\4th Semester\FSC\Assignment#2>
```



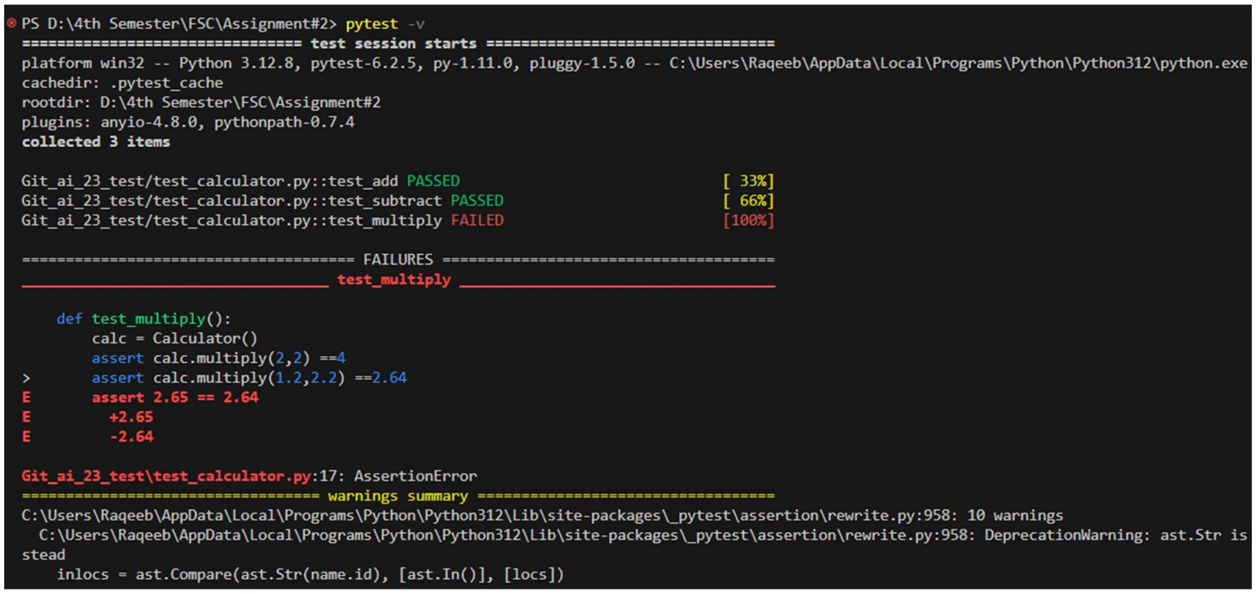
```
print("This is Bug Hunter Practically!")
print("Testing will help me find bugs that users might encounter")
class Calculator:
def add(self, a, b):
return a + b
def subtract(self, a, b):
return a - b
def multiply(self, a, b):
if isinstance(a, float) and isinstance(b, float):
return a * b + 0.01 # Bug: adds a small amount to float multiplicatio
return a * b
def divide(self, a, b):

if b == 0:
return "Cannot divide by zero" # Bug: returns string instead of raisi
return a / b
def power(self, a, b):
if b < 0:
return 0 # Bug: incorrect handling of negative exponents
return a ** b
```



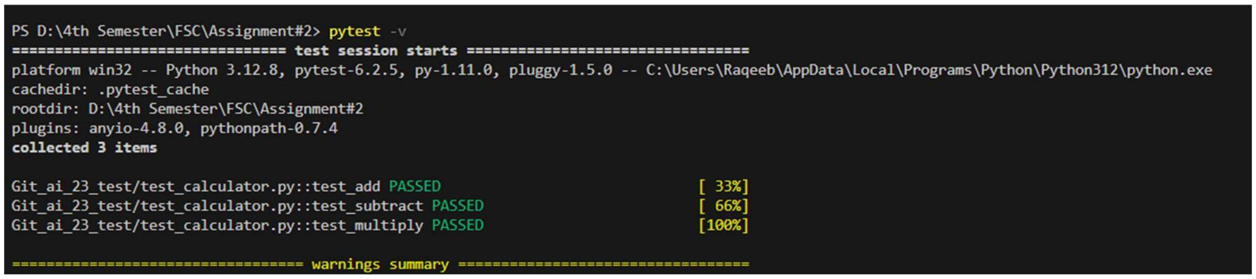
```python
import pytest
from calculator import Calculator
def test_add():
    calc = Calculator()
    assert calc.add(3, 5) == 8
    assert calc.add(-1, 1) == 0
    assert calc.add(-1, -1) == -2
def test_subtract():
    calc = Calculator()
    assert calc.subtract(5, 3) == 2
    assert calc.subtract(1, 5) == -4
    assert calc.subtract(-5, -3) == -2
```



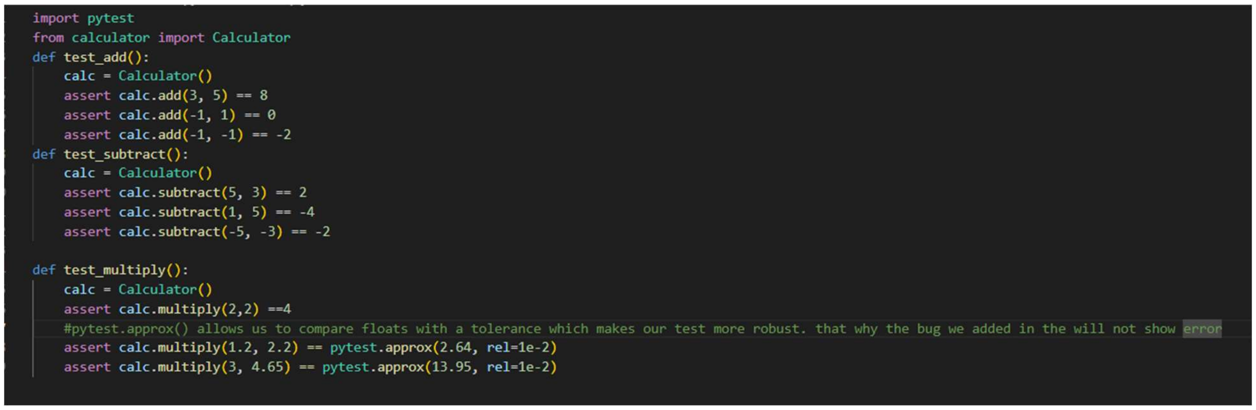
```
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collected 2 items

Git_ai_23_test/test_calculator.py::test_add PASSED                                                         [ 50%]
Git_ai_23_test/test_calculator.py::test_subtract PASSED                                                    [100%]

============================================ warnings summary ============================================
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    inlocs = ast.Compare(ast.Str(name.id), [ast.In()], [locs])

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    expr = ast.IfExp(test, self.display(name), ast.Str(name.id))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    syms.append(ast.Str(sym))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    expls.append(ast.Str(expl))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:823: 42 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:823: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    keys = [ast.Str(key) for key in current.keys()]

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    assertmsg = ast.Str("")

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:935
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:935
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:935
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:935
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:935
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:935
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:935: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    template = ast.BinOp(assertmsg, ast.Add(), ast.Str(explanation))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:947: DeprecationWarning: ast.NameConstant is deprecated and will be removed in Python 3.14; use ast.Constant instead
    clear = ast.Assign(variables, ast.NameConstant(None))

-- Docs: https://docs.pytest.org/en/stable/warnings.html
====================================== 2 passed, 86 warnings in 0.12s ======================================
PS D:\4th Semester\FSC\Assignment#2> 
```



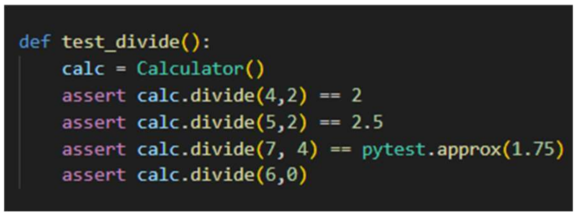
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
=============================== test session starts ================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collected 3 items

Git_ai_23_test/test_calculator.py::test_add PASSED                         [ 33%]
Git_ai_23_test/test_calculator.py::test_subtract PASSED                    [ 66%]
Git_ai_23_test/test_calculator.py::test_multiply FAILED                    [100%]

===================================== FAILURES =====================================
___________________________________ test_multiply ____________________________________

    def test_multiply():
        calc = Calculator()
        assert calc.multiply(2,2) ==4
>       assert calc.multiply(1.2,2.2) ==2.64
E       assert 2.65 == 2.64
E         +2.65
E         -2.64

Git_ai_23_test\test_calculator.py:17: AssertionError
================================= warnings summary =================================
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: 10 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: DeprecationWarning: ast.Str is
stead
    inlocs = ast.Compare(ast.Str(name.id), [ast.In()], [locs])
```



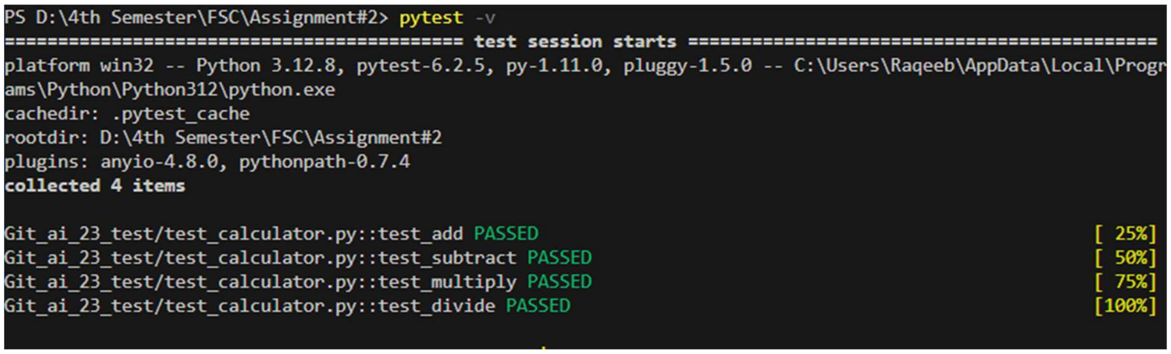
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
=============================== test session starts ================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collected 3 items

Git_ai_23_test/test_calculator.py::test_add PASSED                         [ 33%]
Git_ai_23_test/test_calculator.py::test_subtract PASSED                    [ 66%]
Git_ai_23_test/test_calculator.py::test_multiply PASSED                    [100%]

================================= warnings summary =================================
```



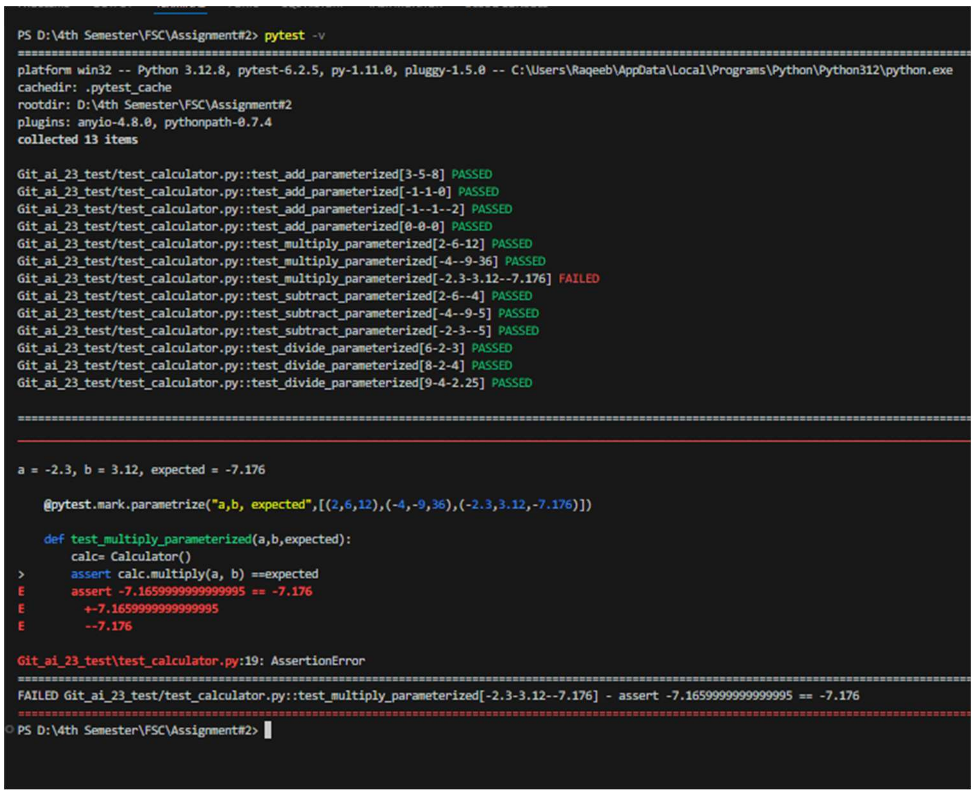
```
import pytest
from calculator import Calculator
def test_add():
    calc = Calculator()
    assert calc.add(3, 5) == 8
    assert calc.add(-1, 1) == 0
    assert calc.add(-1, -1) == -2
def test_subtract():
    calc = Calculator()
    assert calc.subtract(5, 3) == 2
    assert calc.subtract(1, 5) == -4
    assert calc.subtract(-5, -3) == -2

def test_multiply():
    calc = Calculator()
    assert calc.multiply(2,2) ==4
    #pytest.approx() allows us to compare floats with a tolerance which makes our test more robust. that why the bug we added in the will not show error
    assert calc.multiply(1.2, 2.2) == pytest.approx(2.64, rel=1e-2)
    assert calc.multiply(3, 4.65) == pytest.approx(13.95, rel=1e-2)
```



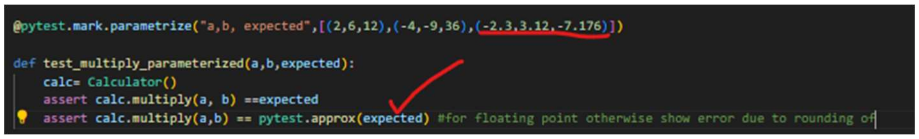
```
def test_divide():
    calc = Calculator()
    assert calc.divide(4,2) == 2
    assert calc.divide(5,2) == 2.5
    assert calc.divide(7, 4) == pytest.approx(1.75)
    assert calc.divide(6,0)
```



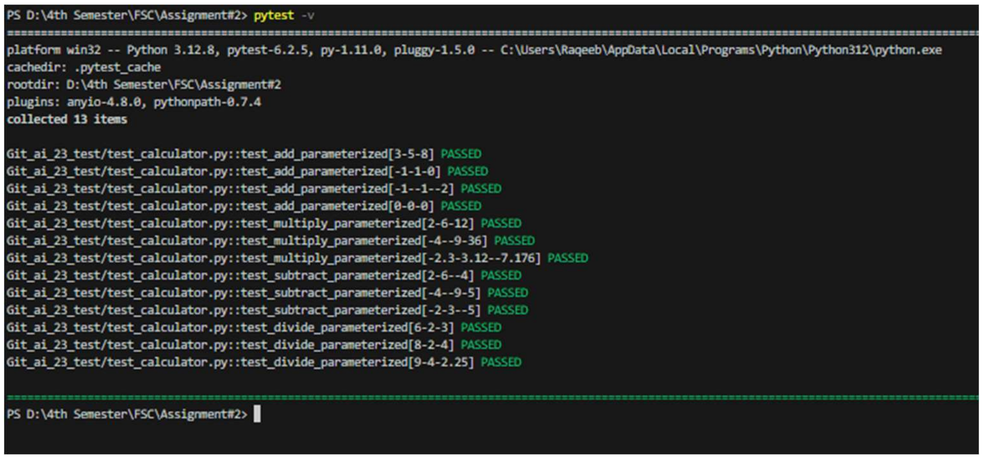
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
============================================ test session starts ============================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Progr
ams\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collected 4 items

Git_ai_23_test/test_calculator.py::test_add PASSED                                                   [ 25%]
Git_ai_23_test/test_calculator.py::test_subtract PASSED                                              [ 50%]
Git_ai_23_test/test_calculator.py::test_multiply PASSED                                              [ 75%]
Git_ai_23_test/test_calculator.py::test_divide PASSED                                                [100%]
```



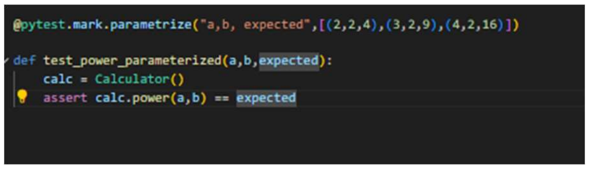
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
==========================================================================================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collected 13 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] FAILED
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED

==========================================================================================================

a = -2.3, b = 3.12, expected = -7.176

    @pytest.mark.parametrize("a,b, expected",[(2,6,12),(-4,-9,36),(-2.3,3.12,-7.176)])

    def test_multiply_parameterized(a,b,expected):
        calc= Calculator()
>       assert calc.multiply(a, b) ==expected
E       assert -7.1659999999999995 == -7.176
E         +-7.1659999999999995
E         --7.176

Git_ai_23_test\test_calculator.py:19: AssertionError
==========================================================================================================
FAILED Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] - assert -7.1659999999999995 == -7.176
==========================================================================================================
PS D:\4th Semester\FSC\Assignment#2>
```



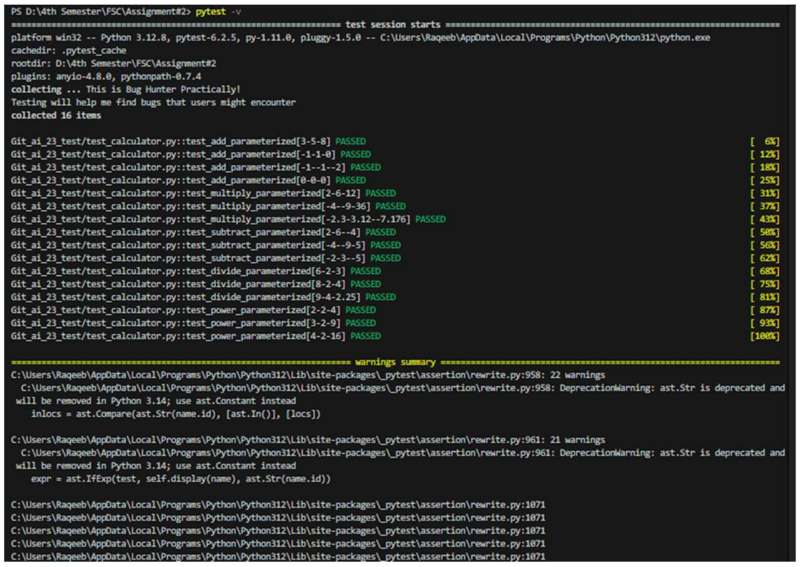
```python
@pytest.mark.parametrize("a,b, expected",[(2,6,12),(-4,-9,36),(-2.3,3.12,-7.176)])

def test_multiply_parameterized(a,b,expected):
    calc= Calculator()
    assert calc.multiply(a, b) ==expected
    assert calc.multiply(a,b) == pytest.approx(expected) #for floating point otherwise show error due to rounding of
```



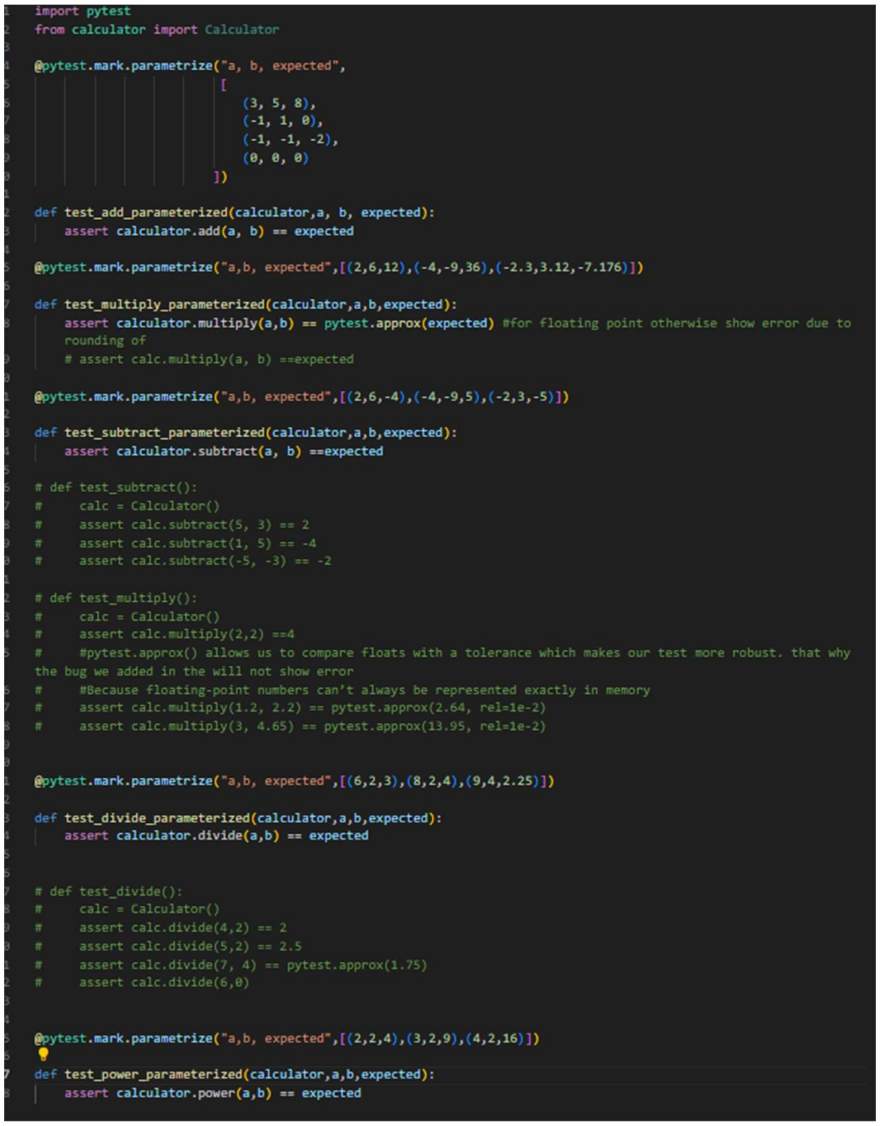
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
==========================================================================================================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collected 13 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] PASSED
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED

==========================================================================================================================
PS D:\4th Semester\FSC\Assignment#2>
```



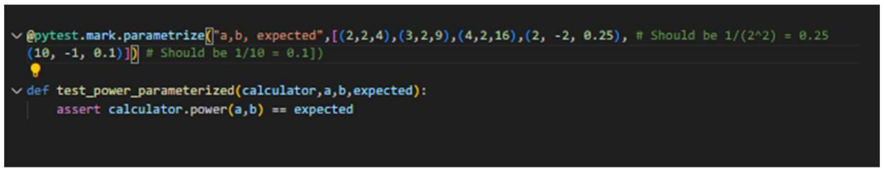
```python
@pytest.mark.parametrize("a,b, expected",[(2,2,4),(3,2,9),(4,2,16)])

def test_power_parameterized(a,b,expected):
    calc = Calculator()
    assert calc.power(a,b) == expected
```



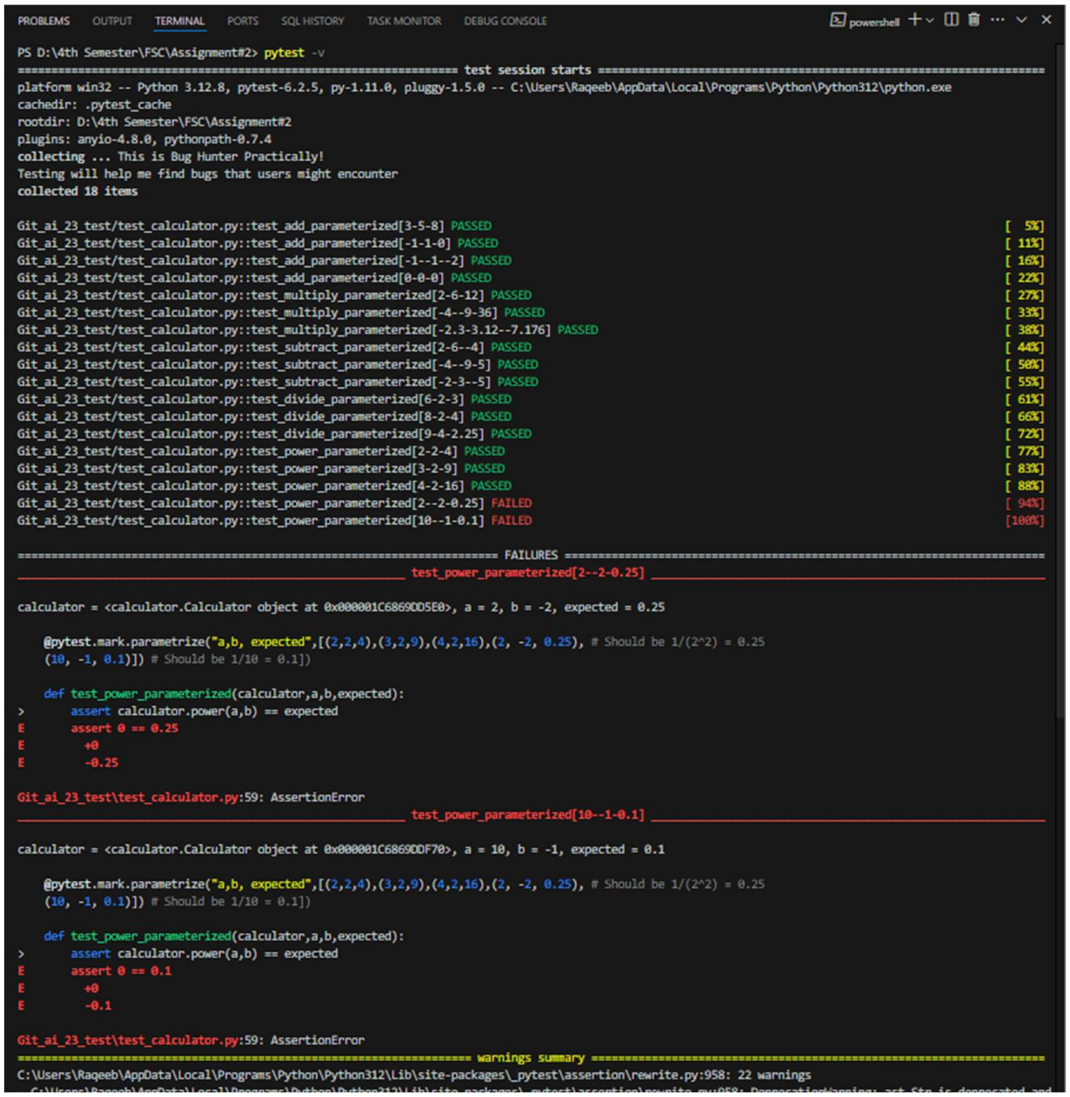
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
=================================================================== test session starts ===================================================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collecting ... This is Bug Hunter Practically!
Testing will help me find bugs that users might encounter
collected 16 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED                                                                        [  6%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED                                                                       [ 12%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED                                                                     [ 18%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED                                                                        [ 25%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED                                                                  [ 31%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED                                                                [ 37%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] PASSED                                                        [ 43%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED                                                                  [ 50%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED                                                                 [ 56%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED                                                                 [ 62%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED                                                                     [ 68%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED                                                                     [ 75%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED                                                                  [ 81%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2-2-4] PASSED                                                                      [ 87%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[3-2-9] PASSED                                                                      [ 93%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[4-2-16] PASSED                                                                     [100%]

==================================================================== warnings summary =====================================================================
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: 22 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: DeprecationWarning: ast.Str is deprecated and
 will be removed in Python 3.14; use ast.Constant instead
    inlocs = ast.Compare(ast.Str(name.id), [ast.In()], [locs])

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961: 21 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961: DeprecationWarning: ast.Str is deprecated and
 will be removed in Python 3.14; use ast.Constant instead
    expr = ast.IfExp(test, self.display(name), ast.Str(name.id))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
```



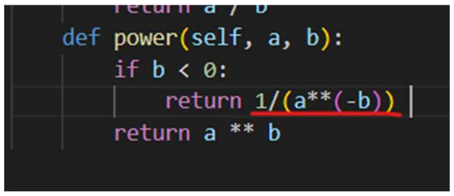
```python
import pytest
from calculator import Calculator

@pytest.mark.parametrize("a, b, expected",
                         [
                            (3, 5, 8),
                            (-1, 1, 0),
                            (-1, -1, -2),
                            (0, 0, 0)
                         ])

def test_add_parameterized(calculator,a, b, expected):
    assert calculator.add(a, b) == expected

@pytest.mark.parametrize("a,b, expected",[(2,6,12),(-4,-9,36),(-2.3,3.12,-7.176)])

def test_multiply_parameterized(calculator,a,b,expected):
    assert calculator.multiply(a,b) == pytest.approx(expected) #for floating point otherwise show error due to
    rounding of
    # assert calc.multiply(a, b) ==expected

@pytest.mark.parametrize("a,b, expected",[(2,6,-4),(-4,-9,5),(-2,3,-5)])

def test_subtract_parameterized(calculator,a,b,expected):
    assert calculator.subtract(a, b) ==expected

# def test_subtract():
#     calc = Calculator()
#     assert calc.subtract(5, 3) == 2
#     assert calc.subtract(1, 5) == -4
#     assert calc.subtract(-5, -3) == -2

# def test_multiply():
#     calc = Calculator()
#     assert calc.multiply(2,2) ==4
#     #pytest.approx() allows us to compare floats with a tolerance which makes our test more robust. that why
the bug we added in the will not show error
#     #Because floating-point numbers can't always be represented exactly in memory
#     assert calc.multiply(1.2, 2.2) == pytest.approx(2.64, rel=1e-2)
#     assert calc.multiply(3, 4.65) == pytest.approx(13.95, rel=1e-2)


@pytest.mark.parametrize("a,b, expected",[(6,2,3),(8,2,4),(9,4,2.25)])

def test_divide_parameterized(calculator,a,b,expected):
    assert calculator.divide(a,b) == expected


# def test_divide():
#     calc = Calculator()
#     assert calc.divide(4,2) == 2
#     assert calc.divide(5,2) == 2.5
#     assert calc.divide(7, 4) == pytest.approx(1.75)
#     assert calc.divide(6,0)


@pytest.mark.parametrize("a,b, expected",[(2,2,4),(3,2,9),(4,2,16)])

def test_power_parameterized(calculator,a,b,expected):
    assert calculator.power(a,b) == expected
```



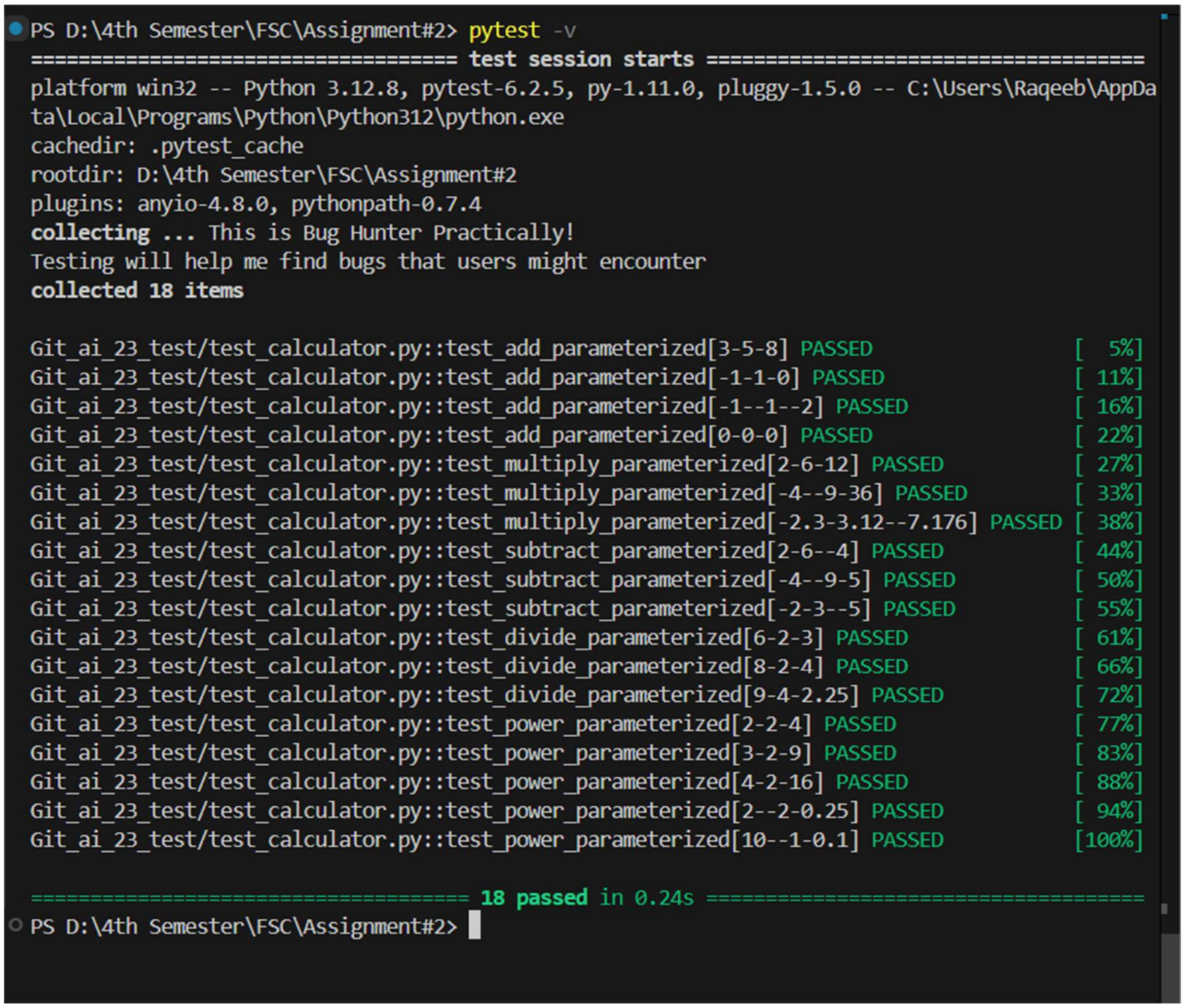
```python
@pytest.mark.parametrize("a,b, expected",[(2,2,4),(3,2,9),(4,2,16),(2, -2, 0.25), # Should be 1/(2^2) = 0.25
(10, -1, 0.1)]) # Should be 1/10 = 0.1])

def test_power_parameterized(calculator,a,b,expected):
    assert calculator.power(a,b) == expected
```



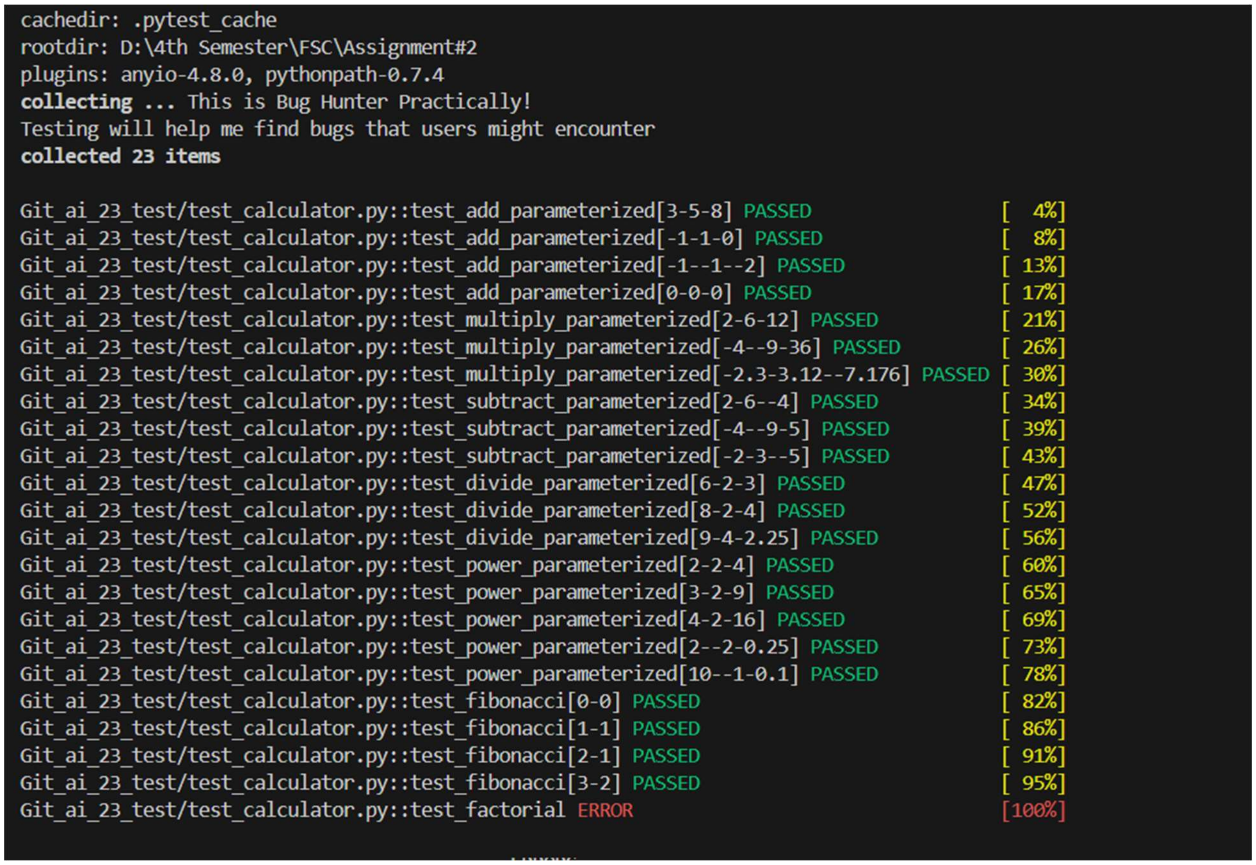
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
=================================================================== test session starts ====================================================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collecting ... This is Bug Hunter Practically!
Testing will help me find bugs that users might encounter
collected 18 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED                                                                          [  5%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED                                                                         [ 11%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED                                                                       [ 16%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED                                                                          [ 22%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED                                                                    [ 27%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED                                                                  [ 33%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] PASSED                                                          [ 38%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED                                                                    [ 44%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED                                                                   [ 50%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED                                                                   [ 55%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED                                                                       [ 61%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED                                                                       [ 66%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED                                                                    [ 72%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2-2-4] PASSED                                                                        [ 77%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[3-2-9] PASSED                                                                        [ 83%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[4-2-16] PASSED                                                                       [ 88%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2--2-0.25] FAILED                                                                    [ 94%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[10--1-0.1] FAILED                                                                    [100%]

========================================================================= FAILURES =========================================================================
___________________________________________________________ test_power_parameterized[2--2-0.25] ____________________________________________________________

calculator = <calculator.Calculator object at 0x000001C6869DD5E0>, a = 2, b = -2, expected = 0.25

    @pytest.mark.parametrize("a,b, expected",[(2,2,4),(3,2,9),(4,2,16),(2, -2, 0.25), # Should be 1/(2^2) = 0.25
    (10, -1, 0.1)]) # Should be 1/10 = 0.1])

    def test_power_parameterized(calculator,a,b,expected):
>       assert calculator.power(a,b) == expected
E       assert 0 == 0.25
E         +0
E         -0.25

Git_ai_23_test\test_calculator.py:59: AssertionError
___________________________________________________________ test_power_parameterized[10--1-0.1] ____________________________________________________________

calculator = <calculator.Calculator object at 0x000001C6869DDF70>, a = 10, b = -1, expected = 0.1

    @pytest.mark.parametrize("a,b, expected",[(2,2,4),(3,2,9),(4,2,16),(2, -2, 0.25), # Should be 1/(2^2) = 0.25
    (10, -1, 0.1)]) # Should be 1/10 = 0.1])

    def test_power_parameterized(calculator,a,b,expected):
>       assert calculator.power(a,b) == expected
E       assert 0 == 0.1
E         +0
E         -0.1

Git_ai_23_test\test_calculator.py:59: AssertionError
===================================================================== warnings summary =====================================================================
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: 22 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: DeprecationWarning: ast.Str is deprecated and
```



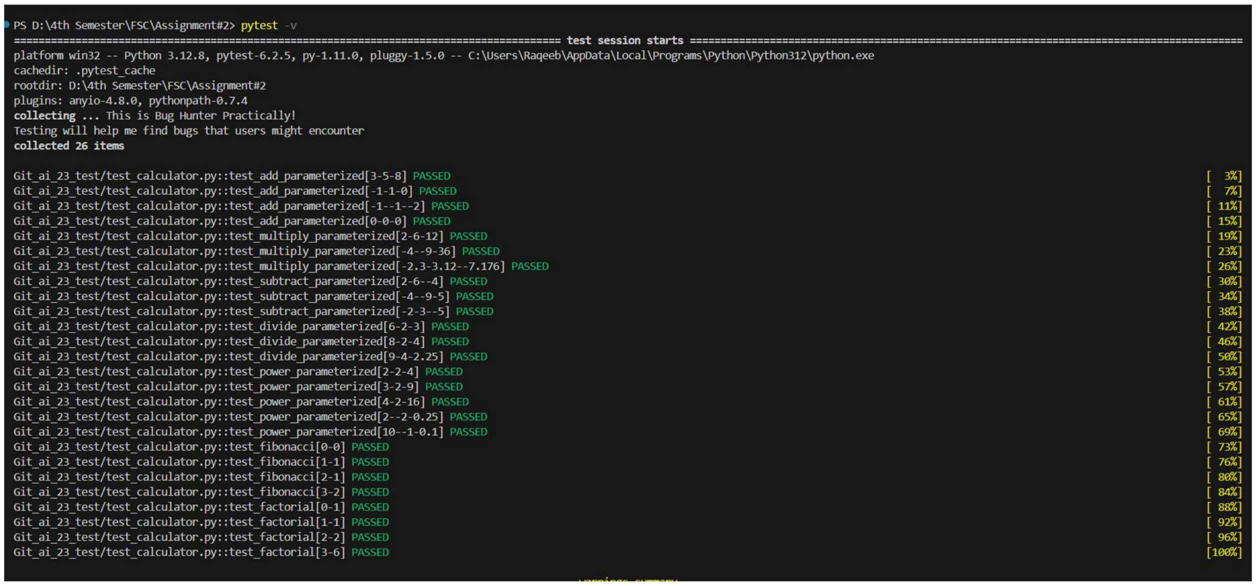
```python
        return a / b
    def power(self, a, b):
        if b < 0:
            return 1/(a**(-b))
        return a ** b
```



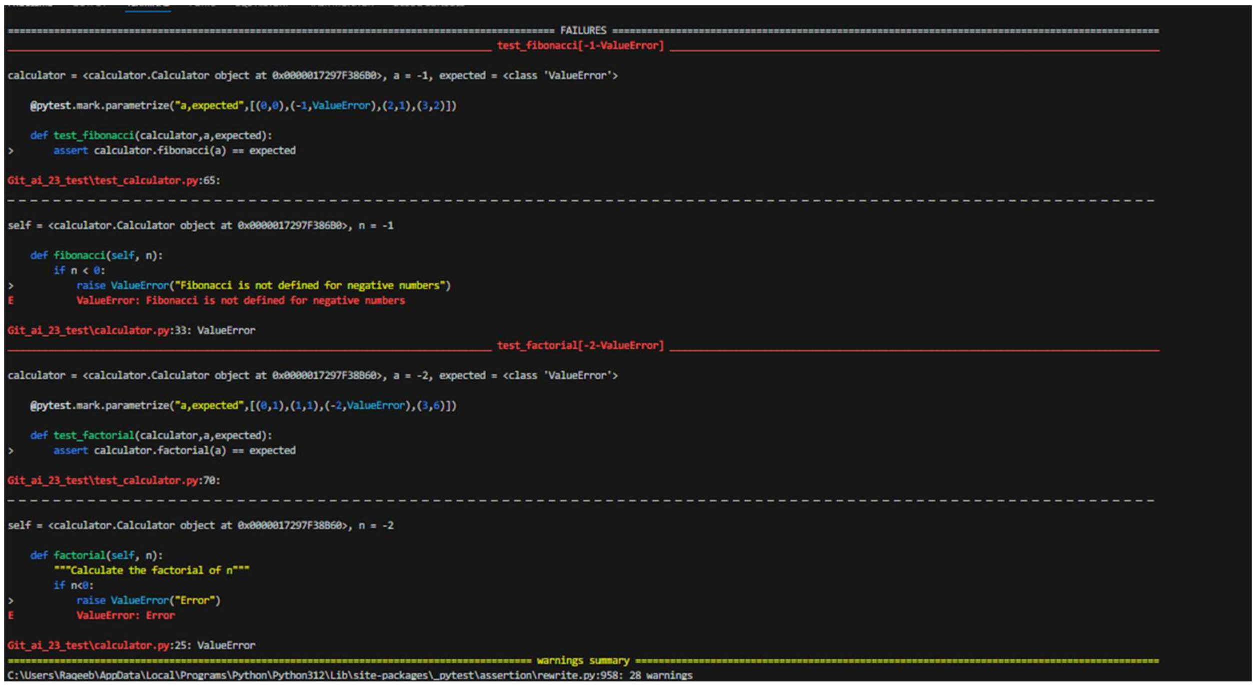
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
===================================== test session starts ======================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppDa
ta\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collecting ... This is Bug Hunter Practically!
Testing will help me find bugs that users might encounter
collected 18 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED                  [  5%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED                 [ 11%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED               [ 16%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED                  [ 22%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED            [ 27%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED          [ 33%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] PASSED  [ 38%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED            [ 44%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED           [ 50%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED           [ 55%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED               [ 61%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED               [ 66%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED            [ 72%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2-2-4] PASSED                [ 77%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[3-2-9] PASSED                [ 83%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[4-2-16] PASSED               [ 88%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2--2-0.25] PASSED            [ 94%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[10--1-0.1] PASSED            [100%]

===================================== 18 passed in 0.24s =======================================
PS D:\4th Semester\FSC\Assignment#2>
```



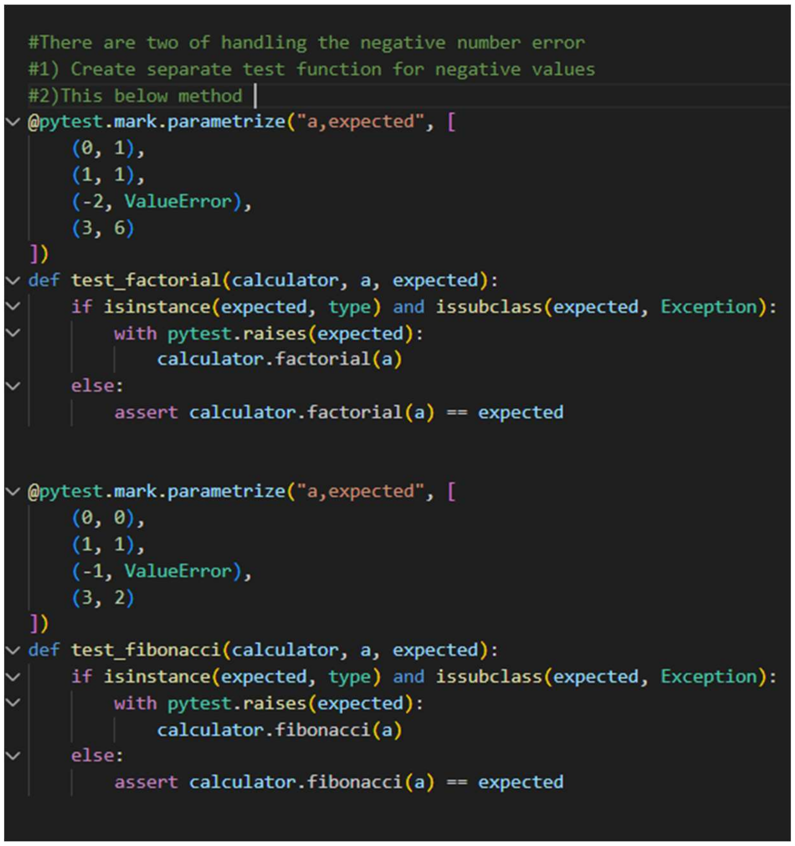
```
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collecting ... This is Bug Hunter Practically!
Testing will help me find bugs that users might encounter
collected 23 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED               [  4%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED              [  8%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED            [ 13%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED               [ 17%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED         [ 21%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED       [ 26%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] PASSED [ 30%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED         [ 34%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED        [ 39%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED        [ 43%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED            [ 47%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED            [ 52%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED         [ 56%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2-2-4] PASSED             [ 60%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[3-2-9] PASSED             [ 65%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[4-2-16] PASSED            [ 69%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2--2-0.25] PASSED         [ 73%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[10--1-0.1] PASSED         [ 78%]
Git_ai_23_test/test_calculator.py::test_fibonacci[0-0] PASSED                         [ 82%]
Git_ai_23_test/test_calculator.py::test_fibonacci[1-1] PASSED                         [ 86%]
Git_ai_23_test/test_calculator.py::test_fibonacci[2-1] PASSED                         [ 91%]
Git_ai_23_test/test_calculator.py::test_fibonacci[3-2] PASSED                         [ 95%]
Git_ai_23_test/test_calculator.py::test_factorial ERROR                               [100%]
```



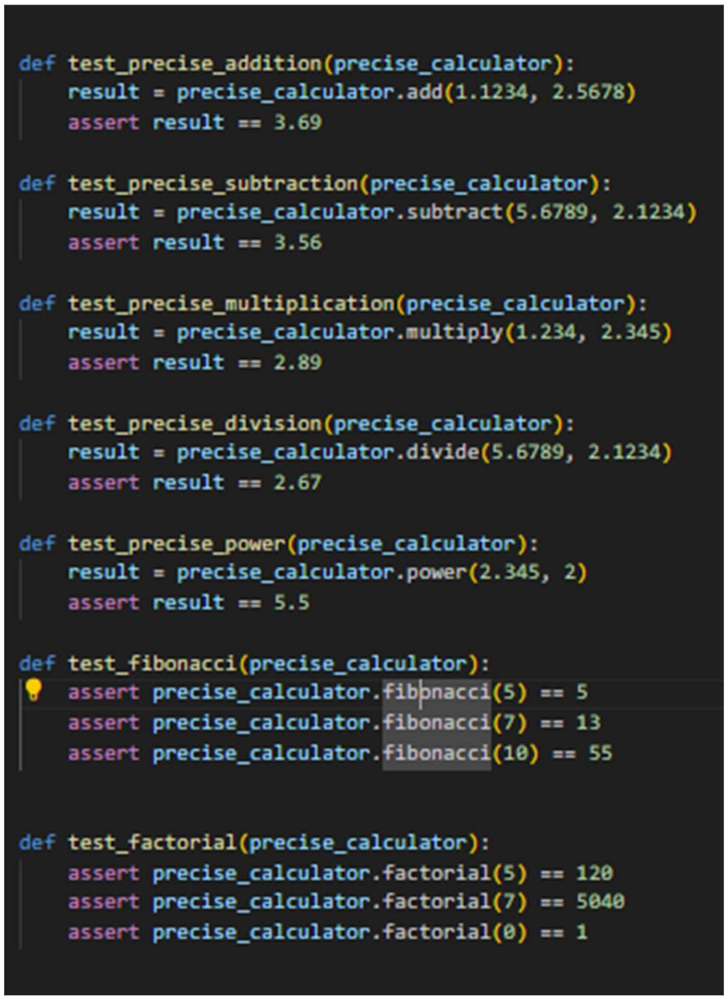
```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
=============================================================== test session starts ===============================================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collecting ... This is Bug Hunter Practically!
Testing will help me find bugs that users might encounter
collected 26 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED                                      [  3%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED                                     [  7%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED                                   [ 11%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED                                      [ 15%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED                                [ 19%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED                              [ 23%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] PASSED                      [ 26%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED                                [ 30%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED                               [ 34%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED                               [ 38%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED                                   [ 42%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED                                   [ 46%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED                                [ 50%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2-2-4] PASSED                                    [ 53%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[3-2-9] PASSED                                    [ 57%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[4-2-16] PASSED                                   [ 61%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2--2-0.25] PASSED                                [ 65%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[10--1-0.1] PASSED                                [ 69%]
Git_ai_23_test/test_calculator.py::test_fibonacci[0-0] PASSED                                                [ 73%]
Git_ai_23_test/test_calculator.py::test_fibonacci[1-1] PASSED                                                [ 76%]
Git_ai_23_test/test_calculator.py::test_fibonacci[2-1] PASSED                                                [ 80%]
Git_ai_23_test/test_calculator.py::test_fibonacci[3-2] PASSED                                                [ 84%]
Git_ai_23_test/test_calculator.py::test_factorial[0-1] PASSED                                                [ 88%]
Git_ai_23_test/test_calculator.py::test_factorial[1-1] PASSED                                                [ 92%]
Git_ai_23_test/test_calculator.py::test_factorial[2-2] PASSED                                                [ 96%]
Git_ai_23_test/test_calculator.py::test_factorial[3-6] PASSED                                                [100%]
```



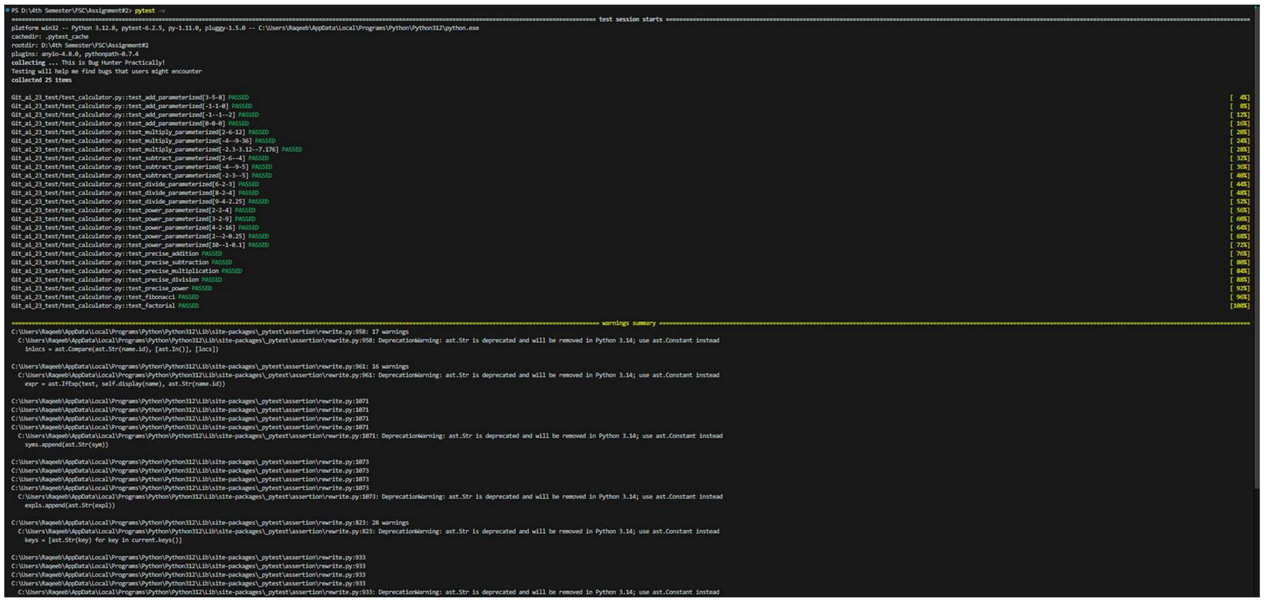
```
=============================================== FAILURES ===============================================
____________________________________ test_fibonacci[-1-ValueError] ____________________________________

calculator = <calculator.Calculator object at 0x0000017297F386B0>, a = -1, expected = <class 'ValueError'>

    @pytest.mark.parametrize("a,expected",[(0,0),(-1,ValueError),(2,1),(3,2)])

    def test_fibonacci(calculator,a,expected):
>       assert calculator.fibonacci(a) == expected

Git_ai_23_test\test_calculator.py:65:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

self = <calculator.Calculator object at 0x0000017297F386B0>, n = -1

    def fibonacci(self, n):
        if n < 0:
>           raise ValueError("Fibonacci is not defined for negative numbers")
E           ValueError: Fibonacci is not defined for negative numbers

Git_ai_23_test\calculator.py:33: ValueError
____________________________________ test_factorial[-2-ValueError] ____________________________________

calculator = <calculator.Calculator object at 0x0000017297F38B60>, a = -2, expected = <class 'ValueError'>

    @pytest.mark.parametrize("a,expected",[(0,1),(1,1),(-2,ValueError),(3,6)])

    def test_factorial(calculator,a,expected):
>       assert calculator.factorial(a) == expected

Git_ai_23_test\test_calculator.py:70:
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

self = <calculator.Calculator object at 0x0000017297F38B60>, n = -2

    def factorial(self, n):
        """Calculate the factorial of n"""
        if n<0:
>           raise ValueError("Error")
E           ValueError: Error

Git_ai_23_test\calculator.py:25: ValueError
=========================================== warnings summary ===========================================
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: 28 warnings
```



```python
#There are two of handling the negative number error
#1) Create separate test function for negative values
#2)This below method
@pytest.mark.parametrize("a,expected", [
    (0, 1),
    (1, 1),
    (-2, ValueError),
    (3, 6)
])
def test_factorial(calculator, a, expected):
    if isinstance(expected, type) and issubclass(expected, Exception):
        with pytest.raises(expected):
            calculator.factorial(a)
    else:
        assert calculator.factorial(a) == expected


@pytest.mark.parametrize("a,expected", [
    (0, 0),
    (1, 1),
    (-1, ValueError),
    (3, 2)
])
def test_fibonacci(calculator, a, expected):
    if isinstance(expected, type) and issubclass(expected, Exception):
        with pytest.raises(expected):
            calculator.fibonacci(a)
    else:
        assert calculator.fibonacci(a) == expected
```



```python
def test_precise_addition(precise_calculator):
    result = precise_calculator.add(1.1234, 2.5678)
    assert result == 3.69

def test_precise_subtraction(precise_calculator):
    result = precise_calculator.subtract(5.6789, 2.1234)
    assert result == 3.56

def test_precise_multiplication(precise_calculator):
    result = precise_calculator.multiply(1.234, 2.345)
    assert result == 2.89

def test_precise_division(precise_calculator):
    result = precise_calculator.divide(5.6789, 2.1234)
    assert result == 2.67

def test_precise_power(precise_calculator):
    result = precise_calculator.power(2.345, 2)
    assert result == 5.5

def test_fibonacci(precise_calculator):
    assert precise_calculator.fibonacci(5) == 5
    assert precise_calculator.fibonacci(7) == 13
    assert precise_calculator.fibonacci(10) == 55


def test_factorial(precise_calculator):
    assert precise_calculator.factorial(5) == 120
    assert precise_calculator.factorial(7) == 5040
    assert precise_calculator.factorial(0) == 1
```



```
PS D:\4th Semester\FSC\Assignment#2> pytest -v
======================================================== test session starts ========================================================
platform win32 -- Python 3.12.8, pytest-6.2.5, py-1.11.0, pluggy-1.5.0 -- C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\python.exe
cachedir: .pytest_cache
rootdir: D:\4th Semester\FSC\Assignment#2
plugins: anyio-4.8.0, pythonpath-0.7.4
collecting ... This is Bug Hunter Practically!
Testing will help me find bugs that users might encounter
collected 25 items

Git_ai_23_test/test_calculator.py::test_add_parameterized[3-5-8] PASSED                                  [  4%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1-1-0] PASSED                                 [  8%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[-1--1--2] PASSED                               [ 12%]
Git_ai_23_test/test_calculator.py::test_add_parameterized[0-0-0] PASSED                                  [ 16%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[2-6-12] PASSED                            [ 20%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-4--9-36] PASSED                          [ 24%]
Git_ai_23_test/test_calculator.py::test_multiply_parameterized[-2.3-3.12--7.176] PASSED                  [ 28%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[2-6--4] PASSED                            [ 32%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-4--9-5] PASSED                           [ 36%]
Git_ai_23_test/test_calculator.py::test_subtract_parameterized[-2-3--5] PASSED                           [ 40%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[6-2-3] PASSED                               [ 44%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[8-2-4] PASSED                               [ 48%]
Git_ai_23_test/test_calculator.py::test_divide_parameterized[9-4-2.25] PASSED                            [ 52%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2-2-4] PASSED                                [ 56%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[3-2-9] PASSED                                [ 60%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[4-2-16] PASSED                               [ 64%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[2--2-0.25] PASSED                            [ 68%]
Git_ai_23_test/test_calculator.py::test_power_parameterized[10--1-0.1] PASSED                            [ 72%]
Git_ai_23_test/test_calculator.py::test_precise_addition PASSED                                          [ 76%]
Git_ai_23_test/test_calculator.py::test_precise_subtraction PASSED                                       [ 80%]
Git_ai_23_test/test_calculator.py::test_precise_multiplication PASSED                                    [ 84%]
Git_ai_23_test/test_calculator.py::test_precise_division PASSED                                          [ 88%]
Git_ai_23_test/test_calculator.py::test_precise_power PASSED                                             [ 92%]
Git_ai_23_test/test_calculator.py::test_fibonacci PASSED                                                 [ 96%]
Git_ai_23_test/test_calculator.py::test_factorial PASSED                                                 [100%]

========================================================= warnings summary =========================================================
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: 17 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:958: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    inlocs = ast.Compare(ast.Str(name.id), [ast.In()], [locs])

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961: 16 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:961: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    expr = ast.IfExp(test, self.display(name), ast.Str(name.id))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1071: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    syms.append(ast.Str(sym))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:1073: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    expls.append(ast.Str(expl))

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:823: 28 warnings
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:823: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
    keys = [ast.Str(key) for key in current.keys()]

C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933
  C:\Users\Raqeeb\AppData\Local\Programs\Python\Python312\Lib\site-packages\_pytest\assertion\rewrite.py:933: DeprecationWarning: ast.Str is deprecated and will be removed in Python 3.14; use ast.Constant instead
```

What are Parameterized tests?

Parameterized tests are a type of test that enables you to evaluate the same logic for various scenarios. Rather than creating multiple tests for the same functionality, you can write a single test that can accommodate different inputs and cover various cases.

When to use parameterized tests:

- The tested logic is simple.
- You are testing the same function but for different scenarios (i.e., different input values).
- The assertion logic is the same for every test.
- When the tests are simple enough to generalize a test function

Benefits of parameterized tests:

1. Improved Test Coverage
2. Reduced Code Duplication
3. Ease of Maintenance
4. Better Readability and Organization

When NOT to use parameterized tests:

- When you are testing different functionalities.
- When the parameterized test has too many arguments.
- When the logic being tested is too complex.
- When tests require unique initializations for each scenario.

Text-Reference

Pytest Fixtures:(Reference)

Pytest fixtures are a powerful feature that allows you to set up and tear down resources needed for your tests.

Scope:(Reference)

The scope of a fixture defines the level on which it's being invoked and destroyed.

- **function**: the default scope, the fixture is destroyed at the end of the test.
- **class**: the fixture is destroyed during teardown of the last test in the class.
- **module**: the fixture is destroyed during teardown of the last test in the module.
- **package**: the fixture is destroyed during teardown of the last test in the package.
- **session**: the fixture is destroyed at the end of the test session.

Application:

Docker:

If we are building a service that interacts with an SFTP server and/or a database, for example. We can run a docker container in the background and simulate the interaction. This allows us to have the proper testing structure of defining the state, action, and asserting the result.

---

3. What bugs you found in the calculator implementation and how your tests helped discover them?

Following are the bugs faced by me during the testing:

1)The addition of some floating value in multiplication. When I tested the expected value was not same is the calculated value because of the addition.

2)Not getting right floating point values in multiplication. Then using the .approx function we get the expected value.

3)Negative value in the power caused us error. As testing shows as the expected value is not matching the derived value which help us to change the logic.

4)For negative number the factorial is not possible so it should show ValueError. Test show us that we cannot compare directly using **assert** for that we have to using **Isinstance and issubclass** functions.

---

4)How using parameterized tests and fixtures improved your testing process?

Parameterized test enables us to test the same code with different test cases so we don’t have to repeat the code.

Eliminated Boilerplate: Instead of instantiating Calculator() in every test, a fixture provided a reusable instance:

Scalability: Fixtures can be shared across test files (e.g., via conftest.py).

Custom Configurations: Advanced use cases e.g., PreciseCalculator were possible without cluttering tests.