

# 1.3.-

```
def test_suma_params(input_n1, input_n2, expected):
    assert suma(input_n1, input_n2) == expected

def test_mayor():
    assert mayor(9,5) ==9
    assert mayor(22,51) ==51
    assert mayor(10,10) ==0
    assert mayor(-1,5) ==5

@pytest.mark.parametrize(
    "input_n1, input_n2, expected",
    [
        (9, 5, 9),
        (22, 51, 51),
        (10, 10, 0),
        (-1, 5, 5)
    ]
)
```

(venv) PS C:\Users\usuariot\Documents\ProgPython> pytest -v

===== test session starts =====

platform win32 -- Python 3.12.0, pytest-7.4.2, pluggy-1.3.0 -- C:\Users\usuariot\Documents\ProgPython\venv\Scripts\python.exe

cachedir: .pytest\_cache

rootdir: C:\Users\usuariot\Documents\ProgPython

collected 12 items

ejerciciosU1/test_prueba1.py::test_suma	PASSED	[ 8%]
ejerciciosU1/test_prueba1.py::test_suma_params[1-1-2]	PASSED	[ 16%]
ejerciciosU1/test_prueba1.py::test_suma_params[0-0-0]	PASSED	[ 25%]
ejerciciosU1/test_prueba1.py::test_suma_params[100--100-0]	PASSED	[ 33%]
ejerciciosU1/test_prueba1.py::test_suma_params[-15--1--16]	PASSED	[ 41%]
ejerciciosU1/test_prueba1.py::test_suma_params[-3-8-5]	PASSED	[ 50%]
ejerciciosU1/test_prueba1.py::test_suma_params[9--3-6]	PASSED	[ 58%]
ejerciciosU1/test_prueba1.py::test_mayor	PASSED	[ 66%]
ejerciciosU1/test_prueba1.py::test_mayor_params[9-5-9]	PASSED	[ 75%]
ejerciciosU1/test_prueba1.py::test_mayor_params[22-51-51]	PASSED	[ 83%]
ejerciciosU1/test_prueba1.py::test_mayor_params[10-10-0]	PASSED	[ 91%]
ejerciciosU1/test_prueba1.py::test_mayor_params[-1-5-5]	PASSED	[100%]

===== 12 passed in 0.05s =====

o (venv) PS C:\Users\usuariot\Documents\ProgPython>

1.4.-

The screenshot shows a Visual Studio Code editor window with the following components:

- EXPLORER:** A list of files in the 'PROGPYTHON' workspace. The file 'test\_prueba1.py' is selected and highlighted in blue.
- EDITOR:** Displays the code for 'test\_prueba1.py'. The code includes a function 'test\_suma\_params' and a function 'test\_mayor' with its parametrized test 'test\_mayor\_params'. The code is as follows:

```
19 )
20 def test_suma_params(input_n1, input_n2, expected):
21     assert suma(input_n1, input_n2) == expected
22
23
24 def test_mayor():
25     assert mayor(9,5) ==9
26     assert mayor(22,51) ==51
27     assert mayor(10,10) ==0
```
- TERMINAL:** Shows the output of running the tests. It displays two failures:

```
===== FAILURES =====
test_mayor

def test_mayor():
    assert mayor(9,5) ==9
    assert mayor(22,51) ==51
> assert mayor(10,10) ==0
E assert 81 == 0
E + where 81 = mayor(10, 10)

ejerciciosU1\test_prueba1.py:27: AssertionError
test_mayor_params[10-10-0]

input_n1 = 10, input_n2 = 10, expected = 0

@pytest.mark.parametrize(
    "input_n1, input_n2, expected",
    [
        (9, 5, 9),
        (22, 51, 51),
        (10, 10, 0),
        (-1, 5, 5)
    ]
)
def test_mayor_params(input_n1, input_n2, expected):
> assert mayor(input_n1, input_n2) == expected
E assert 81 == 0
E + where 81 = mayor(10, 10)

ejerciciosU1\test_prueba1.py:40: AssertionError

===== short test summary info =====
FAILED ejerciciosU1\test_prueba1.py::test_mayor - assert 81 == 0
FAILED ejerciciosU1\test_prueba1.py::test_mayor_params[10-10-0] - assert 81 == 0
===== 2 failed, 10 passed in 0.13s =====

(venv) PS C:\Users\usuariot\Documents\ProgPython>
```

The screenshot shows a Visual Studio Code editor window with the following components:

- File Explorer (Left):** Shows a project structure with folders like 'ejerciciosU1' and files like 'test\_prueba1.py'.
- Editor (Center):** Displays the code for 'test\_prueba1.py'. The code includes a function 'test\_suma\_params' and a function 'test\_mayor' with several assertions. A pytest parametrization is also shown.
- TEST RESULTS (Bottom):** A panel showing the results of the test run. It lists the tests that passed, including 'test\_suma' and 'test\_mayor', along with their parametrized inputs.

```
def test_suma_params(input_n1, input_n2, expected):
    assert suma(input_n1, input_n2) == expected

def test_mayor():
    assert mayor(9,5) ==9
    assert mayor(22,51) ==51
    assert mayor(10,10) ==0
    assert mayor(-1,5) ==5

@pytest.mark.parametrize(
    "input_n1, input_n2, expected",
    [
        (9, 5, 9),
        (22, 51, 51),
        (10, 10, 0),
        (-1, 5, 5)
    ]
)
```

Test Results:

- Test run at 10/23/2023, 7:55:02 PM
  - test\_suma
  - [1-1-2]
  - [0-0-0]
  - [100--100-0]
  - [-15--1--16]
  - [-3-8-5]
  - [9--3-6]
  - test\_mayor
  - [9-5-9]
  - [22-51-51]
  - [10-10-0]
  - [-1-5-5]
- Test run at 10/23/2023, 6:18:56 PM

## 2.2.-

The image shows a VS Code editor window with the following components:

- File Explorer:** Shows a project structure with folders 'ejerciciosU1' and 'test\_prueba1.py'.
- Code Editor:** Contains the following Python code:

```
1 def suma(num1, num2):
2     return num1 + num2
3
4 print(suma(3, 2))
5 print("La suma de", 3, "+", 2, "es", suma(3, 2))
6 print("La suma de " + str(3) + " + " + str(2) + " es " + str(suma(3, 2)))
7
8 res = "La suma de "
9 res += str(3)
10 res += " + " + str(2)
11 res += " es " + str(suma(3, 2))
12 print(res)
13 print(f"La suma de {3} + {2} es {suma(3, 2)}")
14
15 def mayor(num1, num2):
16     if num1 == num2:
17         return 81
18     elif num1 > num2:
19         return num1
20     else:
21         return num2
```
- TEST RESULTS:** Shows a failed test run at 10/23/2023, 7:56:17 PM. The error message is: `AssertionError`. The test case is `[10-10-0]` with `input_n1 = 10, input_n2 = 10, expected = 0`. The test function is `test_mayor_params`.
- PROBLEMS:** Shows the error message: `AssertionError`.
- OUTPUT:** Shows the output of the code execution: `> assert mayor(10,10) ==0`, `E assert 81 == 0`, `+ where 81 = mayor(10, 10)`.
- DEBUG CONSOLE:** Shows the test function definition and the assertion error: `AssertionError`.
- PORTS:** Shows a list of test cases: `[1-1-2]`, `[0-0-0]`, `[100--100-0]`, `[-15--1--16]`, `[-3-8-5]`, `[9--3-6]`, `[9-5-9]`, `[22-51-51]`, `[-1-5-5]`.