Introduction to pytest

In the P06_pytest_project folder, create a folder called **calculator**
In the calculator folder, create an empty file called **__init.py__**
In the calculator folder, create a file called **calculator.py**

Paste the following code in **calculator.py**

```
class Calculator:
    def add(self, a, b):
        return a + b

    def subtract(self, a, b):
        return a - b

    def multiply(self, a, b):
        return a * b

    def divide(self, a, b):
        if b == 0:
            raise ZeroDivisionError("Division by zero error")
        return a / b
```

In the tests folder, create a file called **test_calculator.py**

Paste the following code in **test_calculator.py**

```
from calculator.calculator import Calculator

class TestCalculator:
    def test_add(self):
        # arrange
        a = 4321
        b = 1234
        cal = Calculator()

        # act
        result = cal.add(a, b)

        # assert
        expected = 5555
        assert result == expected
```

Run pytest and ensure that all tests pass

In the **TestCalculator** class in **test_calculator.py** :

- Create at least 1 test case for each method in the Calculator class
- Apply the "Arrange, Act, Assert" pattern to effectively structure your unit tests
- Use the assert method to compare the expected result with the actual result

Paste the methods from the Calculator class and corresponding pytest methods in the table below

function	Calculator code	pytest code

add	def add(self, a, b):
	  return a + b
		def test_add(self):
		  # arrange
		  a = 4321
		  b = 1234
		  cal = Calculator()
		
		  # act
		  result = cal.add(a, b)
		
		  # assert
		  expected = 5555
		  assert result == expected

subtract		
multiply		
divide		

Install and run coverage.py

To install coverage, run the following command

```
pip install coverage
```

To run coverage, run the following command

```
coverage run --branch -m pytest
```

To generate the report in HTML format, run the following 2 commands

```
coverage report -m
coverage html
```