Yahtzee Version 2 Testing: Lower Section

The functions that you will test here are:

- `num_of_a_kind()`
- `yahtzee()`

What to do

Create a new file called `test_yahtzee2.py` in the same folder as `yahtzee2.py`.

Start by importing the functions you want to test from `yahtzee2.py` in the file `test_yahtzee2.py`.

Write six unit tests, covering the following cases:
a) “3 of a kind” found
b) “3 of a kind” not found
c) “4 of a kind” found
d) “4 of a kind” not found
e) “Yahtzee” found
f) “Yahtzee” not found

Each test must have a hardcoded roll and a single assertion.

Use the following template. All unit tests defined in the template **must be present and implemented** in your code (you may **not** omit anything).

```
# TODO: Import the functions you will be testing

# Version 1 tests here

def test_three_of_a_kind_found():
    """
    Tests num_of_a_kind() with a roll that has "3 of a kind".
    """
    roll = # TODO (hardcode a roll that satisfies this test case)
    assert num_of_a_kind(roll, 3) == # TODO
```

```
def test_three_of_a_kind_not_found():
    """
    Tests num_of_a_kind() with a roll that doesn't have "3 of a kind".
    """
    roll = # TODO (hardcode a roll that satisfies this test case)
    assert num_of_a_kind(roll, 3) == # TODO

def test_four_of_a_kind_found():
    """
    Tests num_of_a_kind() with a roll that has "4 of a kind".
    """
    roll = # TODO (hardcode a roll that satisfies this test case)
    assert num_of_a_kind(roll, 4) == # TODO

def test_four_of_a_kind_not_found():
    """
    Tests num_of_a_kind() with a roll that doesn't have "4 of a kind".
    """
    roll = # TODO (hardcode a roll that satisfies this test case)
    assert num_of_a_kind(roll, 4) == # TODO

def test_yahtzee_found():
    """
    Tests yahtzee() with a roll that has a yahtzee.
    """
    roll = # TODO (hardcode a roll that satisfies this test case)
    assert yahtzee(roll) == # TODO

def test_yahtzee_not_found():
    """
    Tests yahtzee() with a roll that has no yahtzee.
    """
    roll = # TODO (hardcode a roll that satisfies this test case)
    assert yahtzee(roll) == # TODO
```

Run your tests by calling `python -m pytest test_yahtzee2.py` in the terminal.

All tests should pass.

How to debug

Your unit tests can be failing for two reasons:

1. Your code has errors

2. Your unit tests have errors

Make sure that you understand what is going wrong and ensure that the tests are correct before changing the code.

Program name

Save your unit tests as `test_yahtzee2.py`.

Demo

https://asciinema.org/a/URtAdxFXO7NgAftq0lvH5yFNo

Submitting

Submit `test_yahtzee2.py` via eClass.