

Analiza i Bazy Danych			
Imię	Sebastian	Grupa	1
Nazwisko	Pilch	Numer Laboratoriów	13

Faza Red:

Napisanie testu dla funkcji sortującej:

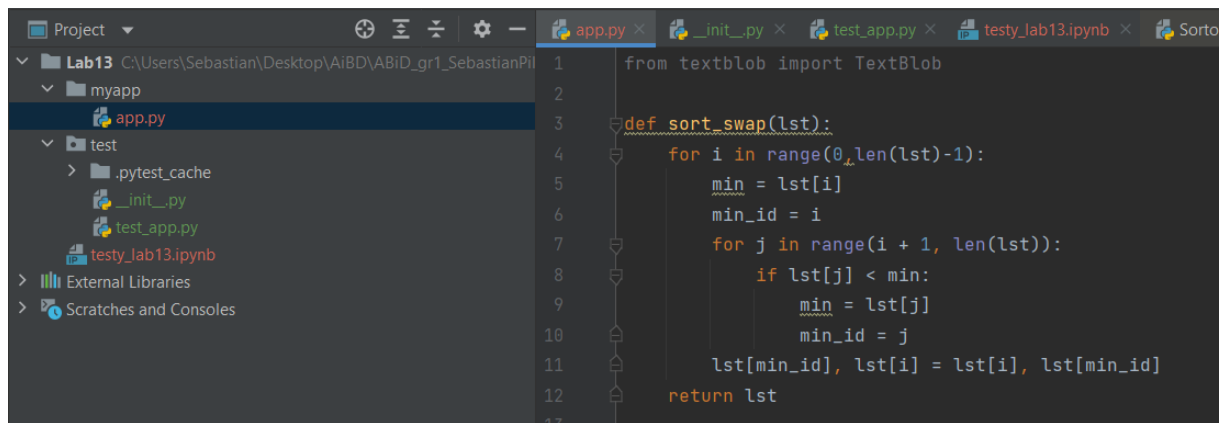
The screenshot shows a Python IDE with a project named 'Lab13'. The file explorer on the left shows a directory structure with 'myapp' and 'test' folders. The 'test' folder contains 'test\_app.py'. The main editor shows the following code in 'test\_app.py':

```
1 from myapp import app
2
3 def test_sort_swap():
4     got = app.sort_swap([2, 6, 12, 9, 3, 7, 11, 4, 5, 10, 8, 1])
5     want = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
6     assert got == want
7
8
9
10
```

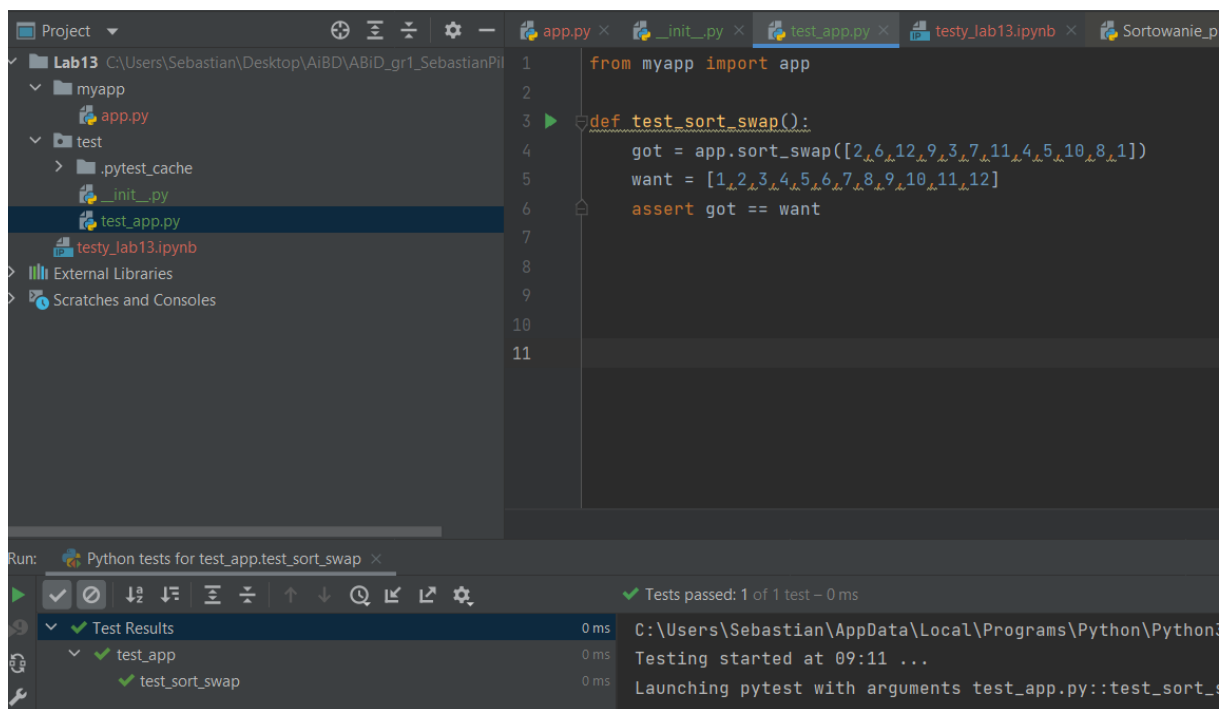
The bottom panel shows the 'Run' output for 'Python tests for test\_app.test\_sort\_swap'. It indicates that 1 test failed. The error message is: 'ERROR: not found: C:\Users\Sebastian\Desktop\...'.

Faza Green:

Napisanie funkcji i przetestowanie jej na prostym przykładzie:



```
1 from textblob import TextBlob
2
3 def sort_swap(lst):
4     for i in range(0, len(lst)-1):
5         min = lst[i]
6         min_id = i
7         for j in range(i + 1, len(lst)):
8             if lst[j] < min:
9                 min = lst[j]
10                min_id = j
11            lst[min_id], lst[i] = lst[i], lst[min_id]
12        return lst
```



```
1 from myapp import app
2
3 def test_sort_swap():
4     got = app.sort_swap([2, 6, 12, 9, 3, 7, 11, 4, 5, 10, 8, 1])
5     want = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
6     assert got == want
```

Run: Python tests for test\_app.test\_sort\_swap

Test Results	Duration	Details
Test Results	0 ms	C:\Users\Sebastian\AppData\Local\Programs\Python\Python3...
test_app	0 ms	Testing started at 09:11 ...
test_sort_swap	0 ms	Launching pytest with arguments test_app.py::test_sort_s...

Jak widac testy nie zgłaszaja błędów

Faza refactor:

Poprawa kodu funkcji, ulepszenie testów. Postanowiłem dodać sprawdzanie typów, tak aby argument był listą int.

```

def sort_swap(lst):
    if all(isinstance(x, int) for x in lst):
        for i in range(0, len(lst)-1):
            min = lst[i]
            min_id = i
            for j in range(i + 1, len(lst)):
                if lst[j] < min:
                    min = lst[j]
                    min_id = j
            lst[min_id], lst[i] = lst[i], lst[min_id]
        return lst
    return None

```

```

from myapp import app
import pytest

testdata = ([4,5,10,8,'a','b',1], [1,4,None,3,2],[2.1,3.1,1,5,4],'int_lst')

@pytest.mark.parametrize('sample', testdata)

def test_sort_swap(sample):
    if sample == 'int_list':
        got = app.sort_swap([2,6,12,9,3,7,11,4,5,10,8,1])
        want = [1,2,3,4,5,6,7,8,9,10,11,12]
        assert got == want
    assert app.sort_swap(sample) is None

```

```

✓ Test Results
  ✓ test_app
    ✓ test_sort_swap
      ✓ (sample0)
      ✓ (sample1)
      ✓ (sample2)
      ✓ (int_lst)

```