

Issue 1

```
>>> str(encode('sos')) #doctest: -FAIL_FAST
'... --- ...'
>>> str(encode("SOS")) #doctest: -FAIL_FAST
'... --- ...'
>>> str(encode("SOS"))
'... --- ...'
>>> str(encode("SOS"))
'... --- ...'
>>> str(encode(123))
Traceback (most recent call last):
...
TypeError: 'int' object is not iterable
```

- `python -m doctest -v hw4_1.py`
(если мы запускаем тест без флагов то все 5 тестов проходят, ошибка только в первом)

```
1 items had failures:
  1 of  5 in hw4_1.encode
5 tests in 2 items.
4 passed and 1 failed.
***Test Failed*** 1 failures.
```

- `python -m doctest -v -o FAIL_FAST hw4_1.py`
(если мы запускаем тест с флагом `FAIL_FAST`, то получив ошибку в первом тесте он должен остановиться, но из-за директивы `-FAIL_FAST` он идет дальше. Но он и в будущем не перейдет к следующему тесту, если у данного нет директивы `-FAIL_FAST`, даже если тест пройден)

```
1 items had failures:
  1 of  3 in hw4_1.encode
3 tests in 2 items.
2 passed and 1 failed.
***Test Failed*** 1 failures.
```

Issue 2

```
testdata = [
    ('SOS', '... --- ...'),
    ('GE', '--- .'),
    ('QRZ?', '---. - .- .-.. ..-..')
]

@pytest.mark.parametrize('result, source', testdata)
def test_encode(source, result):
    assert decode(source) == result
```

- python -m pytest hw4_2.py

```
(avito_aaa_python) C:\Users\basic\PycharmProjects\avito_aaa_python>python -m pytest hw4_2.py
===== test session starts =====
platform win32 -- Python 3.8.5, pytest-7.1.3, pluggy-1.0.0
rootdir: C:\Users\basic\PycharmProjects\avito_aaa_python
collected 3 items

hw4_2.py ...

===== 3 passed in 0.02s =====
```

Issue 3

```
class TestFitTransform(unittest.TestCase):
    def test_cities(self):
        cities = ['Moscow', 'New York', 'Moscow', 'London']
        exp_transformed_cities = [
            ('Moscow', [0, 0, 1]),
            ('New York', [0, 1, 0]),
            ('Moscow', [0, 0, 1]),
            ('London', [1, 0, 0]),
        ]
        transformed_cities = fit_transform(cities)
        self.assertEqual(transformed_cities, exp_transformed_cities)

    def test_cities_unique(self):
        cities = ['Moscow', 'New York', 'Moscow', 'London']
        cities_unique = list(set(cities))
        transformed_cities = fit_transform(cities)
        for city in cities_unique:
            self.assertIn(city, list(map(list, zip(*transformed_cities))))[0])

    def test_colors(self):
        cities = ['green', 'green', 'red']
        exp_transformed_colors = [
            ('green', [0, 1]),
            ('green', [0, 1]),
            ('red', [1, 0]),
        ]
        transformed_cities = fit_transform(cities)
        self.assertEqual(transformed_cities, exp_transformed_colors)

    def test_empty(self):
        with self.assertRaises(TypeError):
            fit_transform()
```

- `python -m unittest hw4_3.py`

```
(avito_aaa_python) C:\Users\basic\PycharmProjects\avito_aaa_python>python -m unittest hw4_3.py
....
-----
Ran 4 tests in 0.001s

OK
```

Issue 4

(тесты повторяют тесты из issue3)

- `python -m pytest hw4_4.py`

```
(avito_aaa_python) C:\Users\basic\PycharmProjects\avito_aaa_python>python -m pytest hw4_4.py
===== test session starts =====
platform win32 -- Python 3.8.5, pytest-7.1.3, pluggy-1.0.0
rootdir: C:\Users\basic\PycharmProjects\avito_aaa_python
collected 4 items

hw4_4.py ....

===== 4 passed in 0.06s =====
```

Issue 5

```
date0 = {'currentDateTime': '01.03.2019T15:27Z'}

def test_get_cases0():
    with patch("json.load") as mocked_what_is_year_now:
        mocked_what_is_year_now.return_value = date0
        assert what_is_year_now() == 2019

date1 = {'currentDateTime': '2019-03-01T15:27Z'}

def test_get_cases1():
    with patch("json.load") as mocked_what_is_year_now:
        mocked_what_is_year_now.return_value = date1
        assert what_is_year_now() == 2019

date2 = {'currentDateTime': '2019/03/01T15:27Z'}

def test_get_cases2():
    with patch("json.load") as mocked_what_is_year_now:
        mocked_what_is_year_now.return_value = date2
        with pytest.raises(ValueError):
            what_is_year_now()
```

- `python -m pytest -q hw5_5.py --cov . --cov-report html`

```
basic@LAPTOP-00GLEIQ3 C:\Users\basic\PycharmProjects\avito_aaa_python
$ python -m pytest -q hw5_5.py --cov . --cov-report html
...

----- coverage: platform win32, python 3.8.5-final-0 -----
Coverage HTML written to dir htmlcov

3 passed in 1.18s
```