Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Курс «Базовые компоненты интернет-технологий»

Отчёт по домашнему заданию

Выполнил:
студент группы ИУ5-33Б Кузнецов В. А.
подпись: , дата:
Проверил:
лектор Гапанюк Ю. Е.
подпись: , дата:

2022 г.

Задание:

1. С использованием механизма итераторов или генераторов реализуйте с помощью концепции ленивых вычислений одну из последовательностей OEIS. Примером могут являться числа Фибоначчи.
2. Для реализованной последовательности разработайте 3-5 модульных тестов, которые, в том числе, проверяют то, что последовательность поддерживает ленивые вычисления.
3. Разработайте веб-сервис с использованием фреймворка Flask, который возвращает N элементов последовательности (параметр N передается в запросе к сервису).
4. Создайте Jupyter-notebook, который реализует обращение к веб-сервису с использованием библиотеки requests и визуализацию полученных от веб-сервиса данных с использованием библиотеки matplotlib.

Текст программы

lucky_numbers.py

```
from itertools import count, islice


def lucky_numbers():
    """
    Генератор счастливых чисел\n
    https://oeis.org/A000959
    """
    yield 1
    sequence = []
    for i in count(3, 2):
        for divider in sequence:
            divider[1] += 1
            if divider[1] % divider[0] == 0:
                break
        else:
            yield i
            sequence.append([i, len(sequence) + 2])
```

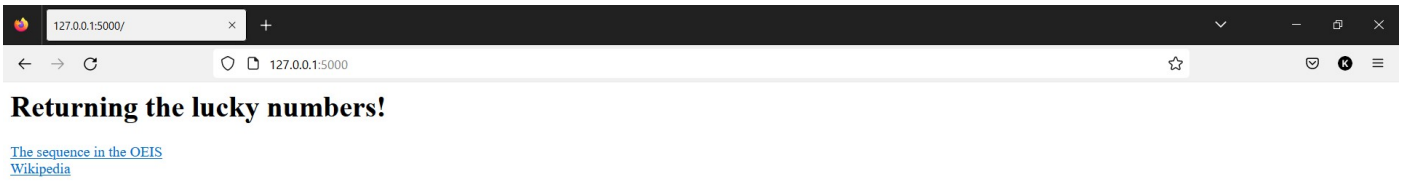
tests.py

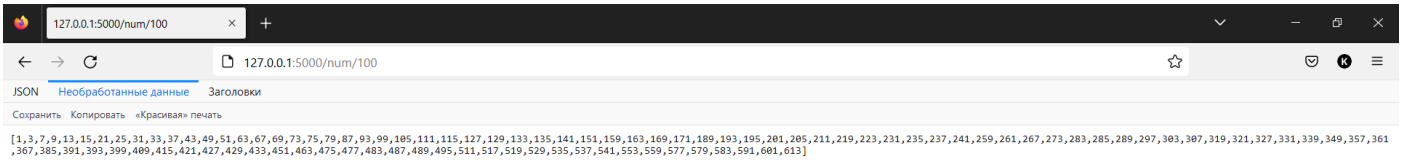
```
import unittest
from types import GeneratorType
from itertools import islice
from lucky_numbers import lucky_numbers


class TestLuckyNumbers(unittest.TestCase):
    def test_is_generator(self):
        self.assertIsInstance(lucky_numbers(), GeneratorType)

    def test_values(self):
        result = [1, 3, 7, 9, 13, 15, 21, 25, 31, 33,
                  37, 43, 49, 51, 63, 67, 69, 73, 75, 79,
                  87, 93, 99, 105, 111, 115, 127, 129, 133,
                  135, 141, 151, 159, 163, 169, 171, 189, 193,
                  195, 201, 205, 211, 219, 223, 231, 235, 237,
                  241, 259, 261, 267, 273, 283, 285, 289, 297,
                  303, 307, 319, 321, 327, 331, 339, 349, 357,
                  361, 367, 385, 391, 393, 399, 409, 415, 421,
                  427, 429, 433, 451, 463, 475, 477, 483, 487,
                  489, 495, 511, 517, 519, 529, 535, 537, 541,
                  553, 559, 577, 579, 583, 591, 601, 613, 615,
```

```
                    619, 621, 631, 639, 643, 645, 651, 655, 673,
                    679, 685, 693, 699, 717, 723, 727, 729, 735,
                    739, 741, 745, 769, 777]
        answer = list(islice(lucky_numbers(), len(result)))
        self.assertEqual(result, answer)

    def test_first_ten(self):
        result = [1, 3, 7, 9, 13, 15, 21, 25, 31, 33]
        numbers = lucky_numbers()
        for i in range(10):
            self.assertEqual(result[i], numbers.__next__())


if __name__ == '__main__':
    unittest.main()
```

```
from flask import Flask
from lucky_numbers import lucky_numbers


app = Flask(__name__)


@app.route("/")
def hello_world():
    html = """<h1>Returning the lucky numbers!</h1>
              <a href="https://oeis.org/A000959">
                 The sequence in the OEIS<br>
              </a>
              <a href="https://ru.wikipedia.org/wiki/Счастливое_число_(lucky_number)">
                 Wikipedia
              </a>"""
    return html


@app.route('/num/<int:cnt>')
def get_fib(cnt):
    numbers = lucky_numbers()
    res = [next(numbers) for _ in range(cnt)]
    return res
```

Пример выполнения





Подключение библиотеки для запросов

[10]:
```
!pip install requests
import requests
```

```
Requirement already satisfied: requests in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (2.28.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from requests) (2022.6.15)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from requests) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from requests) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from requests) (1.26.11)
```

Функции для выполнения запроса к сервису

[5]:
```
def make_url(cnt):
    base_url = 'http://127.0.0.1:5000/num/'
    res = base_url + str(cnt)
    return res

def get_data(cnt):
    url = make_url(cnt)
    r = requests.get(url)
    return r.json()
```

[6]:
```
cnt_list = [5, 10, 15, 20]
for cnt in cnt_list:
    print('{} первых счастливых чисел: {}'.format(cnt, get_data(cnt)))
```

```
5 первых счастливых чисел: [1, 3, 7, 9, 13]
10 первых счастливых чисел: [1, 3, 7, 9, 13, 15, 21, 25, 31, 33]
15 первых счастливых чисел: [1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63]
20 первых счастливых чисел: [1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63, 67, 69, 73, 75, 79]
```

Подключение библиотеки визуализации данных

[11]:
```
!pip install matplotlib
import matplotlib.pyplot as plt
```

```
Requirement already satisfied: matplotlib in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (3.5.3)
Requirement already satisfied: packaging>=20.0 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (9.2.0)
Requirement already satisfied: numpy>=1.17 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (1.23.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (4.37.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: six>=1.5 in c:\users\kuzva\appdata\roaming\jupyterlab-desktop\jlab_server\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

[8]:
```
y_10 = get_data(10)
x_10 = list(range(1, len(y_10)+1))
```

[9]:
```
fig = plt.figure(figsize = (10, 5))
plt.plot(x_10, y_10)
plt.title('Первые {} счастливых чисел'.format(len(y_10)))
plt.show()
```