



**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, искусственный интеллект и системы  
управления» Кафедра «Системы обработки информации и управления»**

**Домашнее задание по курсу  
«Базовые компоненты интернет-технологий»**

**Выполнил:**  
**студент группы ИУ5-33Б**  
**Попов Степан**

**Проверил:**  
**Доцент кафедры ИУ5**  
**Гапанюк Юрий Евгеньевич**

**2022 г.**

## Постановка задачи

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 вебсервису с использованием библиотеки requests и визуализацию полученных от веб-сервиса данных с использованием библиотеки matplotlib.

## Текст программы

### **binomial\_coefs.py (реализация генератора)**

```
def binomial_coefs():
    prev = []
    cur = [1]
    while True:
        yield cur
        prev = cur

        cur = []
        cur.append(1)
        for i in range(len(prev) - 1):
            cur.append(prev[i] + prev[i + 1])
        cur.append(1)

if __name__ == '__main__':
    bin_gen = binomial_coefs()
    res = [next(bin_gen) for _ in range(4)]
    for i in res:
        print(i)
```

### **web\_binomial.py (реализация Flask-сервиса)**

```
from flask import Flask
from flask import render_template
from binomial_coefs import binomial_coefs

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Returning Pascals Rectangle!</p>"

@app.route('/num/<int:cnt>')
def get_coef(cnt):
    bin_gen = binomial_coefs()
    result = [next(bin_gen) for _ in range(cnt + 1)]
    return result

@app.route('/num/<int:cnt>/rectangle')
def get_coef_render(cnt):
    bin_gen = binomial_coefs()
```

```
result = [next(bin_gen) for _ in range(cnt + 1)]
return render_template("num.html", result = result)
```

## templates/num.html

```
<html>
<body>
  <h1>Pascal rectangle!</h1>
  {% for coefs in result %}
  <p>{% for coef in coefs %} <b> {{coef}} </b> {% endfor %}</p>
  {% endfor %}
</body>
</html>
```

## Jupyter-notebook

```
In [33]: import requests
URL = 'http://127.0.0.1:5000'
def make_url(cnt):
    res = URL + '/num/' + str(cnt)
    return res
def get_bin_coefs(cnt):
    url = make_url(cnt)
    r = requests.get(url)
    return r.json()
```

```
In [34]: # Биномиальные коэффициенты от n = 0 до 5
bin_coefs_from_1_to_5 = get_bin_coefs(5)
bin_coefs_from_1_to_5
```

```
Out[34]: [[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1], [1, 5, 10, 10, 5, 1]]
```

```
In [35]: # Биномиальные коэффициенты от n = 0 до 10
bin_coefs_from_1_to_10 = get_bin_coefs(10)
bin_coefs_from_1_to_10
```

```
Out[35]: [[1],
[1, 1],
[1, 2, 1],
[1, 3, 3, 1],
[1, 4, 6, 4, 1],
[1, 5, 10, 10, 5, 1],
[1, 6, 15, 20, 15, 6, 1],
[1, 7, 21, 35, 35, 21, 7, 1],
[1, 8, 28, 56, 70, 56, 28, 8, 1],
[1, 9, 36, 84, 126, 126, 84, 36, 9, 1],
[1, 10, 45, 120, 210, 252, 210, 120, 45, 10, 1]]
```

```
In [37]: # Построение графика для определенного n
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker

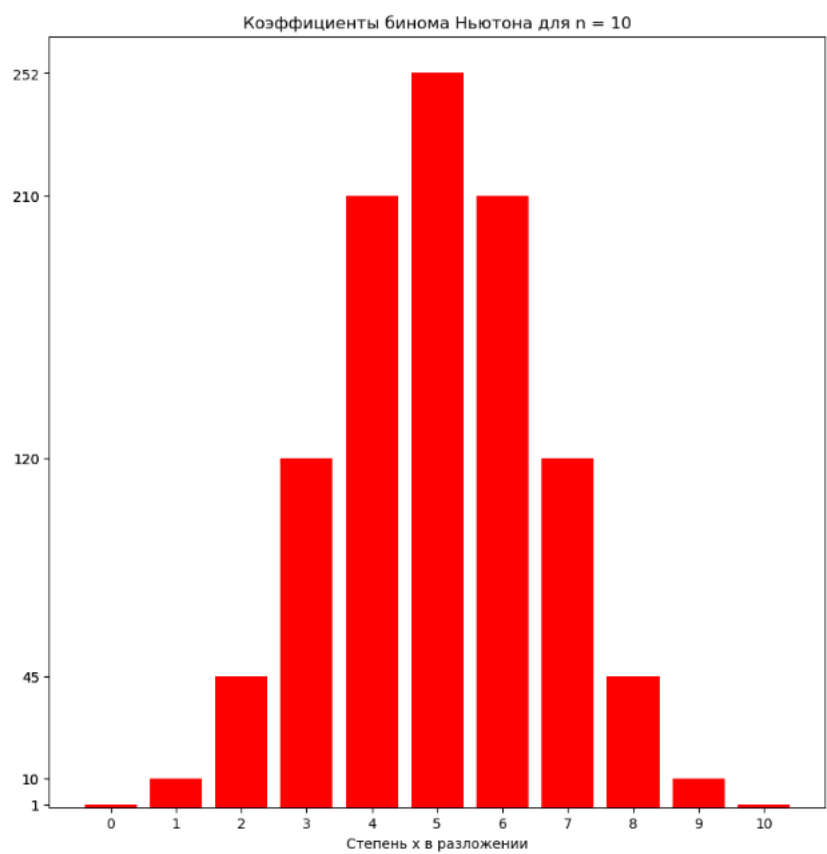
def plot_show(data):
    y = data
    x = list(range(0, len(y)))

    fig, ax = plt.subplots()
    ax.bar(x, y, color = 'r')

    ax.set_yticks(data)
    ax.xaxis.set_major_locator(ticker.MultipleLocator(1))
    ax.set_xlabel('Степень x в разложении')
    ax.set_title('Коэффициенты бинома Ньютона для n = {}'.format(len(data) - 1))

    fig.set_size_inches(10, 10)
    plt.show()
```

```
In [38]: plot_show(bin_coefs_from_1_to_10[-1])
```



# Результаты выполнения

Flask-сервис

← → ↻ ⓘ 127.0.0.1:5000/num/10/rectangle

## Pascal rectangle!

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

1 10 45 120 210 252 210 120 45 10 1