

**ФГАОУ ВО «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»**

**Лабораторная работа №4**

Одномерные массивы

**Вариант № 28**

По дисциплине:

Основы программирования

Выполнил студент 1-го курса группы 243-323

Онищенко А. А.

Проверил

\_\_\_\_\_Никишина И. Н.

**Москва, 2025**

## Постановка задачи

С использованием модуля Random сформировать одномерный массив, состоящий из  $n$  вещественных элементов в котором элементы случайным образом принимают положительный или отрицательный знак и значение от -10 до 10. Для заданного числа  $y$ , такого, что  $a_{\min} < y < a_{\max}$ , вычислить:

1. Произведение элементов массива, значения модуля которого больше  $y$ .
2. Сумму модулей остальных элементов.

## Теоретическая часть

**Массив** — это упорядоченный набор однотипных данных, где каждому элементу присваивается свой порядковый номер (индекс). В Python массивы часто реализуются через списки.

Основные характеристики массива

- **Элемент массива** — отдельная переменная в составе массива
- **Индекс** — порядковый номер элемента (начинается с 0)
- **Размерность** — количество элементов в массиве
- **Одномерный массив** — последовательность элементов, расположенных в одну линию

## Описание программы

Программа написана на алгоритмическом языке Python 3.6, реализована в среде ОС Windows 10 и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

## Описание алгоритма

1. Запросить у пользователя размер массива  $n$  и преобразовать его к целому числу
2. Запросить число  $y$  и преобразовать его к вещественному числу
3. Проверить, что  $y$  находится между минимальным и максимальным элементами массива
4. Создать массив из  $n$  случайных чисел в диапазоне от -10 до 10
5. Инициализировать переменные: произведение = 1, сумма\_модулей = 0
6. Пройти по всем элементам массива:
  - Если модуль элемента больше  $y$  - умножить его на произведение
  - Иначе - добавить модуль элемента к сумме\_модулей
7. Вывести созданный массив
8. Вывести полученное произведение элементов с модулем больше  $y$
9. Вывести сумму модулей остальных элементов

## Описание входных и выходных данных

Входные данные это размер массива  $n$  и число  $y$  исходя из которого происходят вычисления получаемые из пользовательского ввода имеют тип `int` и `float` соответственно. Выходные данные это сам массив размера  $n$ , а также результаты вычислений имеют тип `float`.

## Листинг программы

```
from random import uniform

# Функция для создания массива
def create_array(n):
    # Создаем массив из n случайных чисел от -10 до 10
    return [uniform(-10, 10) for _ in range(n)]

# Функция для вычислений
def calculate(arr, y):
    # Проверяем корректность y
    if not (min(arr) < y < max(arr)):
        raise ValueError("Значение y должно быть между минимальным и максимальным элементами массива")

    # Инициализируем переменные для вычислений
    product = 1 # произведение
    sum_modules = 0 # сумма модулей

    # Проходим по всем элементам массива
    for num in arr:
        # Если модуль числа больше y - умножаем на произведение
        if abs(num) > y:
            product *= num
        # Иначе добавляем модуль числа к сумме
        else:
            sum_modules += abs(num)

    return product, sum_modules

# Основная программа
def main():
    # Ввод данных
    n = int(input("Введите размер массива n: "))
    y = float(input("Введите число y: "))

    # Создаем массив
    array = create_array(n)

    # Выводим созданный массив
    print("Созданный массив:", array)

    try:
        # Вычисляем значения
        prod, summ = calculate(array, y)

        # Выводим результаты
        print(f"Произведение элементов с модулем > {y}: {prod}")
        print(f"Сумма модулей остальных элементов: {summ}")
    except ValueError as e:
        print("Ошибка:", e)

# Запуск программы
if __name__ == "__main__":
    main()
```

## Результаты работы программы

Введите размер массива n: 10

Введите число y: 2

Созданный массив: [-8.728629431630395, 8.874361401414287, -8.861095206550424, -8.722419912059697, -5.09949826529888, -9.69277285119433, -3.9755473987683416, -7.924773719459688, 6.757263171272136, -9.258324582567601]

Произведение элементов с модулем > 2.0: 583270599.977631

Сумма модулей остальных элементов: 0

Введите размер массива n: 10

Введите число y: 4

Созданный массив: [-8.372170825681483, -0.8939628585068391, 5.6104324625753, -0.10533454852046553, -6.839341713273477, 2.9640662520554066, 0.5893178246149411, 3.4680685292081286, -7.544310999483692, -5.039672990782666]

Произведение элементов с модулем > 4.0: 12214.358511108014

Сумма модулей остальных элементов: 8.020750012905781

Введите размер массива n: 20

Введите число y: 8

Созданный массив: [-9.74949808814388, -7.502367544559543, -7.082370919001335, 7.730903322868805, 2.1904685051969803, 2.988684758545844, 1.4309393235008763, -4.304362095693057, -9.474603628156455, 8.906639161019985, 9.938294973690937, -5.191107185308277, 1.4641411581540567, -4.343041836206611, 8.365370836955229, 2.979445364557252, -4.5871226087612875, -8.724084816926068, -4.922736396403369, 8.520300760209118]

Произведение элементов с модулем > 8.0: -5084275.222230996

Сумма модулей остальных элементов: 56.71769101875728

## Список используемой литературы

1. Н.А. Прохоренко, В.А. Дронов, Python 3 и PyQt 5. Разработка приложений: СПб.: БХВ-Петербург, 2017
2. В.П. Рядченко, Методическое пособие по выполнению лабораторных работ.