**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА   
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет управления

Кафедра Информатики и информационных технологий

Направление подготовки / специальность: 09.04.03 Прикладная информатика

Направленность (профиль) / специализация: Информационные технологии в ГМУ

**Отчет по лабораторным работам**

|  |  |
| --- | --- |
| по дисциплине: | Алгоритмы сортировки, поиска и сжатия данных |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  | **АВТОР** |
|  |  | Обучающийся 2 курса группы Ик-721 |
|  |  | заочной формы обучения |
|  |  |  |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ Соколов Дмитрий Александрович |
|  |  | *(подпись) (фамилия, инициалы)* |
|  |  |  |
|  |  | **РУКОВОДИТЕЛЬ** |
|  |  | Кандидат физико-математических наук |
|  |  | *(ученая степень, ученое звание)* |
|  |  | оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г. |
|  |  | *(дата защиты)* |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ Орлов М.Л. |
|  |  | *(подпись) (фамилия, инициалы)* |

Нижний Новгород, 2025г.

Оглавление

[Практическое задание 1 3](#_Toc188559917)

[Задание 1 3](#_Toc188559918)

[Задание 2 5](#_Toc188559919)

[Задание 3 7](#_Toc188559920)

[Практическое задание 2 9](#_Toc188559921)

[Задание 1 9](#_Toc188559922)

[Задание 2 10](#_Toc188559923)

[Задание 3 11](#_Toc188559924)

[Задание 4 12](#_Toc188559925)

[Задание 5 14](#_Toc188559926)

[Задание 6 15](#_Toc188559927)

[Задание 7 18](#_Toc188559928)

[Практическое задание 3 19](#_Toc188559929)

[Задание 1 19](#_Toc188559930)

[Задание 2 23](#_Toc188559931)

[Практическое задание 4 27](#_Toc188559932)

[Задание 1 27](#_Toc188559933)

[Задание 2 28](#_Toc188559934)

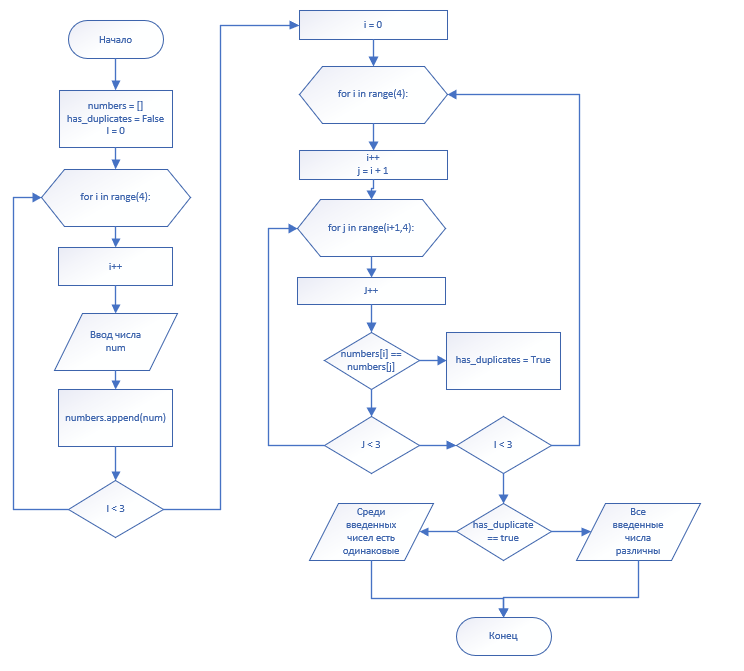
[Практическое задание 5 29](#_Toc188559935)

# Практическое задание 1

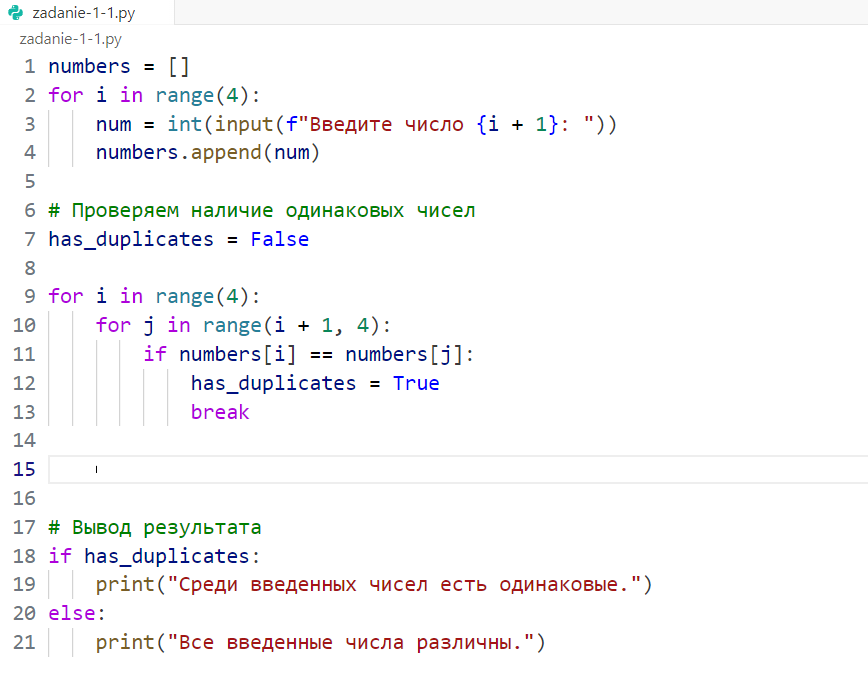
## Задание 1

Разработайте алгоритм, который запрашивает у пользователя четыре целых числа и затем определяет, есть ли среди введенных чисел одинаковые или нет. Нарисуйте блок-схему алгоритма. Программный код может быть реализован на одном из следующих языков программирования: С, С++, С#, Java.

**Блок-схема**

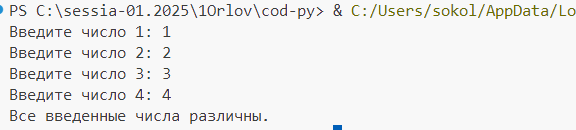


**Программа**

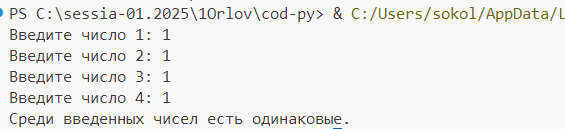


**Вывод результатов на экран**

**Результат 1**

****

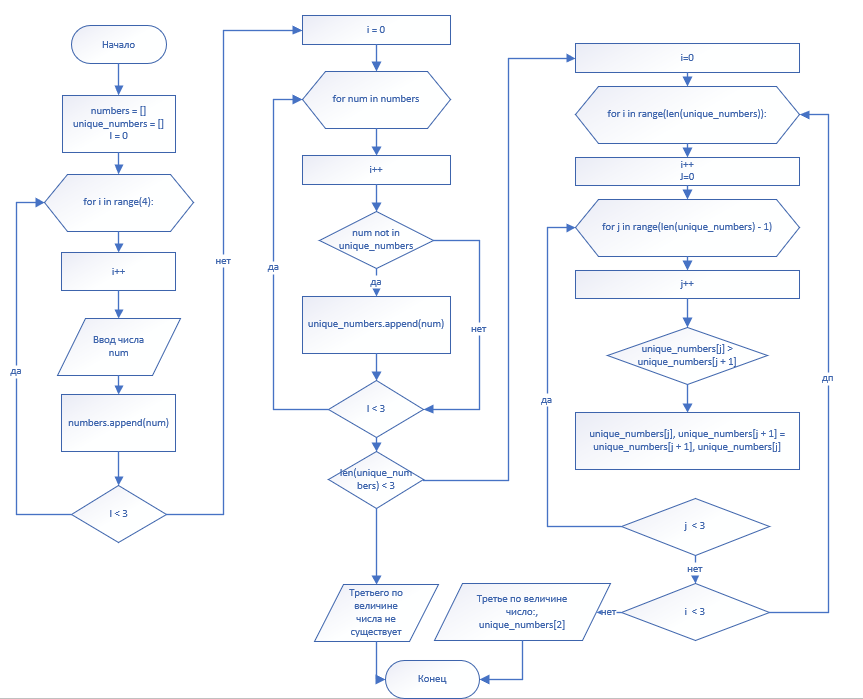
**Результат 2**



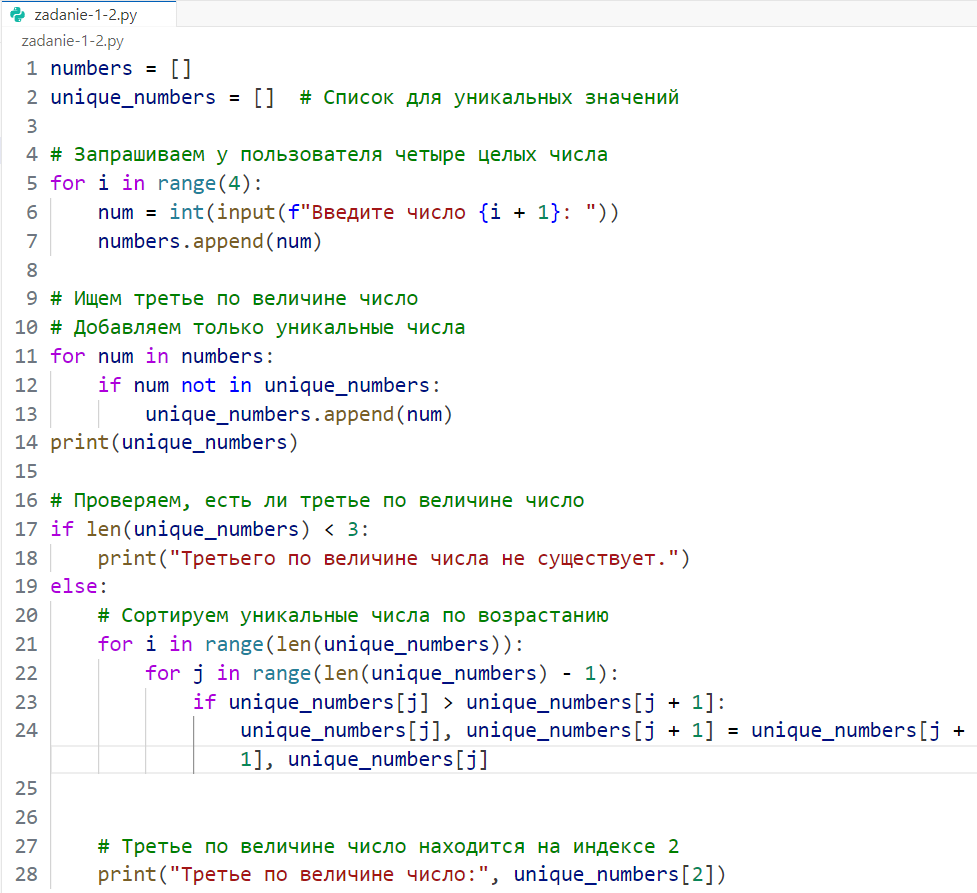
## Задание 2

Разработайте алгоритм, который запрашивает у пользователя четыре целых числа и затем находит третье по величине число, если оно существует. Нарисуйте блок-схему алгоритма. Программный код может быть реализован на одном из следующих языков программирования: С, С++, С#, Java.

**Блок-схема**

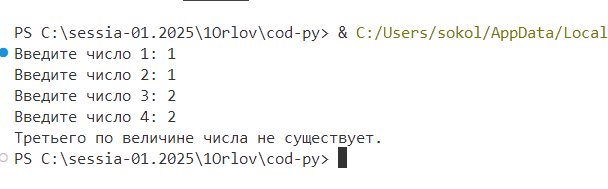
****

**Программа**

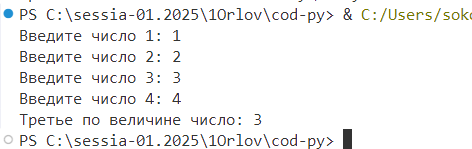
****

**Вывод результатов на экран**

**Результат 1**

****

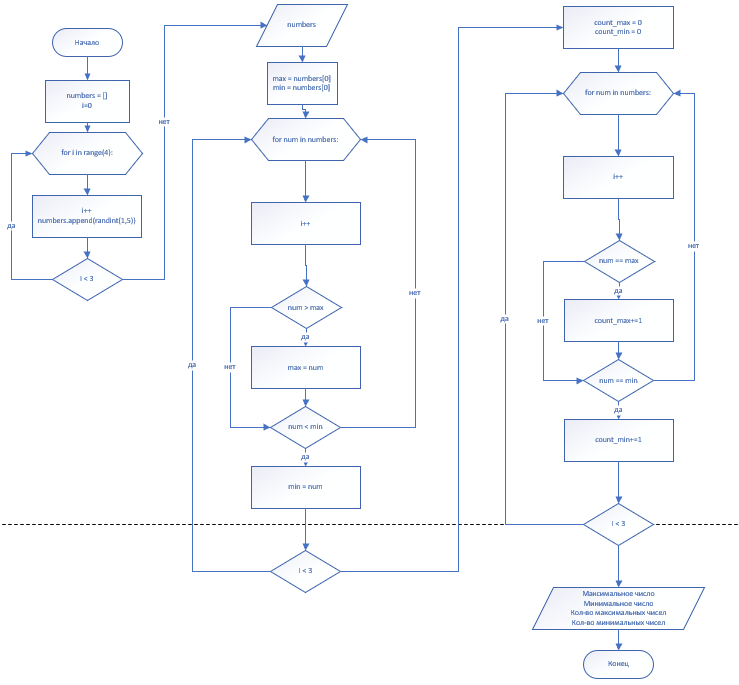
**Результат 2**

****

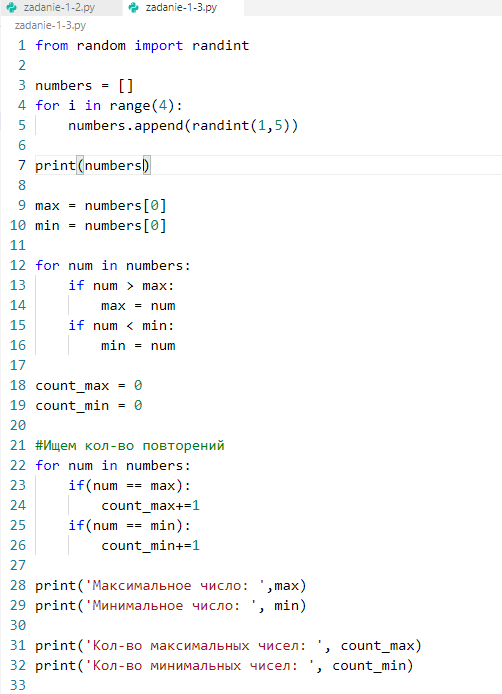
## Задание 3

Разработайте алгоритм, который случайным образом задает четыре целых числа и затем определяет максимальное и минимальное число, а также находит количество максимальных и минимальных чисел среди введенных. Нарисуйте блок-схему алгоритма. Программный код может быть реализован на одном из следующих языков программирования: С, С++, С#, Java.

**Блок-схема**

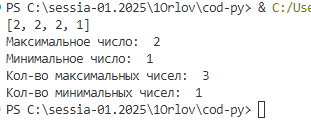
****

**Программа**

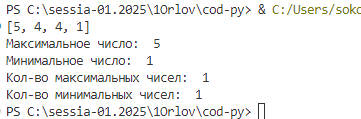
****

**Вывод результатов на экран**

**Результат 1**

****

**Результат 2**

****

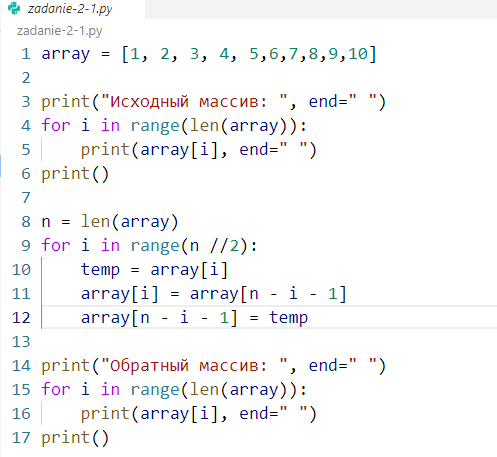
# Практическое задание 2

**Вариант 3**

## Задание 1

Создайте одномерный массив. Выведите его содержимое на экран. Выполните реверс содержимого массива (переверните наоборот: первый элемент меняется местами с последним, второй с предпоследним и т.д.). Выведите содержимое на экран.

**Программа**

****

**Вывод результатов на экран**

**Результат 1**

****

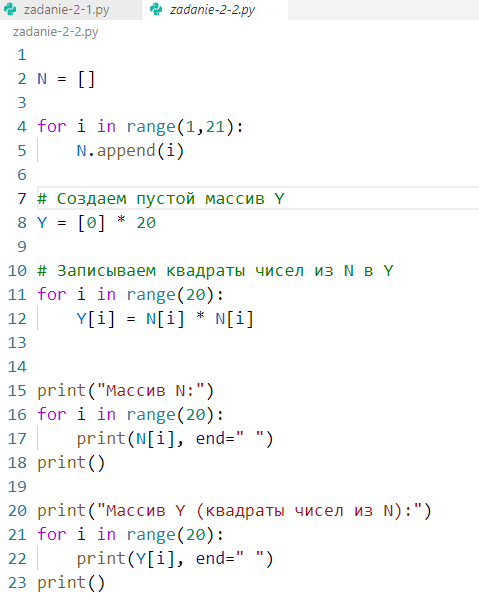
**Результат 2**

****

## Задание 2

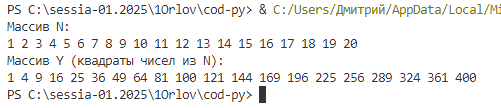
Записать в массив Y(20) квадраты чисел, указанных в массиве N(20).

**Программа**

****

**Вывод результатов на экран**

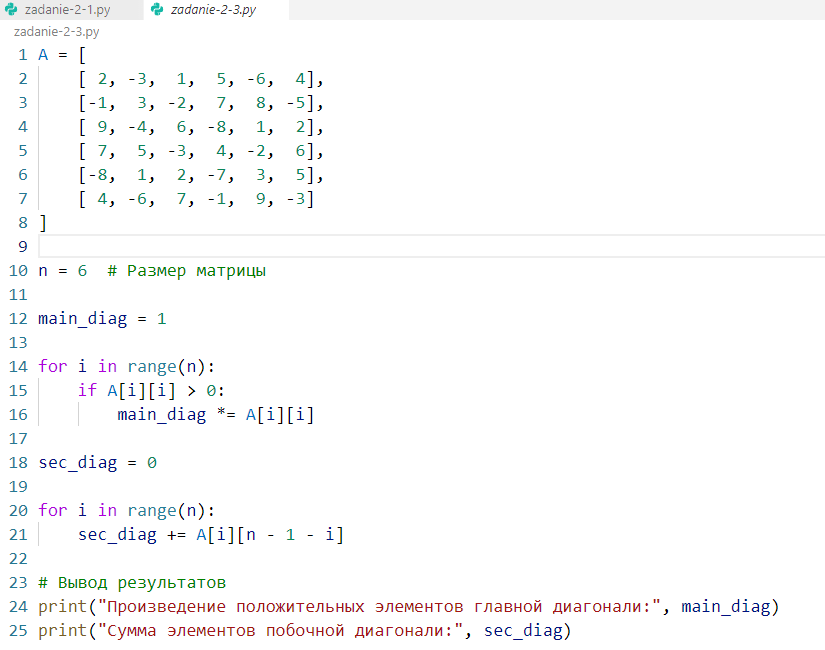
**Результат 1**

****

## Задание 3

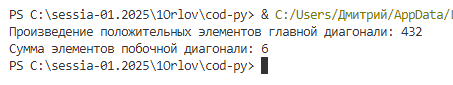
В матрице A(6,6) найти произведение положительных элементов главной диагонали и сумму всех элементов побочной диагонали.

**Программа**

****

**Вывод результатов на экран**

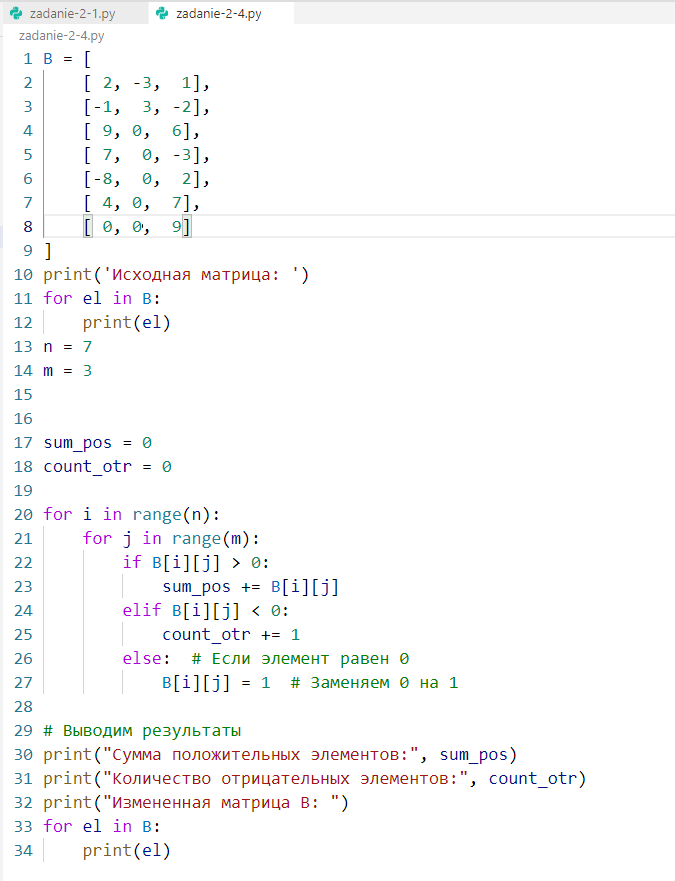
**Результат 1**

****

## Задание 4

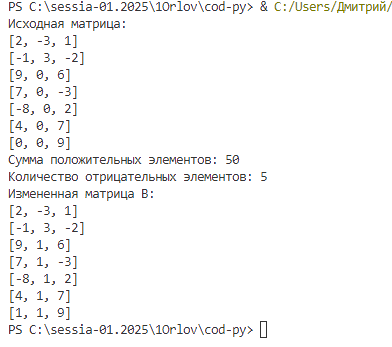
В матрице B(7,3) найти сумму всех положительных элементов, количество отрицательных элементов, а если Bij=0, то вместо него записать +1.

**Программа**

****

**Вывод результатов на экран**

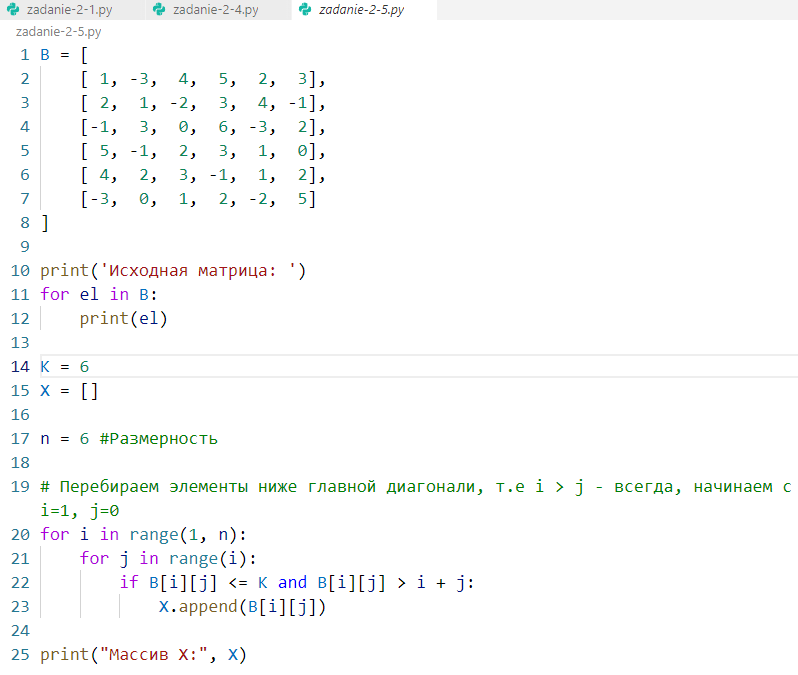
**Результат 1**

****

## Задание 5

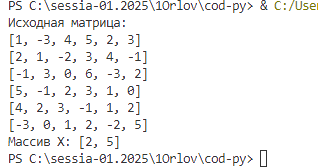
Среди элементов, расположенных ниже главной диагонали матрицы B(6,6), найти элементы, удовлетворяющие условию Bij<=K и Bij>i+j, и сформировать из них одномерный массив X.

**Программа**

****

**Вывод результатов на экран**

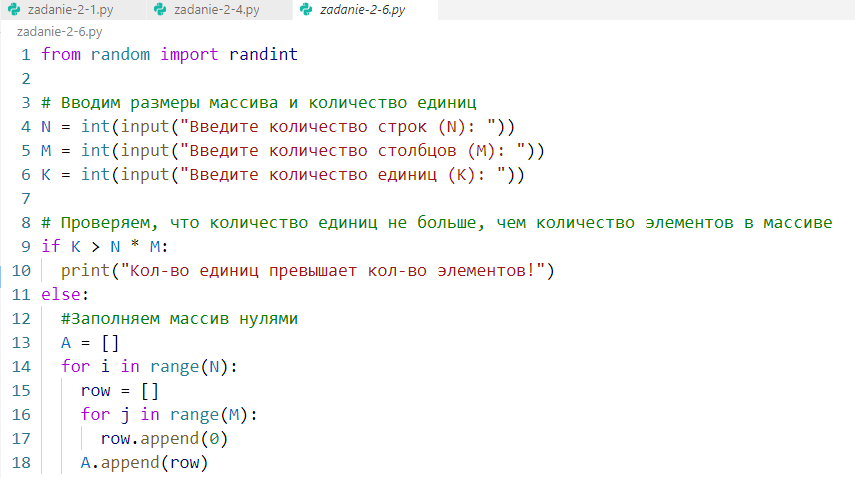
**Результат 1**

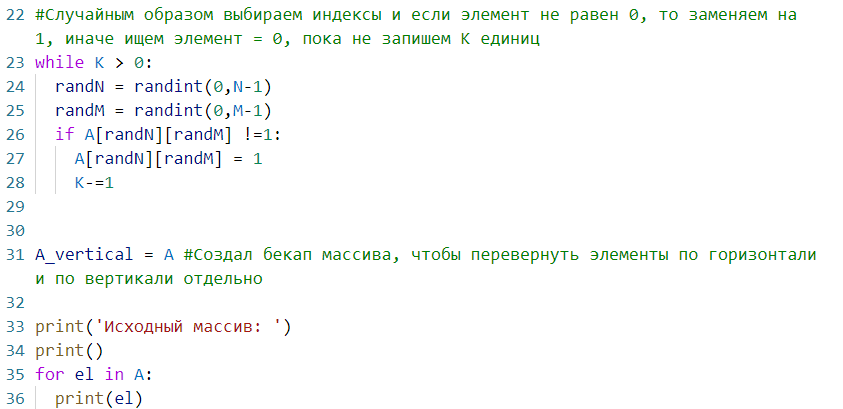
****

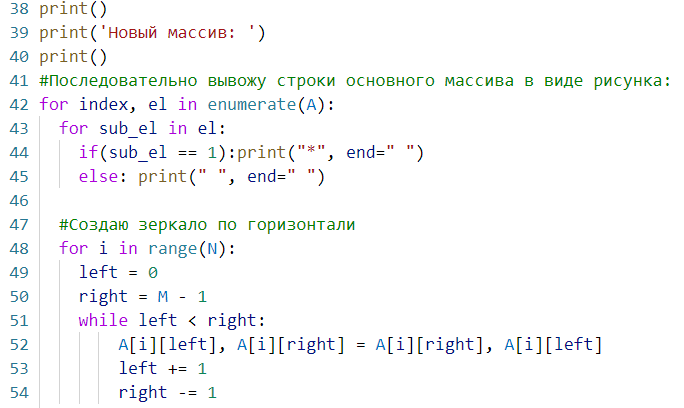
## Задание 6

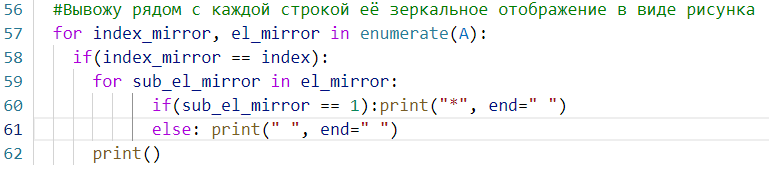
Создайте двумерный массив А(N,M). Значения N и M вводит пользователь с клавиатуры. Элементы массива могут принимать 2 значения – 0 и 1. Количество 1 (K) определяет пользователь. Эти 1 распределяются по массиву случайнам образом.

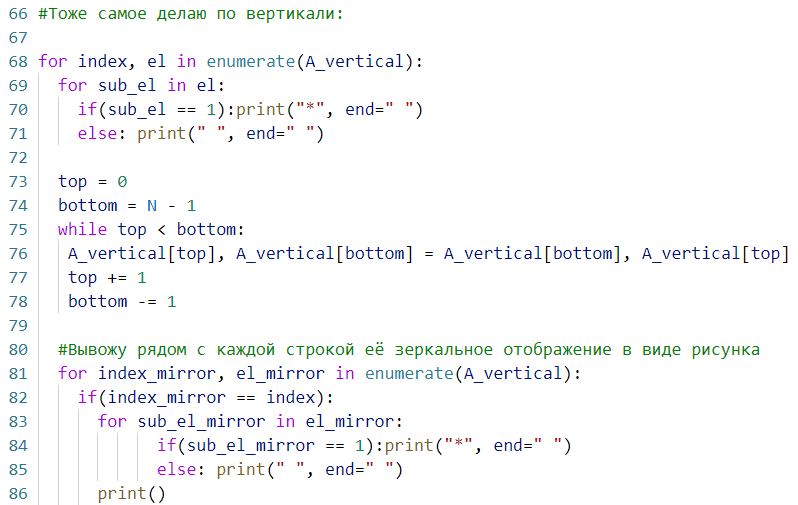
**Программа**

****

****

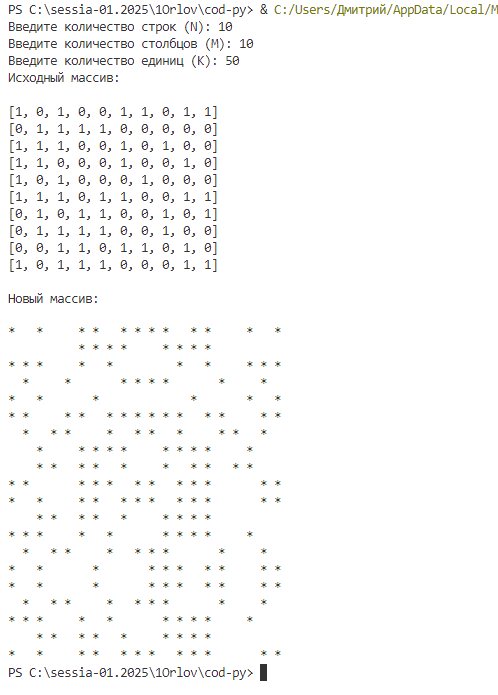
****

****

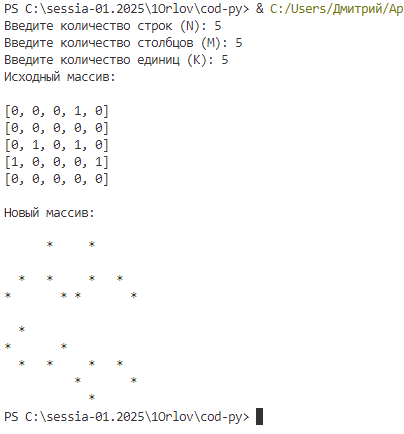
****

**Вывод результатов на экран**

**Результат 1**

****

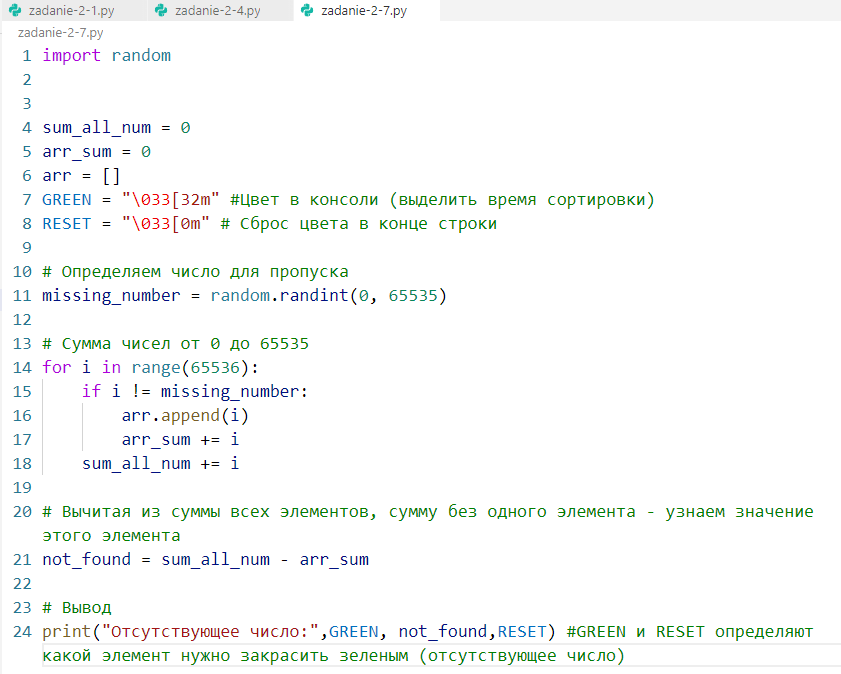
**Результат 2**

****

## Задание 7

Создайте одномерный массив, который содержит полный набор возможных значений типа UInt16, кроме одного. Определите, какого числа не хватает.

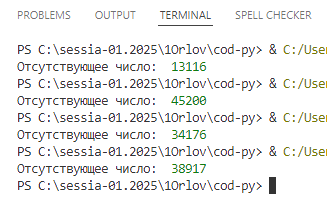
**Программа**

****

**Вывод результатов на экран**

**Результат 1**

**Несколько запусков программы**

****

# Практическое задание 3

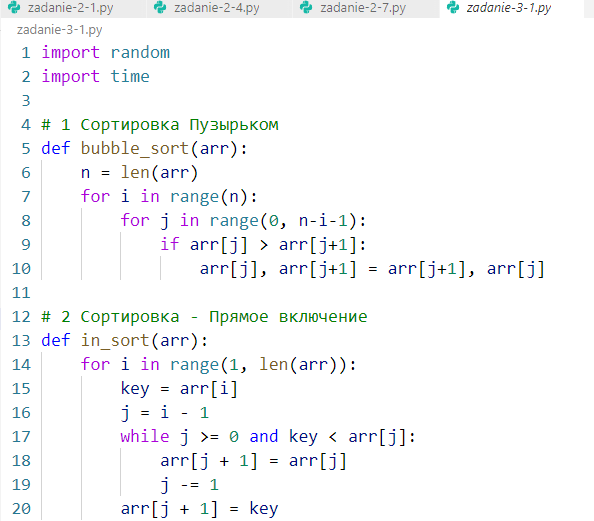
## Задание 1

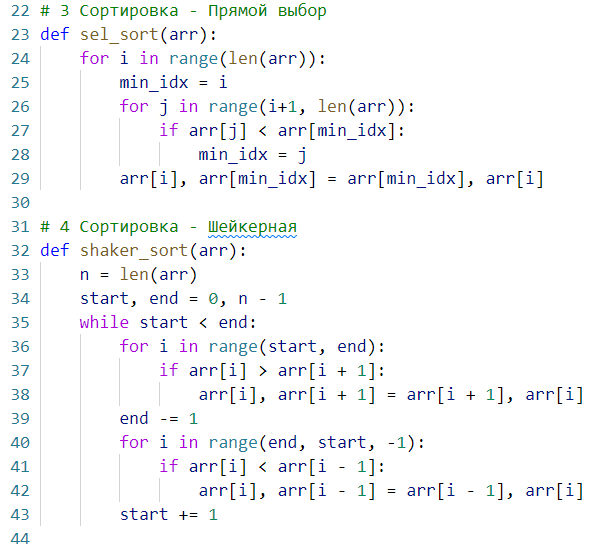
Написать единую программу на языке высокого уровня (С++, C#, Java), реализующую сортировку массива, заполненного случайными числами в диапазоне от 0 до 100, различными методами. Провести сравнительный анализ временной эффективности реализованных методов.

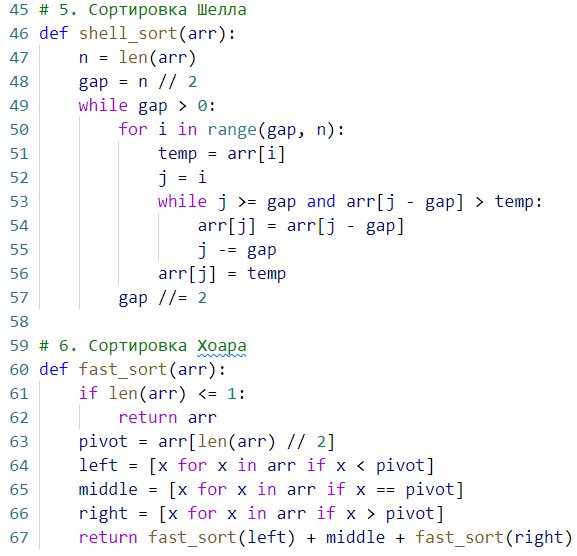
Должны быть реализованы следующие методы сортировки:

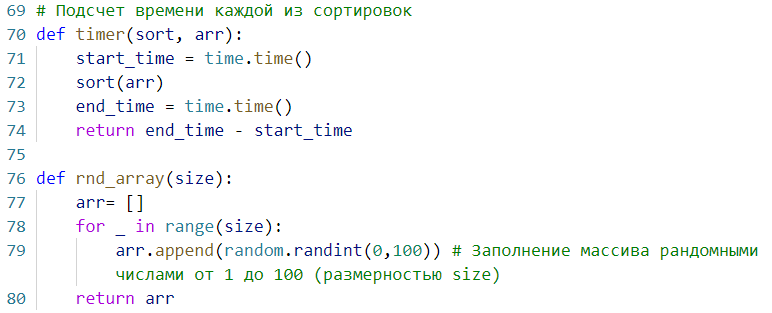
1. Сортировка методом прямого обмена (сортировка методом пузырька).
2. Сортировка методом прямого включения.
3. Сортировка методом прямого выбора.
4. Шейкерная сортировка.
5. Сортировка методом Шелла.
6. Сортировка методом Хоара.

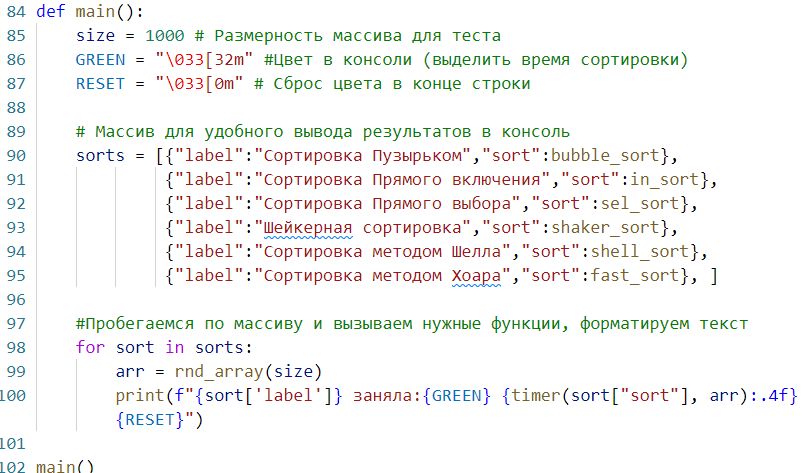
**Программа**

****

****

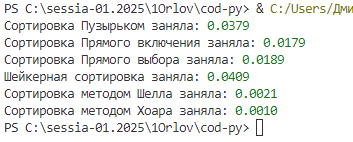
****

****

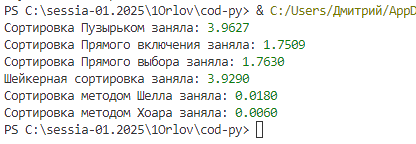
****

**Вывод результатов на экран**

**Результат 1 (Массив из 1000 элементов)**

****

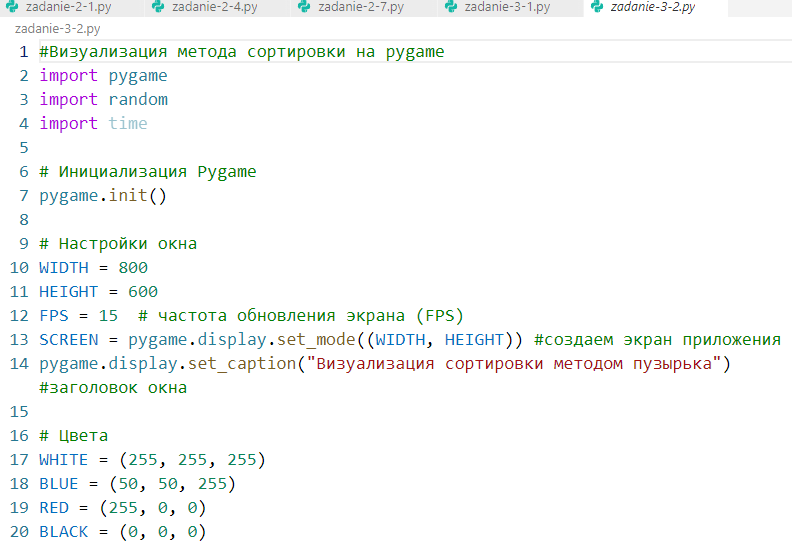
**Результат 2 (Массив из 10 000 элементов)**

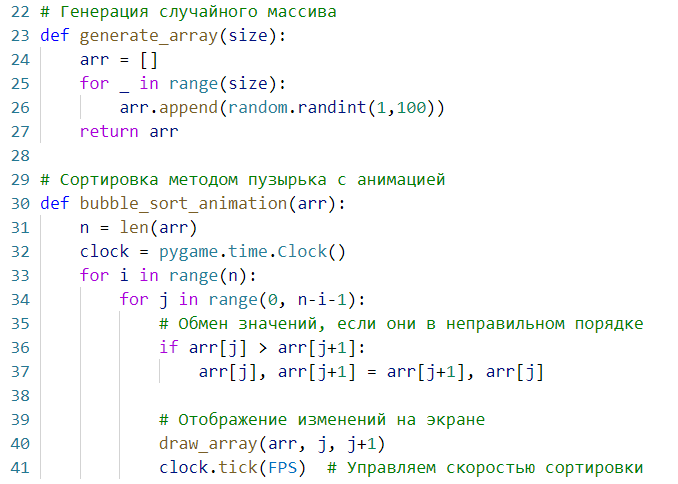
****

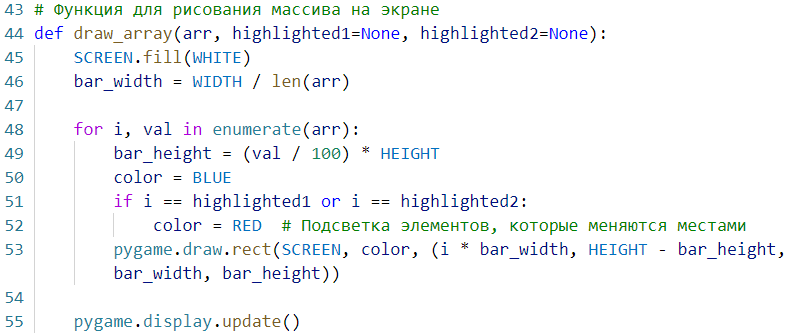
Более наглядно видна разница во времени, с огромным отрывом методы Шелла и Хоара.

## Задание 2

**Программа**

****

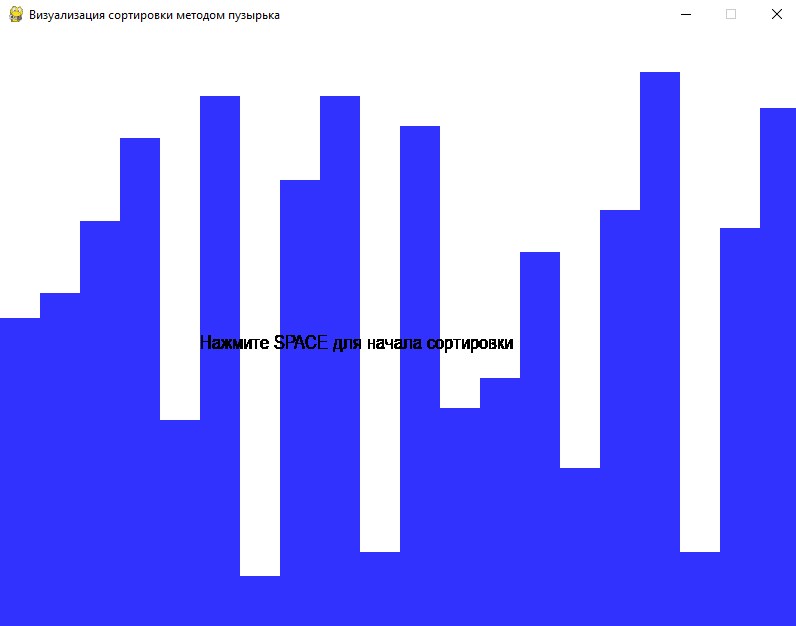
****

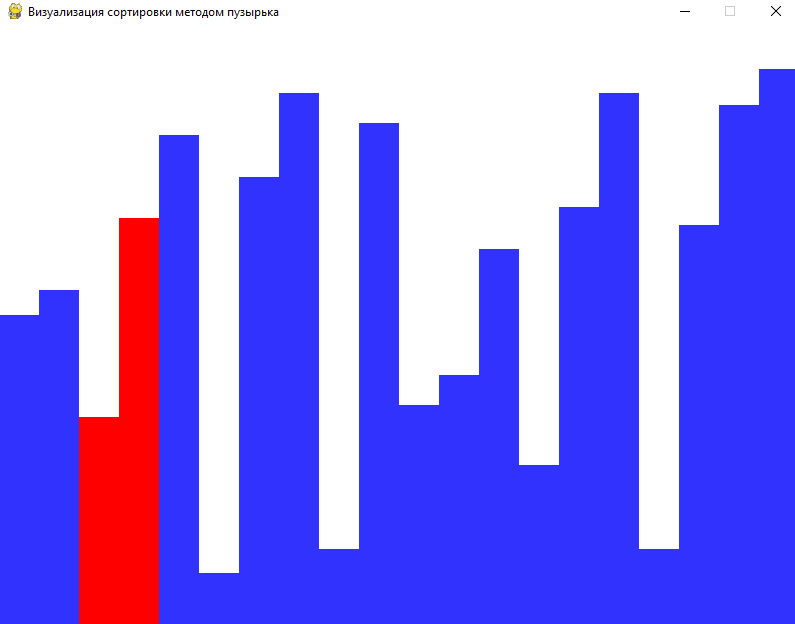
****

****

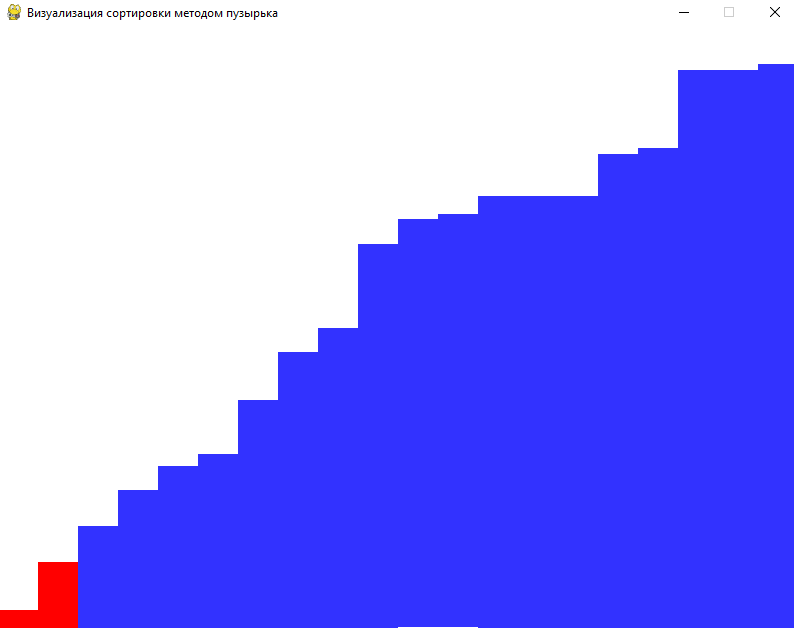
**Вывод результатов на экран**

**Результат 1**

****

****

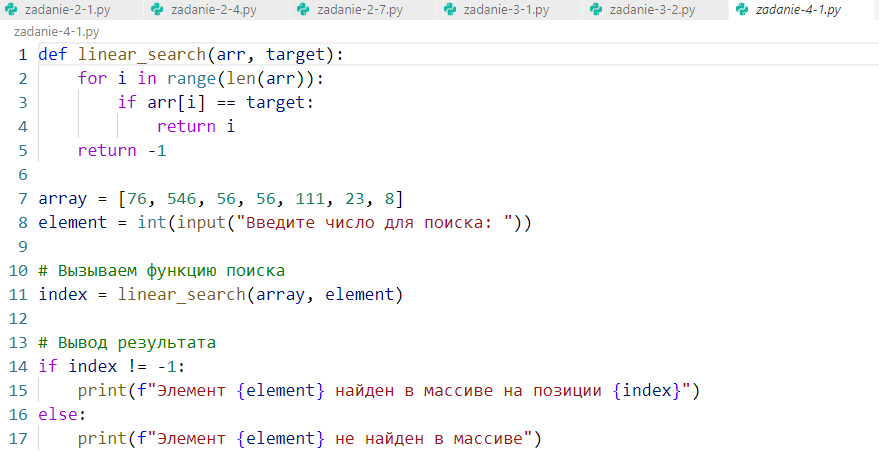
**Итог**

****

# Практическое задание 4

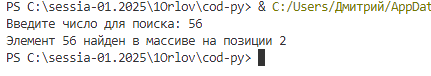
## Задание 1

**Программа**

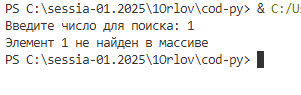
****

**Вывод результатов на экран**

**Результат 1**

****

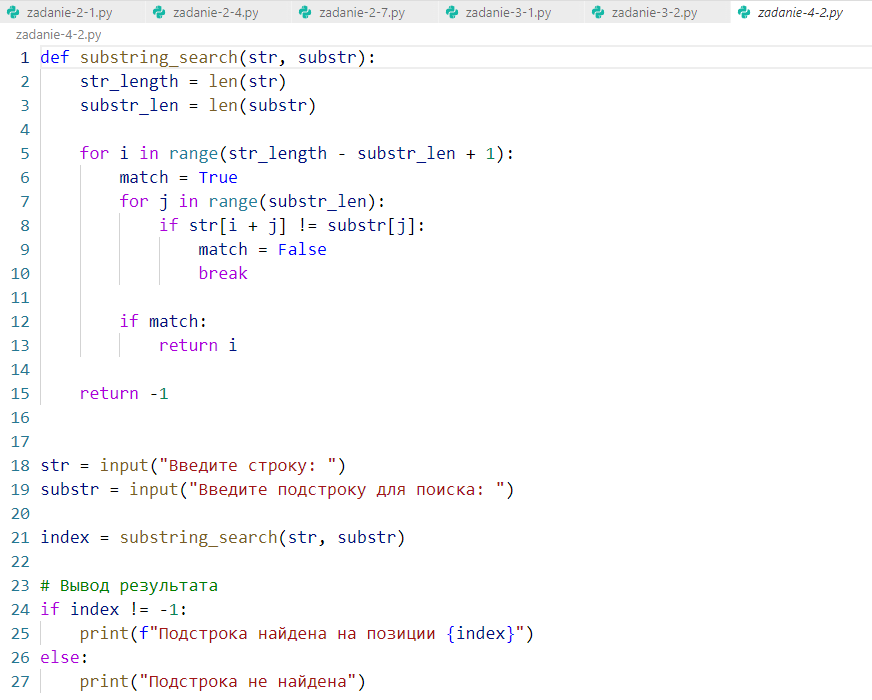
**Результат 2**

****

Задание - найти элемент в неупорядоченном массиве, поэтому использую линейный поиск, но более быстрый – бинарный, где необходимо отсортировать массив по возрастанию.

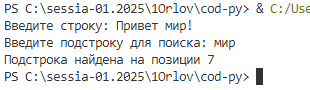
## Задание 2

**Программа**

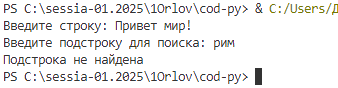
****

**Вывод результатов на экран**

**Результат 1**

****

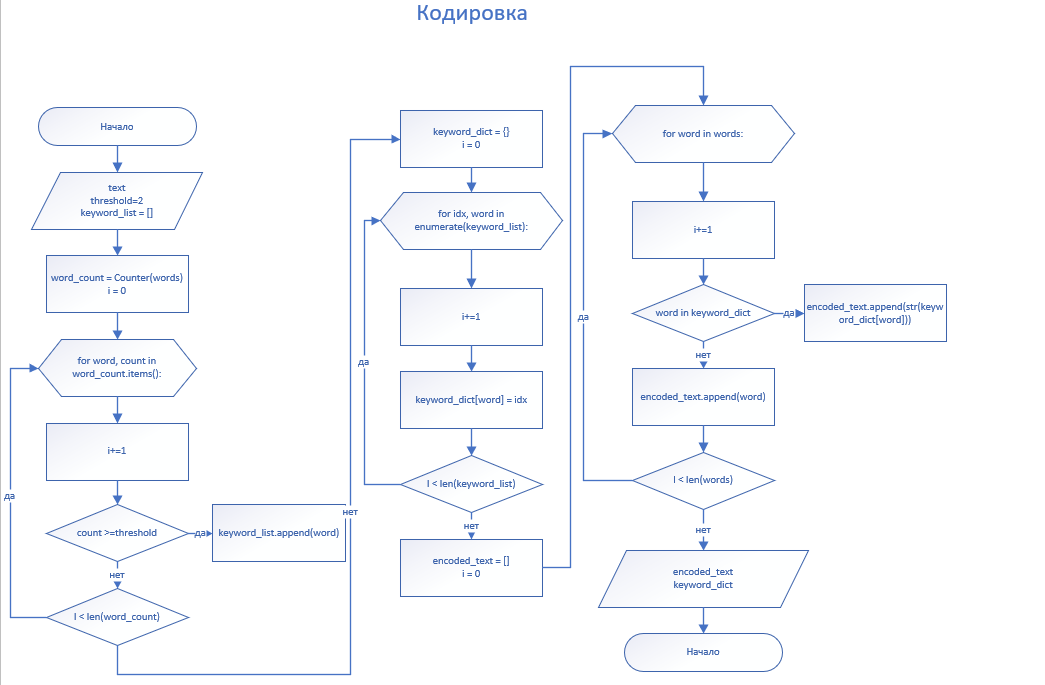
**Результат 2**

****

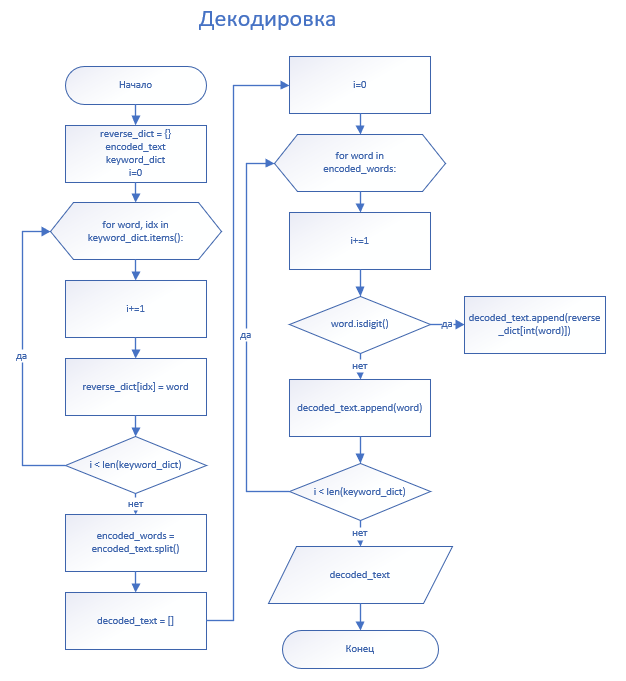
# Практическое задание 5

**Блок-схема**

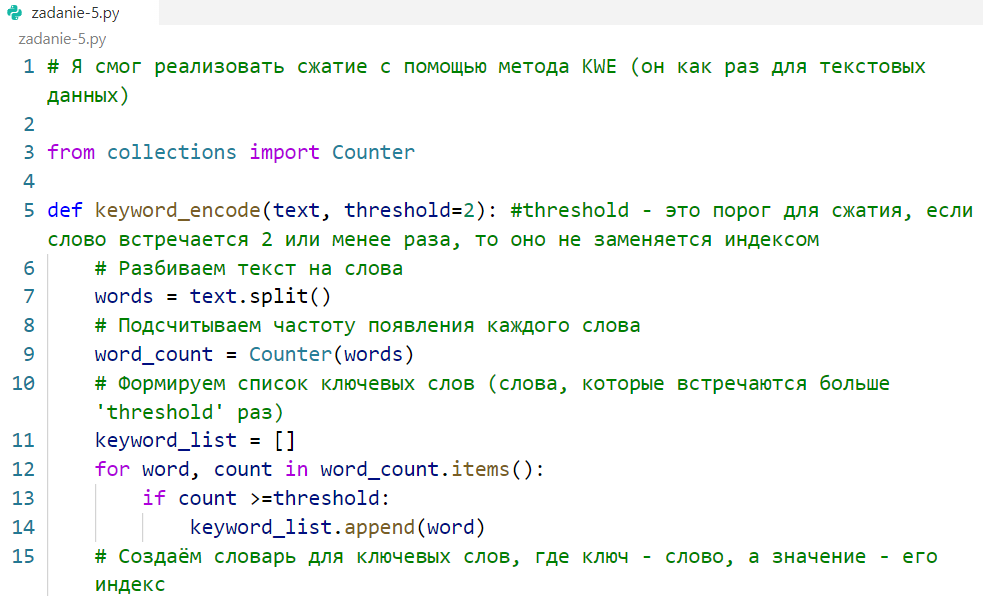
**Кодировка**

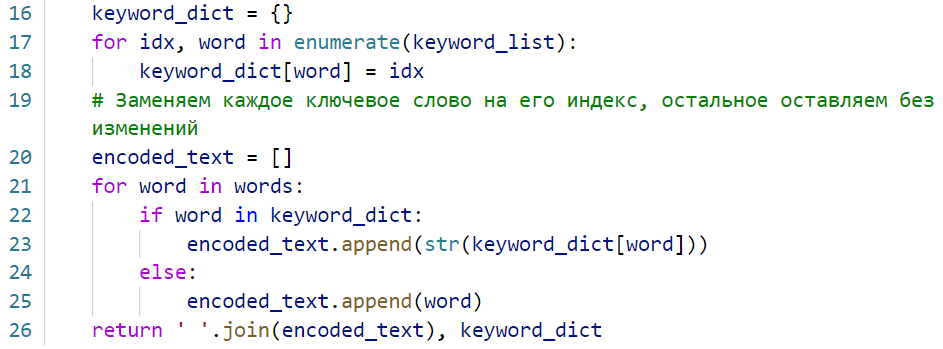
****

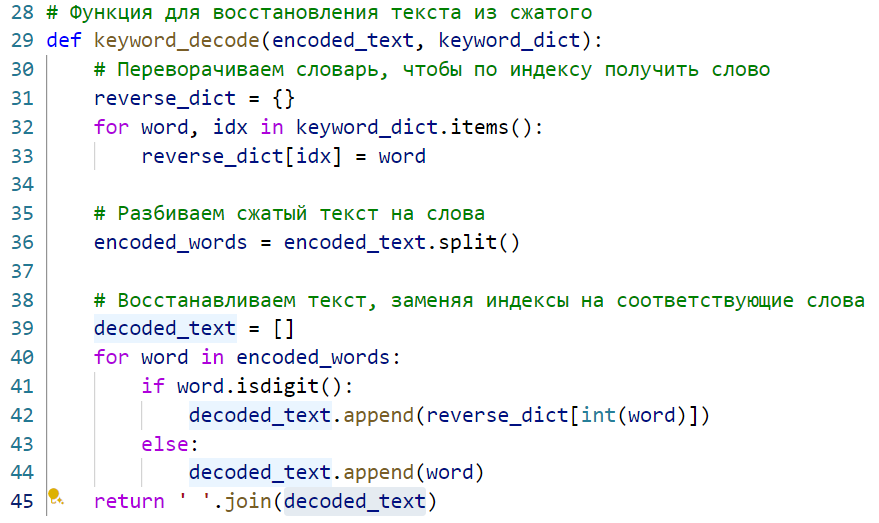
**Декодирование**

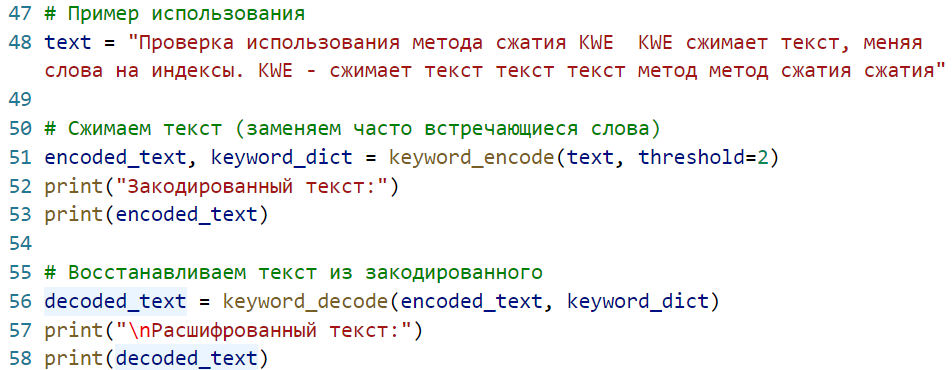
****

**Программа**

****

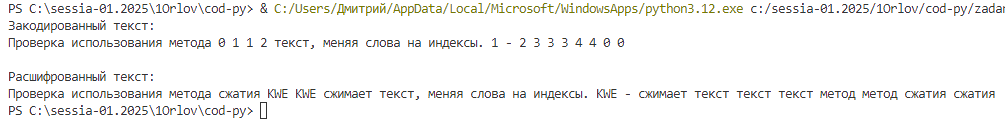
****

****

****

**Вывод результатов на экран**

**Результат**

****