Лабораторная работа №2

Дисциплина: Математические основы защиты информации и информационной безопасности

Аветисян Давид Артурович

Содержание

# 1 Цель работы

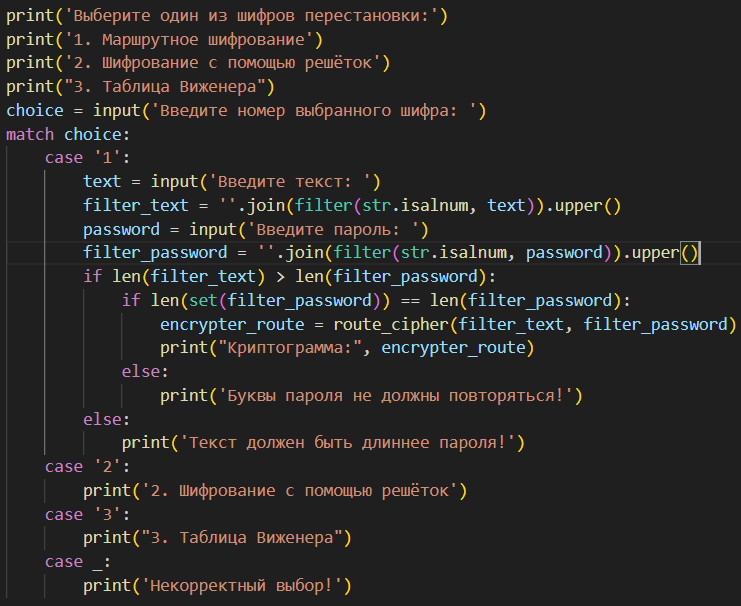
Познакомиться с шифрами перестановки.

# 2 Задание

1. Программно реализовать маршрутное шифрование.
2. Программно реализовать шифрование с помощью решёток.
3. Программно реализовать шифр Виженера.

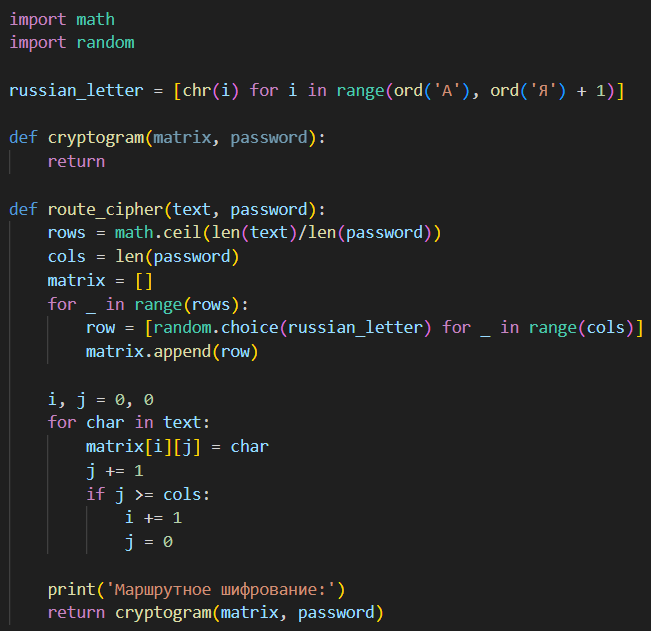
# 3 Выполнение лабораторной работы

1. Все шифрования я реализовывал на языке python. Сначала я реализовал возможность выбора одного из трёх шифров пользователем. Далее при помощи match-case я реализовал выполнение конкретных шифрований. Реализацию я начал с маршрутного шифрования. Я сделал запрос текста и пароля у пользователя, их фильтрацию на наличие пробелов, цифр или знаков пунктуации, а также возвёл всё в верхний регистр. Также я добавил проверки текста и пароля на соответствие требованиям для шифрования.



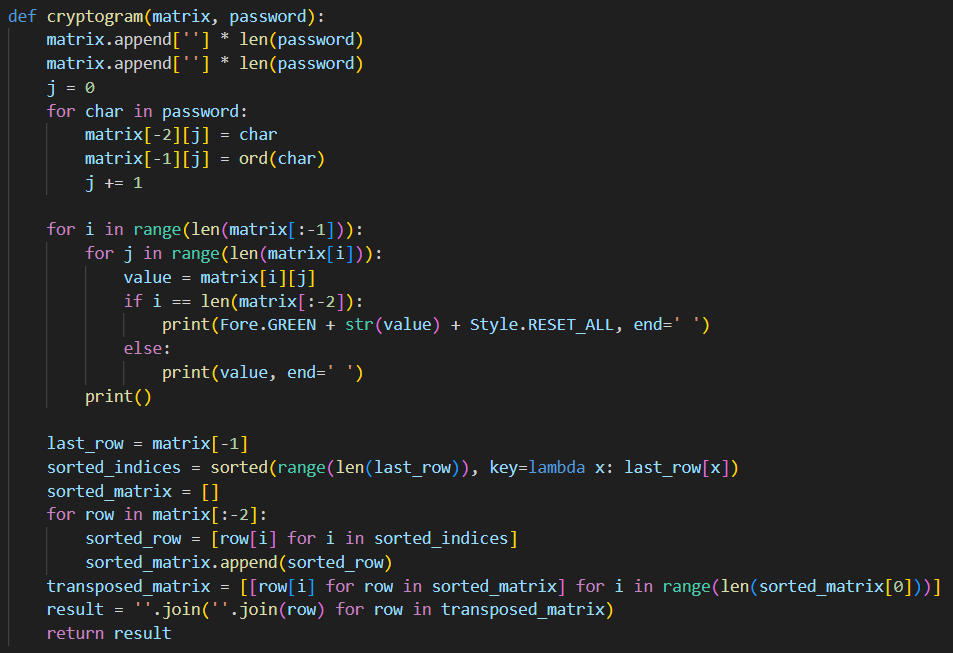
Начало реализации маршрутного шифрования

Я реализовал русский алфавит для удобного заполнения массивов. Далее я реализовал функцию route\_cipher, в которой создал матрицу размером с текст пользователя и шириной с его пароль. Я заполнил её побуквенно текстом пользователя, а пустые места заполнил случайными русскими буквами.



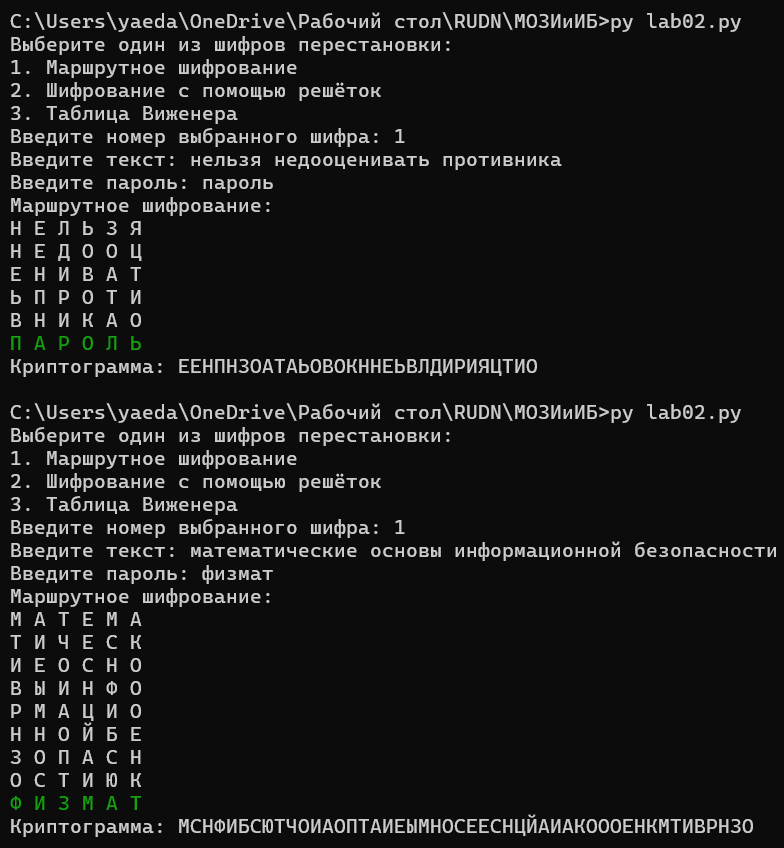
Реализация маршрутного шифрования

После я реализовал общую для первого и второго шифрования функцию cryptogram, которая из заданного массива и введённого пользователем пароля создаёт таблицу, которую выводит, а затем создаётся криптограмму. Данная функция сначала добавляет к одному массиву 2 строчки (с паролем и ASCII номерами букв этого пароля). Далее она берёт последнюю строчку матрицы, сортирует по возрастанию, а затем формирует новую матрицу из старой, учитывая отсортированную последнюю строку. Этот функционал схож как для маршрутного шифрования, так и для шифрования с помощью решёток.



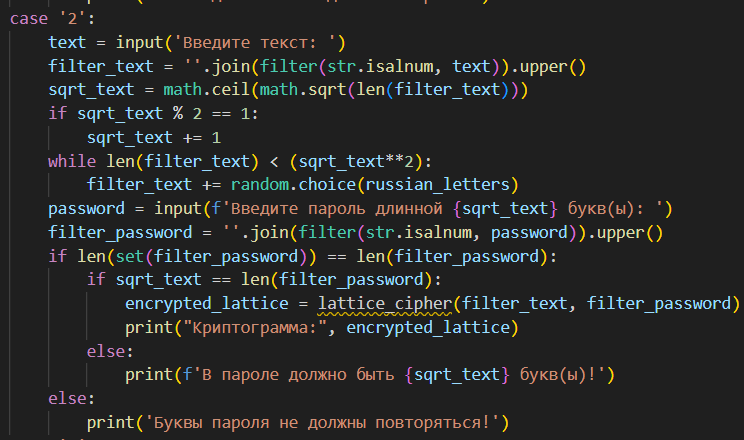
Реализация формирования криптограммы

Далее я запустил два теста через командную строку. Один тест как в теории к лабораторной работе №2. Второй тест для дополнительной проверки. Шифрование совпало с тестом в лабораторной работе №2, и реализовано верно.



Проверка реализации маршрутного шифрования

1. Затем я перешёл к реализации шифрования с помощью решёток. Я аналогично предыдущему шифрованию запросил текст у пользователя, но в данном случае я запрашиваю пароль необходимой длины 2k, как сказано в теории к лабораторной работы. При этом длина текста N должна быть равна k\*\*2. Также я аналогично проверяю текст и пароль на соответствие требованиям для шифрования.



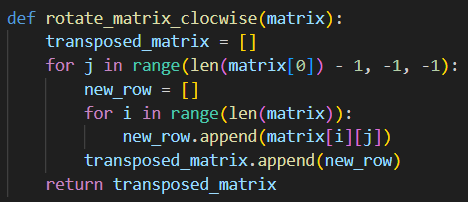
Начало реализации шифрования с помощью решёток

Далее я реализовал функцию lattice\_cipher, в которой я создал матрицу размера 2k и заполнил её нулями. Далее я заполнял каждую четверть значениями от 1 до k\*\*2 в соответствии с инструкцией учебника. Далее я определил случайным образом по одному уникальному значению из получившейся матрицы и записал их индексы. Далее я вывел данную матрицу, закрашивая выбранные значения красным цветом.



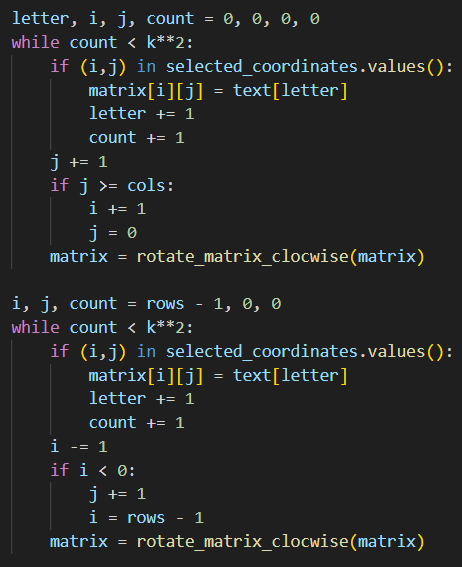
Реализация шифрования с помощью решёток

Для того чтобы заполнить матрицу значениями от 1 до k\*\*2, я реализовал функцию вращения матрицы на 90 градусов по часовой стрелке.

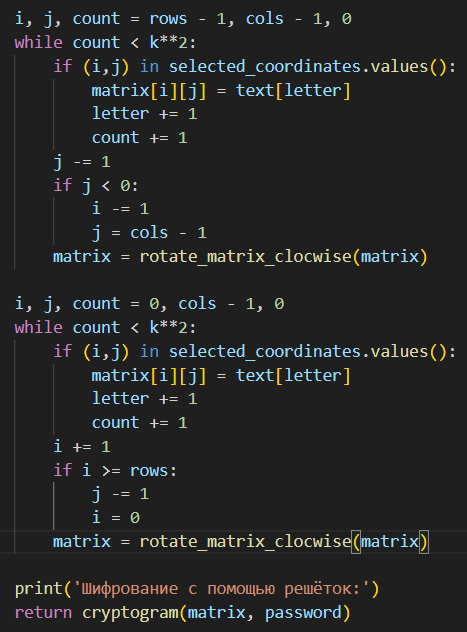


Функция вращения матрицы на 90 градусов по часовой стрелк

Далее для правильного заполнения матрицы текстом пользователя, я реализовал 4 похожих, но разных цикла. Сначала я заполнял все уникальные значения побуквенно текстом пользователя. Затем я поворачивал матрицу на 90 градусов по часовой стрелке, и менял направление заполнения матрицы в соответствии с описанием в лабораторной работе №2. И обязательно в конце использовал ранее реализованную функцию формирования криптограммы.

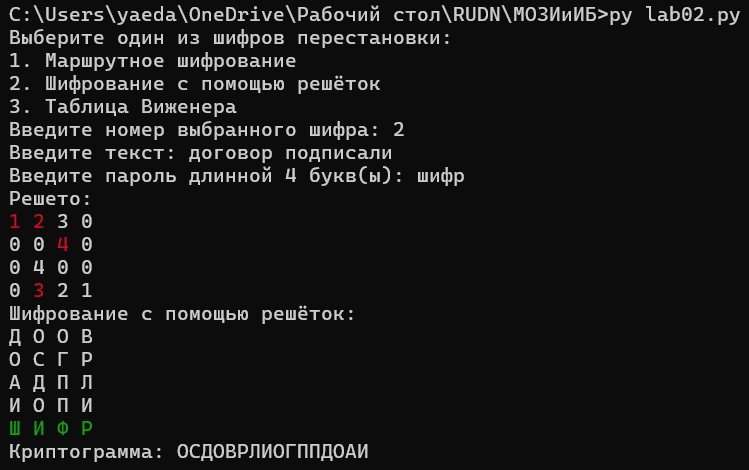


Первый и второй цикл заполнения матрицы текстом пользователя

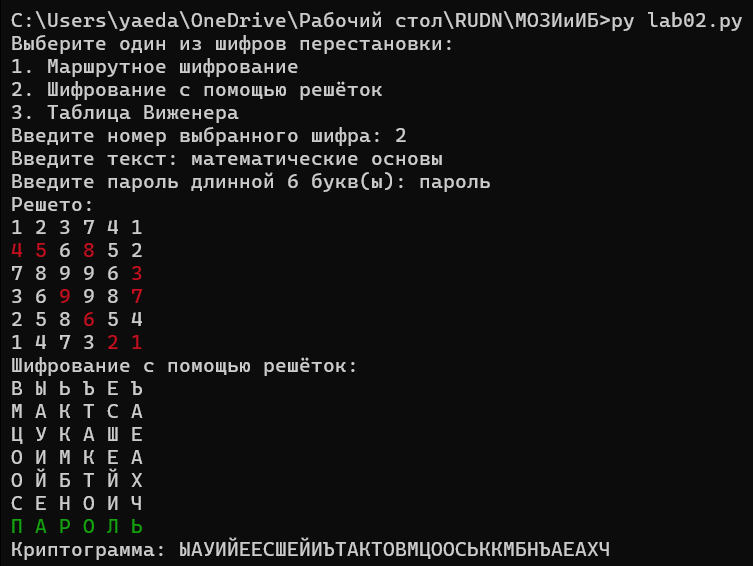


Третий и четвёртый цикл заполнения матрицы текстом пользователя

Далее я запустил два теста через командную строку. Один тест как в теории к лабораторной работе №2. Второй тест для дополнительной проверки. Шифрование совпало с тестом в лабораторной работе №2, и реализовано верно.

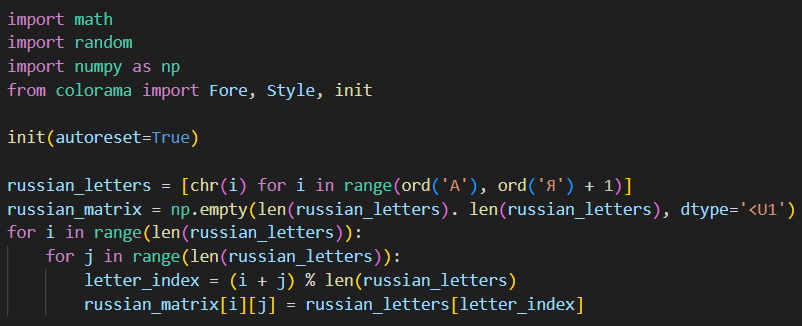


Тест реализации маршрутного шифрования из лабораторной работы №2



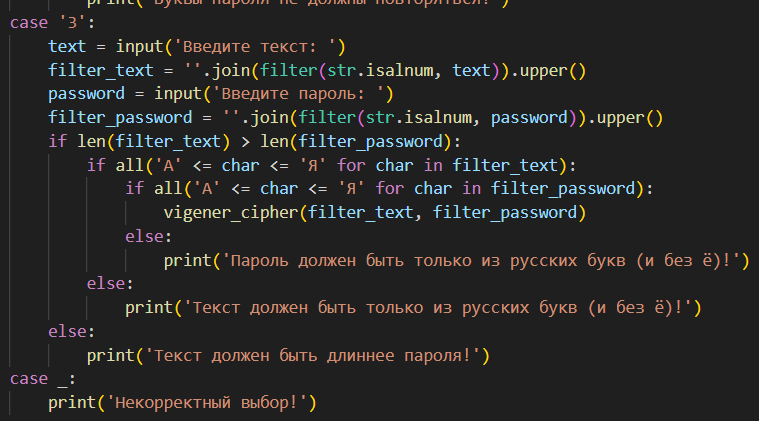
Тест реализации маршрутного шифрования для дополнительной проверки

1. И в конце я перешёл к реализации шифра Виженера. В первую очередь для его реализации нам потребуется таблица с русским алфавитом, где каждая следующая строка сдвигается на одну букву. Данную таблицу я реализовал в виде матрицы, использую ранее составленную матрицу с русским алфавитом.



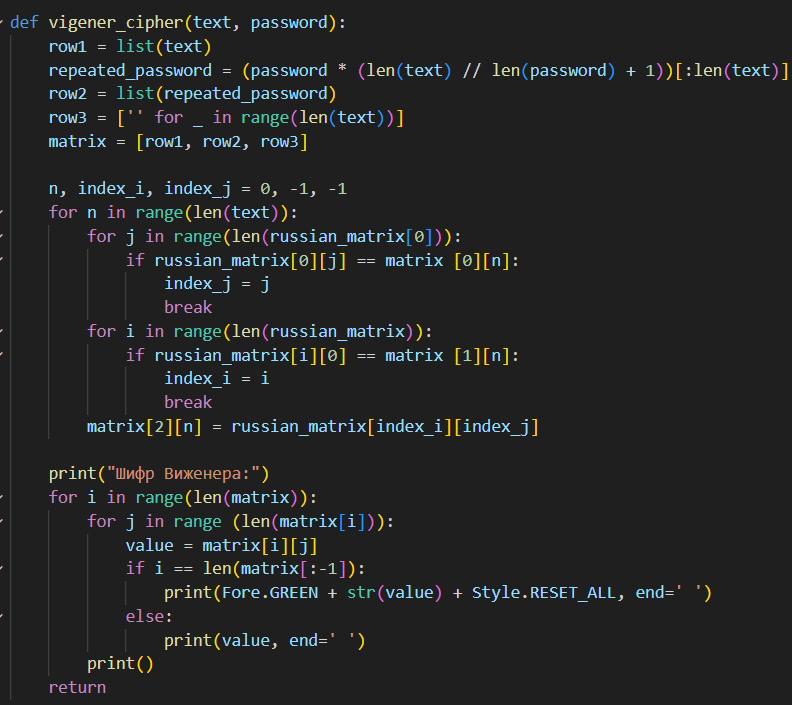
Реализация таблицы с русским алфавитом

Затем я перешёл к реализации шифра Виженера. Я аналогично предыдущим шифрованиям запросил текст и пароль у пользователя, отфильтровал их, а также проверил на соответствие требованиям для шифрования. Так как для шифрования используется таблица с русским алфавитом, то и текст и пароль должны содержать только русские буквы.



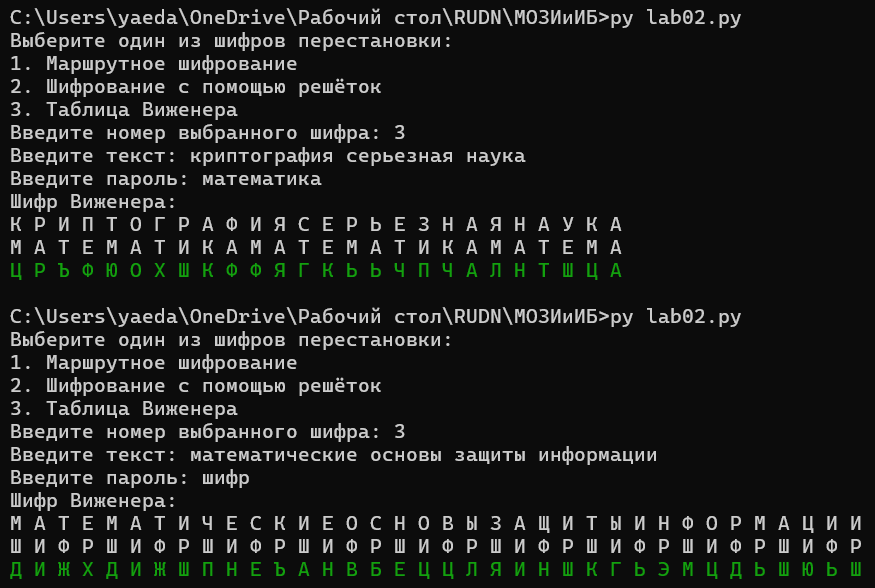
Начало реализации шифра Виженера

Далее я реализовал функцию vigener\_cipher, в которой создаётся матрица, где первая строка - текст пользователя, вторая строка - повторяющийся пароль пользователя, а третья строка пуста (для будущей криптограммы). Далее я брал первую букву первой строки (текста пользователя), находил её в первой строке таблицы с русским алфавитом и записывал index j. Затем я брал первую букву второй строки (повторяющийся пароль), находил её в первом столбце таблицы с русским алфавитом и записывал index i. И наконец я находил в таблице с русским алфавитом букву с индексами (i, j), и записывал её в первый слот третьей строки (будущей криптограммы). Так для каждой буквы текста пользователя.



Реализация шифра Виженера

Далее я запустил два теста через командную строку. Один тест как в теории к лабораторной работе №2. Второй тест для дополнительной проверки. Шифрование совпало с тестом в лабораторной работе №2, и реализовано верно.



Проверка реализации шифра Виженера

# 4 Выводы

Я программно реализовал шифры перестановки.