Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

**Домашняя работа №2**

**Вариант № 4**

Выполнили:

Бакланова А.Г,

Крылов Д.С,

Улитина М.С

Проверил Мусаев А.А.

Санкт-Петербург, 2023

Оглавление

[Оглавление 2](#_Toc129529399)

[Введение. 2](#_Toc129529400)

[Алгоритмы выполнения заданий. 4](#_Toc129529401)

[Задание 1. 4](#_Toc129529402)

[Заключение. 6](#_Toc129529403)

# 

# Введение.

Цель лабораторной работы – выполнение задания с помощью Python. А именно создать программу, которая позволяет произвести хэширование. Задание выполняется с учетом варианта. Вариант работы: 4. Необходимо использовать два алгоритма: умножение, CRC32.

# 

# Алгоритмы выполнения заданий.

## Задание 1.

В первом задании нам необходимо реализовать хэширование. Сначала используем алгоритм умножения.

На рисунке 1 приведен наш код для выполнения задания. Первая строка просит пользователя ввести строку. В 3 строке с помощью split разбиваем строку на отдельные слова, каждое из которых будет хэшироваться. Задаем константу С из интервала от 0 до 1, необходимую для реализации, мы взяли С = 0,1. После пользователь вводит ключ. Создаем пустой массив. Реализуем цикл, где проходим по длине введенной строки, для каждого элемента (в нашем случае – слово) применяем формулу умножения и добавляем полученное значение в ранее созданным массив. Выводим полученный массив.

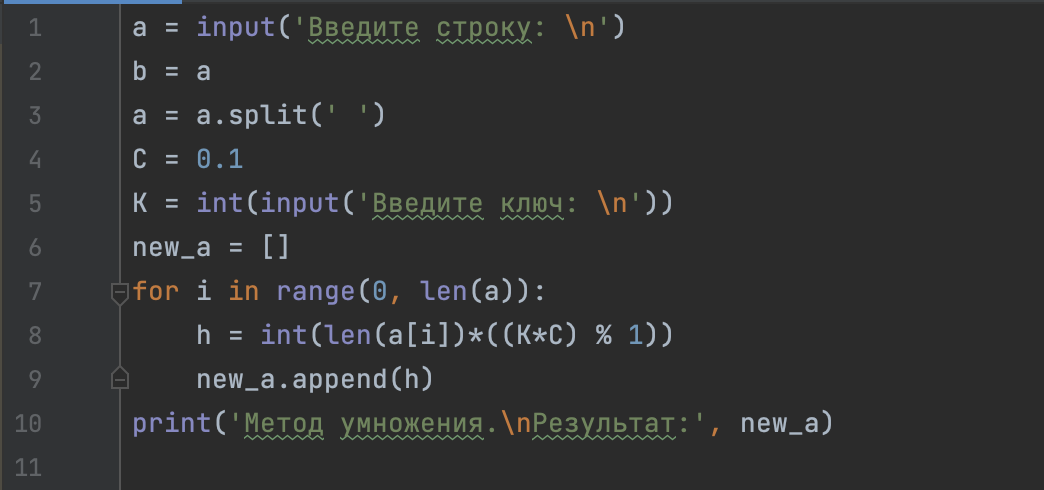


Рисунок 1 – Алгоритм умножения.

На рисунке 2 приведен алгоритм CRC32. Сначала присваиваем переменной порождающий многочлен. После переходим к реализации самого алгоритма: мы берем последовательность битов, далее представляем ее, как многочлен, в котором ест ь коэффициенты «0» и «1» (берем «0», если 0 в последовательности битов, берем «1», если 1 в последовательности битов) и члены со степенями от 0 до n, где n – длина введенной пользователем строки. Далее полученный многочлен умножается на порождающий многочлен (он также образован последовательностью битов).

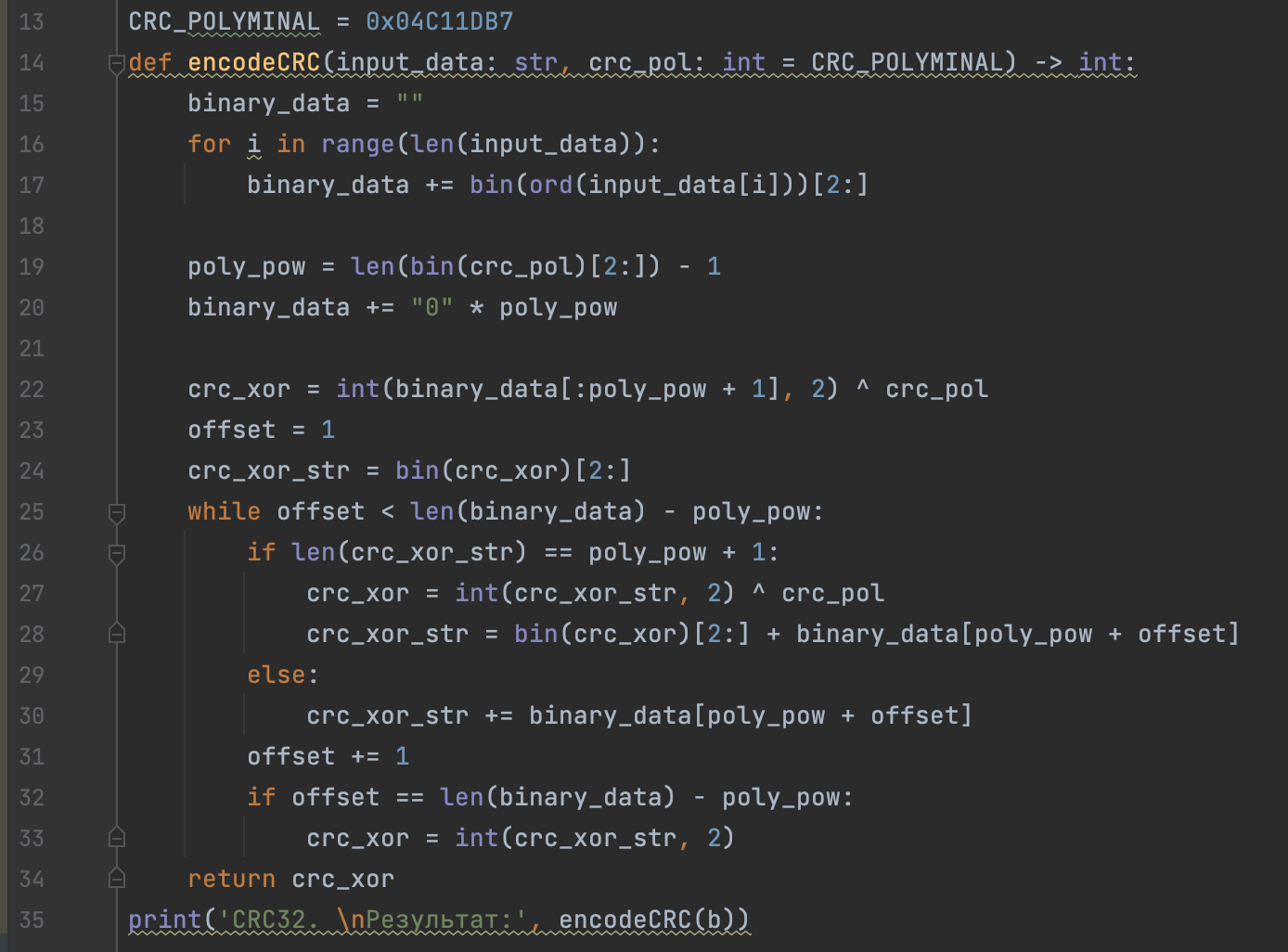


Рисунок 2 – Алгоритм CRC32.

Результат работы программы приведен на рисунке 3.

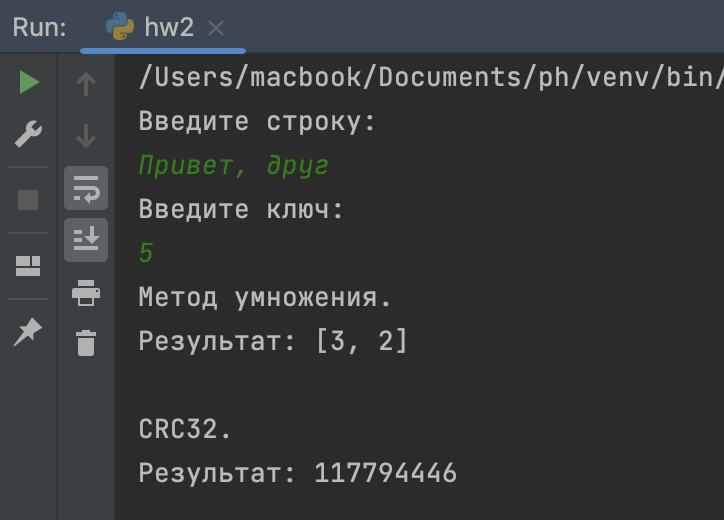


Рисунок 3 – Результат работы.

Таким образом, задание выполнено.

# Заключение.

Задание выполнено, таким образом, можно считать, что цель домашней работы выполнена. Создана программа, которая позволяют произвести хэширование. Задание выполнено с учетом варианта. Вариант работы: 4. Используются два алгоритма: умножения, CRC32. Программы находятся по ссылке: <https://github.com/Anasstasssia/Hw>