**ЛАБОРАТОРНАЯ РАБОТА №5**

**ВАРИАНТ 8**

**КРАЖЕВСКИЙ АЛЕКСЕЙ ИГОРЕВИЧ, 15 ГРУППА**

Условие варианта:



Для выполнения всех заданий необходимо использовать "длинную" арифметику. Разрешается использовать любую готовую библиотеку или написать свою.

Шаг 1. В условиях своего варианта (см. в таблице 1) для заданных чисел p, q и e, необходимо вычислить личный ключ d. Можно использовать либо расширенный алгоритм Евклида или малую теорему Ферма (частный случай теоремы Эйлера).

Шаг 2. Для заданного сообщения X1, вычислить зашифрованное сообщение Y1, используя открытый ключ e.

Шаг 3. Расшифровать сообщение Y1, используя найденный личный ключ d, сравнить результат с исходным сообщением X1.

Шаг 4. Для заданного шифртекста Y2, вычислить исходный открытый текст X2, используя личный ключ d.

**Описание RSA.**

RSA— криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Генерация ключей:

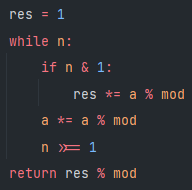
Публичный ключ e нам дан по условию. Для вычисления секретного ключа d нужно использовать формулу Т.е., нужно найти обратный элемент для e.

Шифрование происходит по формуле 

Расшифровка же по формуле 

**Алгоритм возведения в степень.**

Я использовал алгоритм бинарного возведения в степень, работающий за O(log n) и сразу считающий по модулю. Вот сам алгоритм:



Сам алгоритм основан на тождестве:

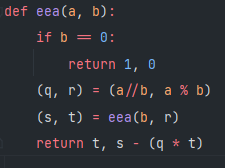


- это для четной степени

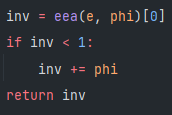
В случае нечетной степени просто переходим к четной

**Нахождение обратного элемента по модулю.**

Сам алгоритм использует расширенный алгоритм Евклида.

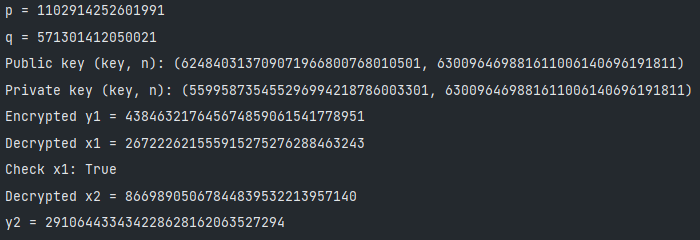


И после считает обратный элемент:



**Значения.**

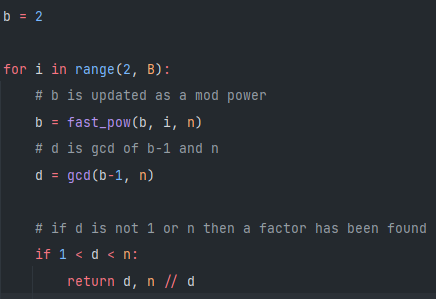
Значения, полученные из варианта и в ходе выполнения кода:



**Дополнительное задание – реализация метода факторизации числа**

Я реализовал p – 1 метод Полланда.

Код метода:



Верхняя граница вводится пользователем. На каждой итерации (от 2 до введенной границы) мы считаем b и d и проверяем на принадлежность промежутку от 1 до числа. Если условие выполняется, значит мы нашли простой делитель.

Результат выполнения для теста с числом 10001 и границей 10.

