МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФГБОУ ВО «Пермский государственный

национальный исследовательский университет»

Физический факультет

Кафедра радиоэлектроники

и защиты информации

ОТЧЁТ

По лабораторной работе «Алгоритм Сильвера-Полига-Хеллмана»

по дисциплине «Криптографические методы защиты информации»

Работы выполнил студент ФЗ-13

Заболотько Семён Ильич

**Постановка задачи**

1. Выбрать значения для логарифма(a, b , q)
2. Через Алгоритм Сильвера-Полига-Хеллмана вычислить степень x(задача дискретного логарифмирования)

**Краткая теория**

Алгоритм Полига — Хеллмана (также называемый алгоритм Сильвера — Полига — Хеллмана) — [детерминированный алгоритм](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) [дискретного логарифмирования](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B5_%D0%BB%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) в [кольце вычетов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D1%8C%D1%86%D0%BE_%D0%B2%D1%8B%D1%87%D0%B5%D1%82%D0%BE%D0%B2) по модулю [простого числа](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE). Одной из особенностей алгоритма является то, что для [простых](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) чисел специального вида можно находить дискретный логарифм за [полиномиальное](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_P) время.

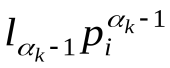
Данный алгоритм был придуман американским математиком [Роландом Сильвером](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B8%D0%BB%D1%8C%D0%B2%D0%B5%D1%80,_%D0%A0%D0%BE%D0%BB%D0%B0%D0%BD%D0%B4&action=edit&redlink=1) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Roland Silver), но впервые был опубликован другими двумя американскими математиками [Стивеном Полигом](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D0%BE%D0%BB%D0%B8%D0%B3,_%D0%A1%D1%82%D0%B8%D0%B2%D0%B5%D0%BD&action=edit&redlink=1) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [Stephen Pohlig](https://en.wikipedia.org/wiki/Stephen_Pohlig)) и [Мартином Хеллманом](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD,_%D0%9C%D0%B0%D1%80%D1%82%D0%B8%D0%BD) в [1978 году](https://ru.wikipedia.org/wiki/1978_%D0%B3%D0%BE%D0%B4) в статье «An improved algorithm for computing logarithms over GF(p) and its cryptographic significance», которые независимо от Роланда Сильвера разработали данный алгоритм.

**Применение:**

Алгоритм Полига—Хеллмана крайне эффективен, если p−1раскладывается на небольшие простые множители. Это очень важно учитывать при выборе параметров криптографических схем. Иначе схема будет ненадёжной.

**Идея Алгоритма:**

Этот алгоритм использует следующий подход: пусть G – группа порядка *n*, и *n=*https://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-0q4don.png- каноническое разложение на простые сомножители. Если*x=*log*ga* mod *n*, то, вычислив *xi=*log*ga*mod https://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-LwuSbY.png, для 1 *≤ i ≤ k*, можно восстановить *x*, используя китайскую теорему об остатках.

Для того чтобы вычислить *xi*, вычисляют коэффициенты *l*0, *l*1,…,https://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-57Ph1l.pngв*pi*-чном представлении числа *xi*: *xi=l*0+*l*1*pi*+…+, где 0 ≤*lj*≤ *pi—*1.

**Суть алгоритма:**

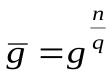
Вход: *g* – порождающий элемент циклической группы порядка *n*,*a*https://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-Oj0XSz.pngG.

Ш.1. Найти каноническое разложение *n=*https://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-qzcEus.png.

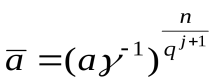
Ш.2. Для *i* от 1 до *k* выполнить следующие действия:

1. Задать *q=pi*, α=αi.

2. Задать γ=1, *l*-1=0.

3. Вычислить .

4. Для *j* от 1 до α—1 выполнить:

4.1. Вычислить γ=γhttps://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-X5Zncf.png,

4.2. Вычислить *li=*https://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-NzVXnY.png. (например, используя алгоритм «шаг младенца - шаг великана» или прямой поиск).

5. Вычислить *xi=l*0+*l*1*q*+…+*lα—*1*qα*—1.

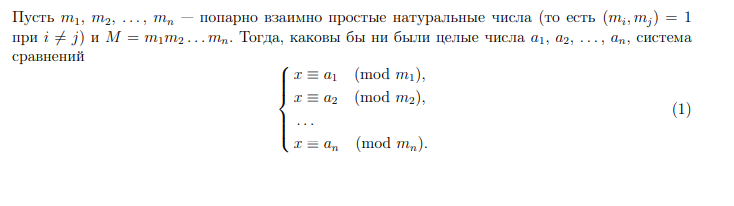
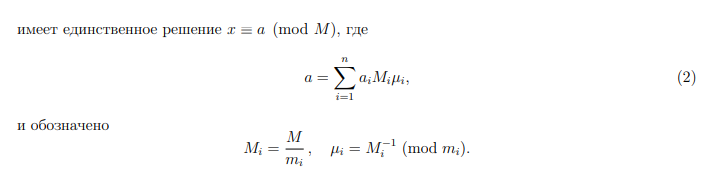
Ш.3. Используя Китайскую теорему об остатках, решить систему сравнений *x≡xi*(mod https://studfile.net/html/2706/30/html_ucnqUlA_3F.Qpzu/img-5UV4iB.png) .

Выход. *x=*log*ga* mod *n*.

**Китайская теорема об остатках:**

В [математике](https://en.wikipedia.org/wiki/Mathematics) китайская теорема об остатках гласит, что если известны остатки [евклидова деления](https://en.wikipedia.org/wiki/Euclidean_division) целого [числа](https://en.wikipedia.org/wiki/Integer)n на несколько целых чисел, то можно однозначно определить остаток от деления n на произведение этих целых чисел при условии, что [делители](https://en.wikipedia.org/wiki/Divisor) попарно взаимно [просты](https://en.wikipedia.org/wiki/Pairwise_coprime) (никакие два делителя не имеют общего делителя, кроме 1).

Например, если мы знаем, что остаток от деления n на 3 равен 2, остаток от деления n на 5 равен 3, а остаток от деления n на 7 равен 2, то, не зная значения n , мы можем определить, что остаток от n , разделенный на 105 (произведение 3, 5 и 7), равен 23. Важно отметить, что это говорит нам о том, что если n — [натуральное число](https://en.wikipedia.org/wiki/Natural_number) меньше 105, то 23 — единственное возможное значение n .

****

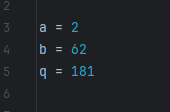
**Замечания**

Для применения алгоритма Полига-Хеллмана необходимо знать разложение p-1 на множители. В общем случае задача факторизации — достаточно трудоёмкая, однако если делители числа — небольшие (в том смысле, о котором сказано выше), то это число можно быстро разложить на множители даже методом последовательного деления. Таким образом, в том случае, когда эффективен алгоритм Полига-Хеллмана, необходимость факторизации не усложняет задачу.

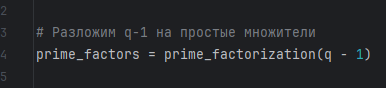
**Выполнение**

Наша задача – получить значение степени x при известных параметрах логарифма a, b, q

Возьмем a, b, q из примера с лекции

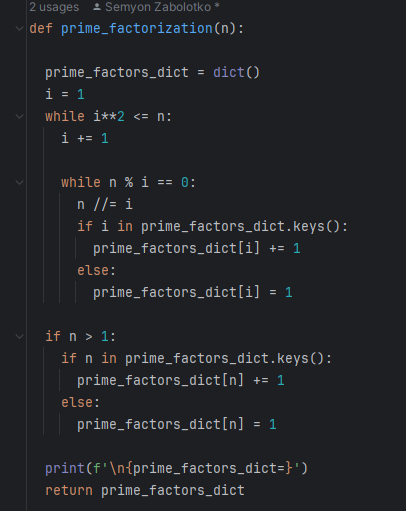
****

Первым делом Разложим q-1 на простые множители

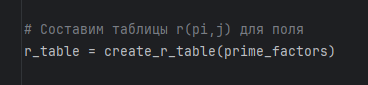
****

Представим простые множители в виде словаря(основание: степень)

Далее с помощью факторизации простых чисел(взято из интернета) разложим число и пополним словарь

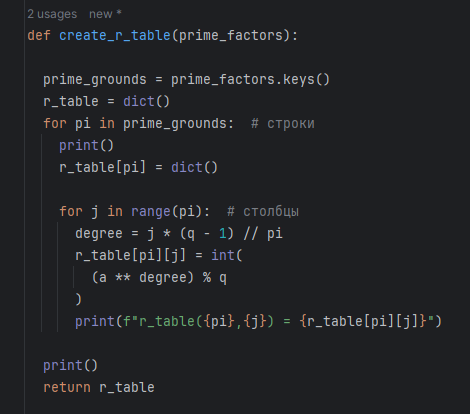
****

Теперь составим таблицу, но для удобства она будет реализована не в виде матрицы, а в виде словаря, так как некоторые значения в ней должны быть пустыми(X)

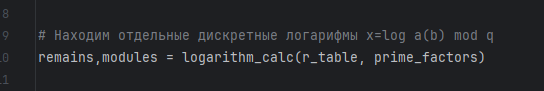
****

По порядку вычисляем каждую строку отдельно(все значения для каждого основания простого множителя) – сначала показатель степени, а потом a в этой степени по модулю

Значения в таблице вычислены при помощи стандартной формулы из книги Сонг Яна(132стр)

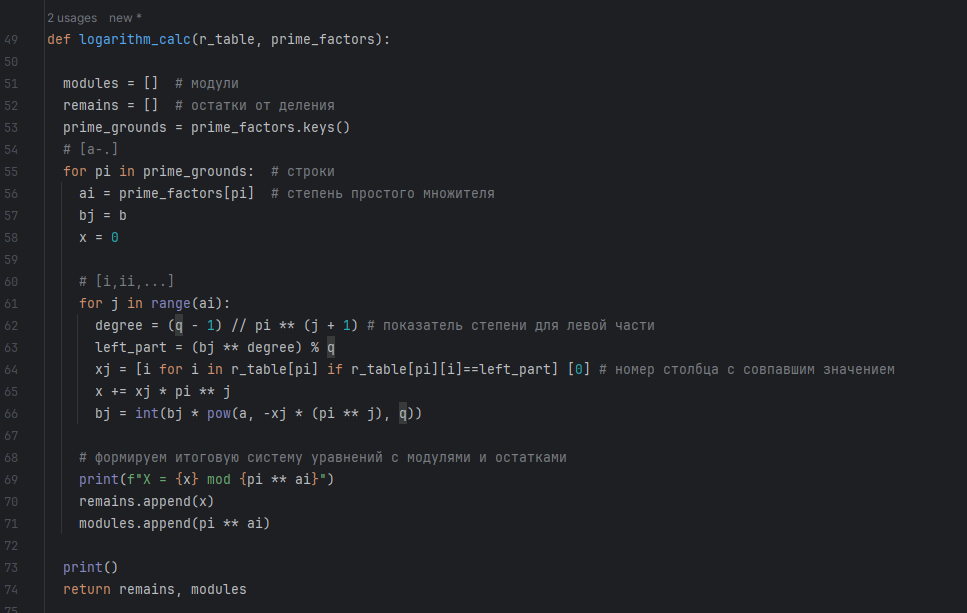
****

После вычисления таблицы можем приступить к нахождению отдельных логарифмов, используюя таблицу и множители

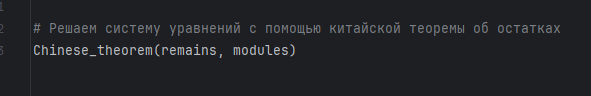
****

Итогом вычисления наших логарифмов будет набор из остатков от деления и модулей, которые мы будем использовать в системе уравнений.

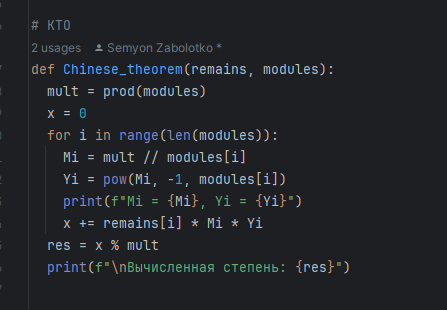
В цикле мы рассчитываем так называемую левую часть – на каждом элементе таблицы у нас получится xj, который мы добавляем в общую сумму x.

****

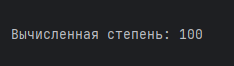
Теперь нам осталось решить систему уравнений с помощью КТО

****

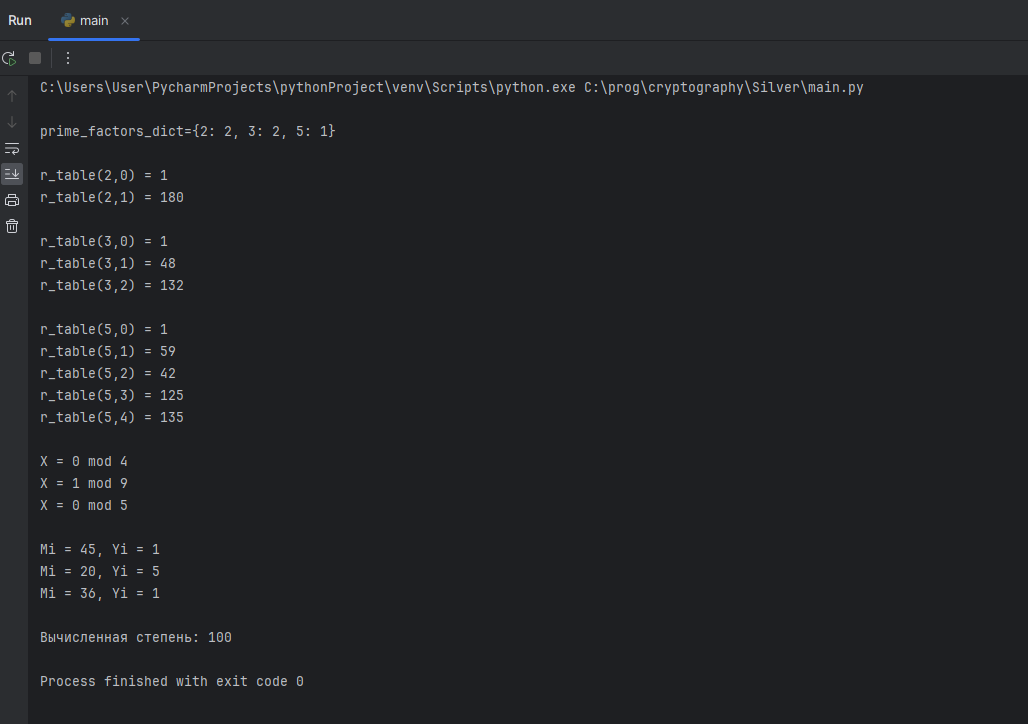
Сначала вычисляется произведение всех модулей и далее в цикле оно делится нацело на каждый из модулей. От полученного значения Mi берётся обратное по модулю число. Далее соответствующий остаток умножается на Mi и на его обратное. Полученное значение добавляется в общую сумму и после прохождения всех уравнений делится с остатком на общее произведение.

****

В итоге получается степень логарифма

****

Полный консольный вывод

****

**Выводы**

Мы научились применять алгоритм Сильвера-Полига-Хеллмана для вычисления степени x при известных параметрах a, b, q(задача дискретного логарифмирования)