Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Криптографические протоколы**

Работу выполнила: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д. Н. Баева

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А. А. Крамаренко

**Цель работы:** реализовать программный продукт решения квадратичного сравнения с указанием промежуточных результатов. Программа должна показывать случаи, когда решения существуют, находить количество решений или указывать, что нахождение затруднительно.

**Ход работы:**

Сравнение с одним неизвестным *x* имеет вид ,где *m∈ N, m>1*. При *n = 2* сравнение называется квадратичным сравнением.Заметим, что сравнение такого вида можно упростить. Для этого положим   и решим сравнение первой степени   относительно t. Получим сравнение вида  , или, что то же самое, что .

Пусть – простое число и – целое число, взаимно простое с . Число *a* называется квадратичным вычетом по модулю *p*, если существует такое, что В противном случае число *a* называется квадратичным невычетом по модулю *p*.

Для определения количества решений квадратичных сравнений используется символ Лежандра.

Символом Лежандра называется выражение, обозначаемое ( ), равное 1, если *a* — квадратичный вычет по модулю *p*; −1, если *a* — невычет по модулю *p* и 0, если *a* кратно *p*. Формула:( ) ≡ (mod p).

Для квадратичных сравнений возможны следующее ситуации:

1. модуль является простым числом;
2. модуль является составным – простое число в степени;
3. модуль является составным – возможно его разложение на простые множители.

Реализация состоит из применения следующих функций. Сперва функция is\_prime(number) проверяет, является ли число простым. Она делит число на все числа от 2 до квадратного корня из числа. Если число делится без остатка, то оно не является простым. Далее метод factorize(n): разлагает число *n* на простые множители. Она делит *n* на все числа, начиная с 2, пока оно не станет равным 1. Все делители добавляются в список factors. Соответствующие функции представлены на рисунке 1.

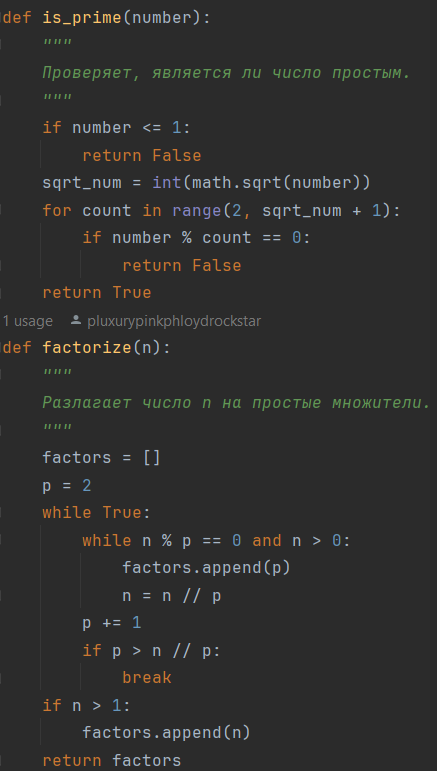


Рисунок 1 – Проверка на простоту числа и разложение на простые множители.

После вычисляется символ Лежандра. Для этого используется функция legendre\_symbol(a, p). Символ Лежандра определяет, является ли число *a* квадратичным вычетом по модулю простого числа *p*. Если *a* делится на *p* без остатка, символ Лежандра равен 0. Если *a* в степени *(p - 1) / 2* по модулю *p* равно 1, символ Лежандра равен 1. В противном случае символ Лежандра равен -1. Функция, вычисляющая данное значение, представлена на рисунке 2.

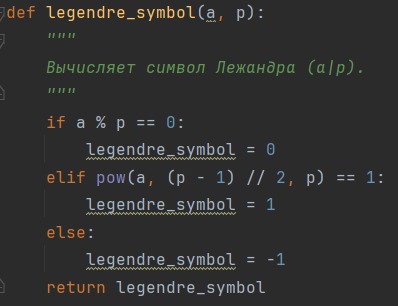


Рисунок 2 – Символ Лежандра.

Решение квадратичного сравнения происходит в функции solve\_quadratic\_congruence(a, b, m). Она проверяет различные случаи в зависимости от значения модуля (простое число, степень простого числа, составное число) и вычисляет решения сравнения. Если символ Лежандра равен 1, то есть *a* является квадратичным вычетом по модулю *m*, функция ищет решения следующим образом: 1) если модуль – простое число и больше 2, то функция вычисляет квадратный корень из *a* по модулю *m* и использует его для нахождения двух решений; 2) если модуль – степень простого числа, функция перебирает все числа от 0 до *m* и проверяет, являются ли они решениями сравнения; 3) если модуль – составное число, которое раскладывается на простые множители, функция вновь выполняет перебор всех значений и проверяет, являются ли они решениями сравнения.

После определения всех функций код запрашивает у пользователя ввод необходимых коэффициентов, а затем вызывает функцию solve\_quadratic\_congruence(a, b, m) для получения решений. Если решения существуют, они выводятся на экран. Если решений нет, выводится соответствующее сообщение.

**Листинг программы**

Файл LW\_2.py

# решение квадратичных сравнений вида ax^2 ≡ b (mod m)

import math

def is\_prime(number):

"""

Проверяет, является ли число простым.

"""

if number <= 1:

return False

sqrt\_num = int(math.sqrt(number))

for count in range(2, sqrt\_num + 1):

if number % count == 0:

return False

return True

def factorize(n):

"""

Разлагает число n на простые множители.

"""

factors = []

p = 2

while True:

while n % p == 0 and n > 0:

factors.append(p)

n = n // p

p += 1

if p > n // p:

break

if n > 1:

factors.append(n)

return factors

def legendre\_symbol(a, p):

"""

Вычисляет символ Лежандра (a|p).

"""

if a % p == 0:

legendre\_symbol = 0

elif pow(a, (p - 1) // 2, p) == 1:

legendre\_symbol = 1

else:

legendre\_symbol = -1

return legendre\_symbol

def solve\_quadratic\_congruence(a, b, m):

"""

Решает квадратичное сравнение ax^2 ≡ b (mod m).

:param a: Коэффициент a

:param b: Коэффициент b

:param m: Модуль m

:return: Список решений (если они есть)

"""

# список решений

solutions = []

# вычисление символа Лежандра

legendre = legendre\_symbol(a, m)

if legendre == 1:

if is\_prime(m):

print("Сравнение с простым модулем")

# случай, когда m - простое число

if m > 2:

# a имеет квадратный корень по модулю m

sqrt\_a = pow(a, (m + 1) // 4, m)

x1 = (sqrt\_a \* b) % m

x2 = m - x1

solutions.append(x1)

solutions.append(x2)

return solutions

else:

return solutions

# случай, когда m - степень простого числа:

elif not is\_prime(m) and len(factorize(m)) == 1:

print("Сравнение с составным модулем - степень простого числа")

for x in range(m):

if (a \* x\*\*2) % m == b:

solutions.append(x)

return solutions

# случай, когда m - составное число, которое раскладывается на простые множители

else:

print("Сравнение с составным модулем - множители простого числа")

for x in range(m):

if (a \* x \*\* 2) % m == b:

solutions.append(x)

return solutions

else:

return solutions

a = int(input("Введите коэффициент а: "))

b = int(input("Введите коэффициент b: "))

m = int(input("Введите коэффициент m: "))

# получение списка решений

solutions = solve\_quadratic\_congruence(a, b, m)

if solutions :

print(f"Решения сравнения {a}\*x^2 ≡ {b} (mod {m}): {solutions}")

else:

print(f"Сравнение {a}\*x^2 ≡ {b} (mod {m}) не имеет решений.")