МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КУБГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчет**

**по практическому заданию №1**

**по курсу**

**«КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ»**

Работу выполнила

Студентка 49 группы

Шестак В. А.

Преподаватель:

Крамаренко А.А.

Краснодар 2025

**Цель работы:** реализовать программный продукт нахождения квадратичных вычетов и множества квадратичных невычетов по заданному простому модулю с пояснением всех промежуточных шагов решения задачи.

**Теория:**

Квадратичный вычет — это такое целое число a, для которого существует такое целое число x, что , где p – некоторое простое число. Другими словами, а является квадратом целого числа в модульной арифметике по модулю p.

Например, если то множество квадратичных вычето будет состоять из чисел, которые могут быть представлены в виде Таким образом, квадратичные вычеты для p=7 будут: 1,2,4.

Основное свойство квадратичных вычетов заключается в том, что они имеют квадратные корни, то есть для каждого квадратичного вычета a найдется целое число x, такое что .

Квадратичные вычеты играют важную роль в различных областях математики и криптографии, включая криптографию на основе алгоритма RSA и различные методы шифрования.

Символ Лежандра играет важную роль в определении квадратичных вычетов и невычетов по модулю p. Он определяется следующим образом:

Символ Лежандра является расширением символа Эйлера () и позволяет определить, является ли число a квадратичным вычетом или невычетом.

Таким образом, мы можем использовать символ Лежандра для классификации чисел как квадратичные вычеты или невычеты по модулю p.

**Ход работы;**

1. Функция extended\_euclid(a, b)

Эта функция реализует расширенный алгоритм Евклида. Она возвращает:

* НОД gcd(a,b),
* Коэффициенты xx и yy, такие что a⋅x+b⋅y=gcd(a,b)

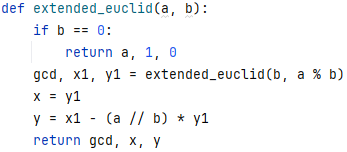


Рисунок 1 – Функция расширенного алгоритма Евклида

Пример: Для a=15, b=10:

gcd(15,10)=5,

x=−1, y=2, так как 15⋅(−1)+10⋅2=5.

2. Функция diophantine\_solution(a, b, c)

Эта функция решает линейное диофантово уравнение a⋅x+b⋅y=c.

**Шаги:**

Находит gcd(a,b) и коэффициенты x0​, y0​ с помощью extended\_euclid.

Проверяет, делится ли c на gcd(a,b). Если нет, то решений нет.

Если решение существует, масштабирует x0​ и y0​ на ​.

Находит общее решение для k в диапазоне от -5 до 5.

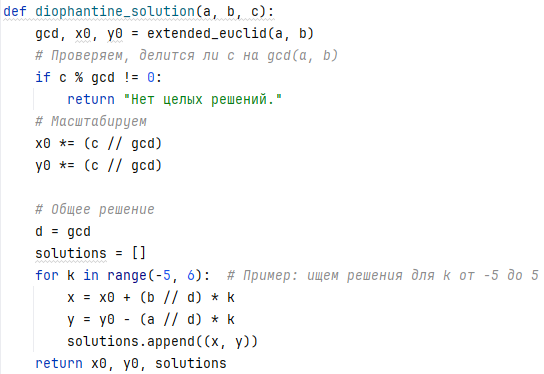


Рисунок 2 – Диофантовое уравнение

Пример:  
Для a=15, b=10, c=5:

gcd(15,10)=5,

x0=−1, y0​=2,

Общее решение:

x=−1+2k, y=2−3k.

3. Пример использования

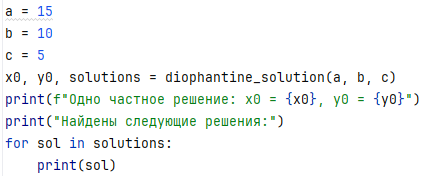


Рисунок 3 – Пример использования

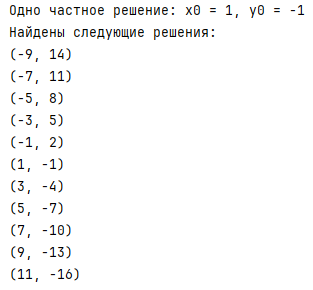
****

Рисунок 4 – Вывод программы

**Листинг программы:**

def legendre\_symbol(a, p):  
 *"""  
 Вычисляет символ Лежандра a/p  
 """* print("Вычисляем символ Лежандра для числа", a/p)  
 ls = pow(a, (p - 1) // 2, p)  
  
 result = ls if ls != p - 1 else -1  
 print("Символ Лежандра для числа", a/p, "равен", result)  
  
 return result  
  
def quadratic\_residues\_or\_non(p):  
 *"""  
 Находит множество квадратичных вычетов/невычетов по модулю p  
 """* residues = []  
 non\_residues = []  
 for a in range(1, p):  
 print("------")  
 print("Проверяем цифру", a)  
 if legendre\_symbol(a, p) == 1:  
 residues.append(a)  
 print("Цифра", a, "является квадратичным вычетом")  
 else:  
 non\_residues.append(a)  
 print("Цифра", a, "является квадратичным невычетом")  
 return residues, non\_residues  
  
p = 7  
residues, non\_residues = quadratic\_residues\_or\_non(p)  
print("-------")  
print("Множество квадратичных вычетов по модулю", p, ":", residues)  
print("Множество квадратичных невычетов по модулю", p, ":", non\_residues)