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 выполнил

Студент 49 группы

Пинский Д. А.

Преподаватель:

Крамаренко А.А.

Краснодар 2025

**Цель работы:** Реализовать программный продукт решения сравнений первой степени с указанием всех промежуточных шагов вычисления (текущее значение коэффициентов в расширенном алгоритме Евклида), программный продукт так же должен реализовывать возможность того, что сравнение не имеет решений или имеет больше одного решения. В первом случае сообщать пользователю с пояснением, во втором строить все возможные решения.

**Теория:**

Сравнения первой степени имеют вид ax ≡ b mod m, где a, b, и m — целые числа, а x — неизвестное.

Расширенный алгоритм Евклида используется для нахождения НОД(a, m) и коэффициентов x и y таких, что ax + my = НОД(a, m).

Существование и единственность решений:

- Если b не делится на НОД(a, m), то уравнение не имеет решений.

- Если b делится на НОД(a, m), то существует бесконечно много решений вида x0 + (m/НОД(a, m))k, где x0 — частное решение, а k — любое целое число.

**Ход работы:**

1. Расширенный алгоритм Евклида: Находит НОД(a, m) и коэффициенты x, y.

2. Проверка на существование решений: Если b не делится на НОД(a, m), выводит сообщение об отсутствии решений.

3. Строит все возможные решения: Если b делится на НОД(a, m), находит частное решение x0 и выводит все решения вида x0 + (m/НОД(a, m))k.

**Пример:**

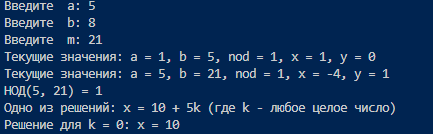


Рисунок 1 – пример работы программы

Листинг программы:

def euclid(a, b):

    if a == 0:

        return b, 0, 1

    nod, x1, y1 = euclid(b % a, a)

    x = y1 - (b // a) \* x1

    y = x1

    print(f"Текущие значения: a = {a}, b = {b}, nod = {nod}, x = {x}, y = {y}")

    return nod, x, y

def degree\_comparison(a, b, m):

    nod, x, y = euclid(a, m)

    print(f"НОД({a}, {m}) = {nod}")

    if b % nod != 0:

        print(f"Уравнение не имеет решений. НОД({a}, {m}) = {nod}, а {b} не делится на {nod},"

              " значит, уравнение не имеет решений в целых числах.")

        return

    x0 = (x \* (b // nod)) % (m // nod)

    a //= nod

    b //= nod

    m //= nod

    print(f"Одно из решений: x = {x0} + {a}k (где k - любое целое число)")

    for k in range(nod):

        total = (x0 + k \* (m // nod)) % (m // nod)

        print(f"Решение для k = {k}: x = {total}")

if \_\_name\_\_ == "\_\_main\_\_":

    a = int(input("Введите  a: "))

    b = int(input("Введите  b: "))

    m = int(input("Введите  m: "))

    degree\_comparison(a, b, m)