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 решения диофантового уравнения первой степени с помощью расширенного алгоритма Евклида с указанием всех промежуточных результатов.

**Теория:**

1. Алгоритм Евклида

Алгоритм Евклида используется для нахождения наибольшего общего делителя (НОД) двух чисел. Он основан на следующем свойстве:

gcd(a,b)=gcd(b,a mod  b)

Алгоритм продолжает рекурсивно применять это свойство, пока b не станет равным нулю. В этом случае a будет НОД.

2. Расширенный алгоритм Евклида

Расширенный алгоритм Евклида не только находит НОД двух чисел a и b, но и находит такие целые числа x и y, что:

a⋅x+b⋅y=gcd(a,b)

Это уравнение называется линейным диофантовым уравнением.

3. Линейные диофантовы уравнения

Линейное диофантово уравнение имеет вид:

a⋅x+b⋅y=c,

где a, b, c — целые числа. Уравнение имеет целые решения (x,y) тогда и только тогда, когда c делится на gcd(a,b).

Если (x0,y0)— частное решение, то общее решение можно записать в виде:

где k — произвольное целое число.

.

**Ход работы;**

1. Функция extended\_euclid(a, b)

Эта функция реализует расширенный алгоритм Евклида. Она возвращает:

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

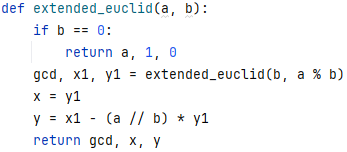


Рисунок 1 – Функция расширенного алгоритма Евклида

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.

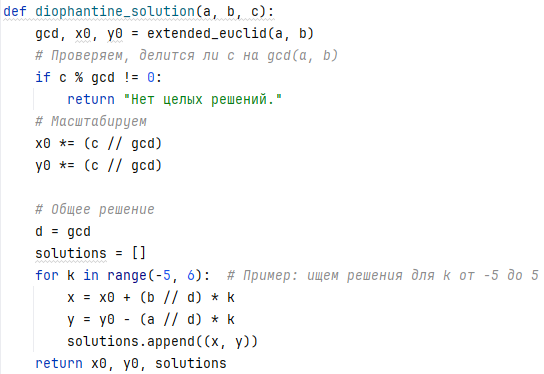


Рисунок 2 – Диофантовое уравнение

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

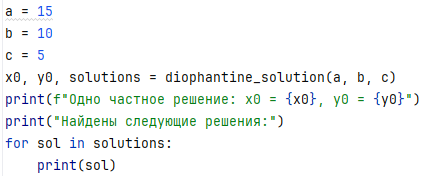


Рисунок 3 – Пример использования

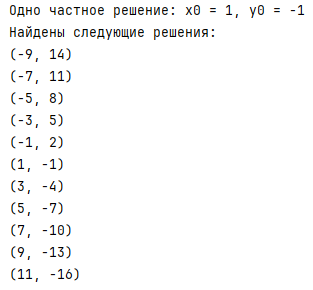
****

Рисунок 4 – Вывод программы

**Листинг программы:**

def extended\_euclid(a, b):  
 if b == 0:  
 return a, 1, 0  
 gcd, x1, y1 = extended\_euclid(b, a % b)  
 x = y1  
 y = x1 - (a // b) \* y1  
 return gcd, x, y  
def diophantine\_solution(a, b, c):  
 gcd, x0, y0 = extended\_euclid(a, b)  
  
 *# Проверяем, делится ли c на gcd(a, b)* if c % gcd != 0:  
 return "Нет целых решений."  
 *# Масштабируем* x0 \*= (c // gcd)  
 y0 \*= (c // gcd)  
  
 *# Общее решение* d = gcd  
 solutions = []  
 for k in range(-5, 6): *# Пример: ищем решения для k от -5 до 5* x = x0 + (b // d) \* k  
 y = y0 - (a // d) \* k  
 solutions.append((x, y))  
  
 return x0, y0, solutions  
*# Пример использования*a = 15  
b = 10  
c = 5  
x0, y0, solutions = diophantine\_solution(a, b, c)  
print(f"Одно частное решение: x0 = {x0}, y0 = {y0}")  
print("Найдены следующие решения:")  
for sol in solutions:  
 print(sol)