

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ
компьютерной безопасности и
криптографии

Решений сравнений с помощью теоремы Эйлера

ЛАБОРАТОРНАЯ РАБОТА

студента 4 курса 431 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Серебрякова Алексея Владимировича

Научный руководитель

доцент, к. п. н.

подпись, дата

А. С. Гераськин

Саратов 2022

Описание алгоритма

Решение с помощью теоремы Эйлера

$$ax \equiv b \pmod{m}; (a, m) = 1, a^{\phi(m)} \equiv 1 \pmod{m}$$

$$x = a^{\phi(m)-1} \cdot b - \text{решение}$$

$$a \cdot a^{\phi(m)-1} \cdot b \equiv b \pmod{m}$$

Код программы

```
#include <bits/stdc++.h>
#include <iostream>

using namespace std;

int validated_input()
{
    int s = 0;
    while (!(cin >> s))
    {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        printf("! Неверный ввод. Повторите ввод, начиная с первого неверного элемента.\n");
    }

    return s;
}

int phi(int n)
{
    int result = n;
    for (int i = 2; i * i <= n; ++i)
        if (n % i == 0)
        {
            while (n % i == 0)
                n /= i;
            result -= result / i;
        }
    if (n > 1)
        result -= result / n;
    return result;
}

int modexp(int x, int y, int N)
{
    if (y == 0)
        return 1;

    int z = modexp(x, y / 2, N);

    if (y % 2 == 0)
        return (z * z) % N;
    else
        return (x * z * z) % N;
}
```

```

}

int norm(int x, int m)
{
    while (x < 0)
        x += m;

    return x % m;
}

int gcd(int a, int b)
{
    return b ? gcd(b, a % b) : a;
}

void program(int a, int b, int m)
{
    printf("\nИсходное уравнение:\n %d * x = %d (mod %d)\n", a, b, m);

    a = norm(a, m);
    b = norm(b, m);

    int d = gcd(a, m);
    printf("\nНОД(%d, %d) = %d\n", a, b, d);

    if (b % d == 0)
    {
        a /= d;
        b /= d;
        m /= d;
    }

    printf("\nСокращенное уравнение:\n %d * x = %d (mod %d)\n", a, b, m);

    printf("\nВычислим phi:\n  phi(%d) = %d\n", m, phi(m));

    int first = norm(b * modexp(a, phi(m) - 1, m), m);
    printf("\nРешение:\n  x0 = %d", first);

    for (int i = 1; i < d; i++)
        printf("\n  x%d = %d", i, first + m * i);

    printf("\nКонец работы программы\n");
}

int main()
{
    setlocale(0, "");

    int a, b, m;

    printf("\nРешение уравнения вида\n ax = b (mod m)\n");
    printf("можно свести к нахождению функции Эйлера, вычисляющей количество взаимно простых чисел с заданным числом,");
    printf("\na затем найти одно из решений искомого выражения по формуле\n x0 = b * a^(phi(m)-1) (mod m)\n");

```

```

printf("\nОстальные решения можно найти по формуле\n x = x0 + mk,  k = 1..НОД(a,m)\n");

printf("\nВведите коэффициент a: ");
a = validated_input();

printf("\nВведите коэффициент b: ");
b = validated_input();

printf("\nВведите коэффициент m: ");
m = validated_input();

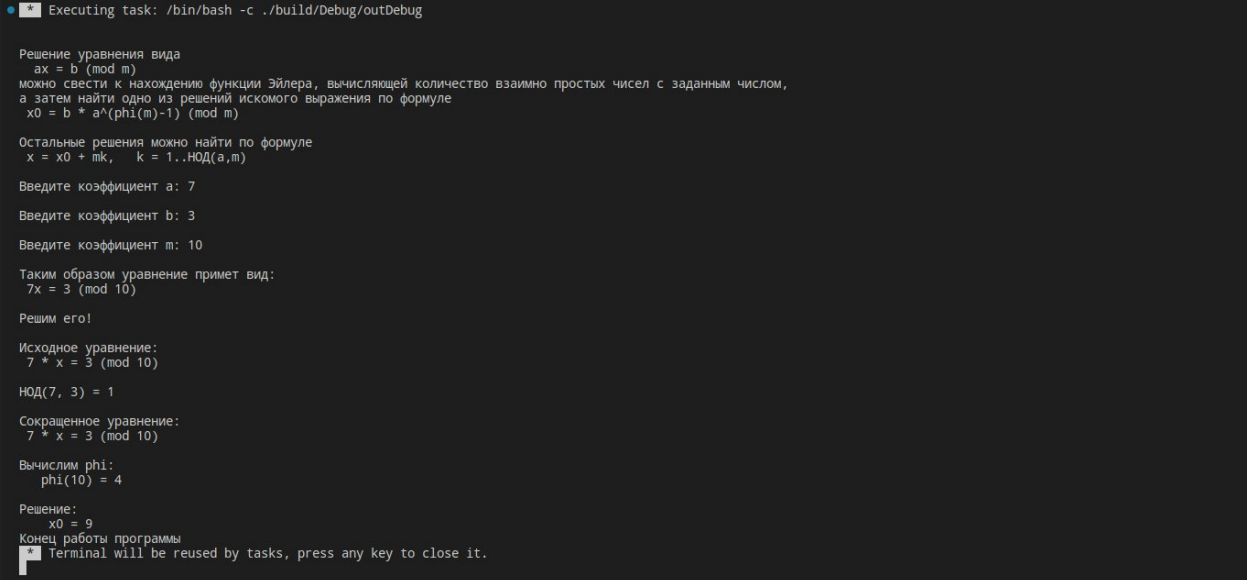
printf("\nТаким образом уравнение примет вид:\n %dx = %d (mod %d)\n", a, b, m);
printf("\nРешим его!\n");

program(a, b, m);

return 0;
}

```

Пример запуска программы



```

Executing task: /bin/bash -c ./build/Debug/outDebug

Решение уравнения вида
ax = b (mod m)
можно свести к нахождению функции Эйлера, вычисляющей количество взаимно простых чисел с заданным числом,
а затем найти одно из решений искомого выражения по формуле
x0 = b * a^(phi(m)-1) (mod m)

Остальные решения можно найти по формуле
x = x0 + mk,  k = 1..НОД(a,m)

Введите коэффициент a: 7
Введите коэффициент b: 3
Введите коэффициент m: 10

Таким образом уравнение примет вид:
7x = 3 (mod 10)

Решим его!

Исходное уравнение:
7 * x = 3 (mod 10)

НОД(7, 3) = 1

Сокращенное уравнение:
7 * x = 3 (mod 10)

Вычислим phi:
phi(10) = 4

Решение:
x0 = 9
Конец работы программы
Terminal will be reused by tasks, press any key to close it.

```

Executing task: /bin/bash -c ./build/Debug/outDebug

Решение уравнения вида

$$ax \equiv b \pmod{m}$$

можно свести к нахождению функции Эйлера, вычисляющей количество взаимно простых чисел с заданным числом, а затем найти одно из решений искомого выражения по формуле

$$x_0 \equiv b * a^{\phi(m)-1} \pmod{m}$$

Остальные решения можно найти по формуле

$$x = x_0 + mk, \quad k = 1..НОД(a,m)$$

Введите коэффициент a: 17

Введите коэффициент b: 4

Введите коэффициент m: 24

Таким образом уравнение примет вид:

$$17x \equiv 4 \pmod{24}$$

Решим его!

Исходное уравнение:

$$17 * x \equiv 4 \pmod{24}$$

$$НОД(17, 4) = 1$$

Сокращенное уравнение:

$$17 * x \equiv 4 \pmod{24}$$

Вычислим phi:

$$\phi(24) = 8$$

Решение:

$$x_0 = 20$$

Конец работы программы

* Terminal will be reused by tasks, press any key to close it.