РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Тема: Вычисление наибольшего общего делителя

дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Койфман Кирилл Дмитриевич

Группа: НФИмд-01-25

Введение

Цель работы

Получение практических навыков реализации алгоритмов, проверяющих числа на простоту.

Задачи

1. Реализовать алгоритмы проверки чисел на простоту.

Ход работы

Для решения поставленной задачи реализуем описанные в тексте лабораторной работы алгоритмы для проверки чисел на простоту на языке программирования C++ (Листинг-1):

```
bool FermTest(int n)
    {
        if (n < 5)
            return false;
        if (n % 2 == 0)
            return false;

        //take random number in [2, n - 2]
        int a = generateRandomInteger(2, n - 2);

        int gamma = static_cast<int>(std::pow(a, n - 1)) % n;

        return gamma == 1 ? true : false;
    }

    int JacobiSymbolAlgo(int n, int a)
    {
        int g = 1;

        while (true)
        {
            if (a == 0)
                return 0;
            if (a == 1)
                return g;

            int k = 0;
            int a1 = a;
            while (a1 % 2 == 0)
            {
                a1 /= 2;
                k++;
            }

            int s = 1;
            if (k % 2 != 0)
            {
                if ((n - 1) % 8 == 0 || (n + 1) % 8 == 0)
                {
                    s = 1;
                }
                else if ((n - 3) % 8 == 0 || (n + 3) % 8 == 0)
                {
                    s = -1;
                }
            }

            if (a1 == 1)
                return g * s;

            if ((n - 3) % 4 == 0 && (a1 - 4) % 3 == 0)
            {
                s = -s;
```

```
            }

            a = n % a1;
            n = a1;
            g = g * s;
        }
    }

    bool SolovayShtrassenTest(int n)
    {
        if (n < 5)
            return false;
        if (n % 2 == 0)
            return false;

        int a = generateRandomInteger(2, n - 2);

        int gamma = static_cast<int>(std::pow(a, (n - 1) / 2.0)) % n;

        if (gamma != 1 && gamma != n - 1)
            return false;

        int s = JacobiSymbolAlgo(n, a);

        if ((gamma - s) % n == 0)
            return false;
        else
            return true;
    }
```

Листинг-1(реализация расширенного бинарного алгоритма Евклида)

Заключение

В ходе проделанной лабораторной работы мной были получены навыки по реализации алгоритмов, осуществляющих проверку чисел на простоту.