

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

Проверка чисел на простоту с помощью теста Соловея-Штрассена

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

3. Тест Соловей–Штрассена

Теорема 3. Пусть n — нечетное. Тогда для того, чтобы n было простым необходимо и достаточно, чтобы для каждого $a \in \mathbb{Z}_n^*$ было выполнено $a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \pmod{n}$.

4. Алгоритм Соловей-Штрассена

Вход: n — нечетное число

Выход: Сообщение о простоте или возможной простоте числа

```
1: for  $i = 1, \dots, k$  do
2:    $a := \text{rand}(2, \dots, n - 1)$ ;
3:   if  $\text{НОД}(a, n) > 1$  then
4:     PRINT 'n — составное';
5:     RETURN;
6:   if  $a^{\frac{n-1}{2}} \not\equiv \frac{a}{n}$  then
7:     PRINT 'n — составное';
8:     RETURN;
9: PRINT 'n — вероятно простое'
```

Вероятность ошибки, если выдан ответ 'вероятно простое', не превосходит $\left(\frac{1}{2}\right)^k$

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 gcd(int a, int b)
{
    int temp;

    while (b != 0)
    {
        temp = a;
        a = b;
        b = temp % b;
    }

    return a;
}

int modexp(int x, int y, int N)
{
    int result(1);

    for (int i = 0; i < y; i++)
```

```

    {
        result *= x;
        result %= N;
    }

    return result;
}

int jacobi(int a, int b)
{
    if (gcd(a, b) != 1)
        return 0;

    else
    {
        int r = 1;

        if (a < 0)
        {
            a = -a;

            if (b % 4 == 3)
                r = -r;
        }

        for (int i = 0; a != 0; i++)
        {
            int t = 0;

            while (a % 2 == 0)
            {
                t += 1;
                a /= 2;
            }

            if (t % 2 == 1 && (b % 8 == 3 || b % 8 == 5))
                r = -r;

            if (a % 4 == 3 && b % 4 == 3)
                r = -r;

            int c = a;

            a = b % c;
            b = c;
        }

        return r;
    }
}

int main()
{
    srand(time(0));
    setlocale(0, "");
    int n, k, a;

    printf("\nВведите число n: ");
    n = validated_input();

```

```

printf("\nПроверим число %d на простоту по тесту Соловей-Штрассена:", n);
printf("\n Если n нечетное, то оно простое <=> для всех a из Z*n выполняется  $a^{((n-1)/2)} = (a/n) \pmod n$ ");

printf("\nВведите количество проверок k: ");
k = validated_input();

for (int i = 0; i < k; i++)
{
    a = (rand() % (n - 2)) + 2;

    printf("\n[a = %4d] - свидетель простоты числа %d", a, n);

    if (gcd(n, a) > 1)
    {
        printf("\n[gdc]Число %d - составное\n\n", n);
        return 0;
    }

    int aa(modexp(a, (n - 1) / 2, n)), bb((n + jacobi(a, n)) % n);

    if (aa != bb)
    {
        printf("\n[modexp]Число %d - составное\n\n", n);
        return 0;
    }
}

printf("\nЧисло %d - Вероятно простое\n\n", n);

return 0;
}

```

Пример запуска программы

```

Введите число n: 1249

Проверим число 1249 на простоту по тесту Соловей-Штрассена:
Если n нечетное, то оно простое <=> для всех a из Z*n выполняется  $a^{((n-1)/2)} = (a/n) \pmod n$ 

Введите количество проверок k: 10

[a = 646] - свидетель простоты числа 1249
[a = 56] - свидетель простоты числа 1249
[a = 141] - свидетель простоты числа 1249
[a = 86] - свидетель простоты числа 1249
[a = 358] - свидетель простоты числа 1249
[a = 1248] - свидетель простоты числа 1249
[a = 487] - свидетель простоты числа 1249
[a = 924] - свидетель простоты числа 1249
[a = 992] - свидетель простоты числа 1249
[a = 165] - свидетель простоты числа 1249
Число 1249 - Вероятно простое

* Terminal will be reused by tasks, press any key to close it.
* Executing task: /bin/bash -c ./build/Debug/outDebug

Введите число n: 123132164

Проверим число 123132164 на простоту по тесту Соловей-Штрассена:
Если n нечетное, то оно простое <=> для всех a из Z*n выполняется  $a^{((n-1)/2)} = (a/n) \pmod n$ 

Введите количество проверок k: 10

[a = 103166258] - свидетель простоты числа 123132164
[gdc]Число 123132164 - составное

```

