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 Рабина-Миллера

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. _____ А. С. Гераськин

подпись, дата

Саратов 2021

Описание алгоритма

Как и тесты Ферма и Соловея — Штрассена, тест Миллера — Рабина опирается на проверку ряда равенств, которые выполняются для простых чисел. Если хотя бы одно такое равенство не выполняется, это доказывает что число составное[5].
Для теста Миллера — Рабина используется следующее утверждение:

> Пусть n — простое число и $n-1=2^s d$, где d — нечётно. Тогда для любого a из $\mathbb{Z}_n$ выполняется хотя бы одно из условий:
> 1. $a^d \equiv 1 \pmod{n}$
> 2. Существует целое число $r < s$ такое что $a^{2^r d} \equiv -1 \pmod{n}$

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

   for (int i = 0; i < y; i++)
   {
      result *= x;
      result %= N;
   }

   return result;
}

void mr(int n, int k)
{
   int s(0), d(n - 1), a, x;
   bool fin(true);

   while (d != 0 && d % 2 == 0)
   {
      s++;
      d /= 2;
   }

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

      if (x != 1 && x != n - 1)
         for (int r = 0; r < s; r++)
```

```
        {
            x = (x * x) % n;

            if (x == n - 1)
            {
                printf("\n[a = %4d] - свидетель простоты числа %d", a, n);
                break;
            }
        }

      if (x != n - 1)
        fin = true;

   }

   fin ? printf("\nЧисло %d - составное\n\n", n) : printf("\nЧисло %d - вероятно простое\n\n", n);
}

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

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

   printf("\nПроверим число %d на простоту по тесту Миллера-Рабина:\n  ", n);
   printf("Пусть n - простое, n - 1 = d * 2^s, d - нечетно.\n  ");
   printf("Тогда для всех a из Z*n выполнится хотя бы одно условие:");
   printf("\n      1 a^d = 1 (mod n)");
   printf("\n      2 Существует целое r<s : a^(d * 2^r) = -1 (mod n)\n");

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

   mr(n, k);

   return 0;
}
```

Пример запуска программы

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


Введите число n: 3469

Проверим число 3469 на простоту по тесту Миллера-Рабина:
  Пусть n - простое, n - 1 = d * 2^s, d - нечетно.
  Тогда для всех a из Z*n выполнится хотя бы одно условие:
      1 a^d = 1 (mod n)
      2 Существует целое r<s : a^(d * 2^r) = -1 (mod n)

Введите количество проверок k: 10

[a = 1413] - свидетель простоты числа 3469
[a =  909] - свидетель простоты числа 3469
[a = 2361] - свидетель простоты числа 3469
[a = 2798] - свидетель простоты числа 3469
[a =  326] - свидетель простоты числа 3469
Число 3469 - 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: 3470

Проверим число 3470 на простоту по тесту Миллера-Рабина:
  Пусть n - простое, n - 1 = d * 2^s, d - нечетно.
  Тогда для всех a из Z*n выполнится хотя бы одно условие:
      1 a^d = 1 (mod n)
      2 Существует целое r<s : a^(d * 2^r) = -1 (mod n)

Введите количество проверок k: 10

Число 3470 - составное

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