

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

Построение больших простых чисел с помощью теста Люка

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 2022

Описание алгоритма

Критерий Люка. Тест простоты Люка – это тест простоты натурального числа n . Для его разложения необходимо знать разложение числа $n - 1$ на множители. Для простого числа n простые множители числа $n - 1$ вместе с некоторым основанием a составляют сертификат Пратта, который позволяет подтвердить за полиномиальное время, что число n является простым. Алгоритм: Пусть $n > 1$ – натуральное число. Если существует целое a такое, что $1 < a < n$ и $a^{n-1} \equiv 1 \pmod{n}$ и для любого простого делителя q числа $n - 1$ $a^{(n-1)/q} \not\equiv 1 \pmod{n}$ то n простое. Если такого числа a не существует, то n – составное число.

Код программы

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

using namespace std;

vector<int> prime_dividers;

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;
}

void fact(int n)
{
    prime_dividers = {};

    int tmp(n);

    for (auto cnt = 2; cnt * cnt <= tmp; cnt++)
    {
        if (tmp % cnt == 0)
            prime_dividers.push_back(cnt);

        while (tmp % cnt == 0)
            tmp = tmp / cnt;
    }

    if (tmp > 1)
        prime_dividers.push_back(tmp);
}
```

```

    // printf("\nДелители n - 1 = %d", n - 1);
    // for(auto e: prime_dividers)
    //     printf(" [%d]", e);
}

```

```

int fastPow(int num, int deg)

```

```

{
    int result = 1;

    while (deg)
    {
        if (deg % 2 == 0)
        {
            deg /= 2;
            num *= num;
        }
        else
        {
            deg--;
            result *= num;
        }
    }
}

```

```

    return result;
}

```

```

bool lucasTest(int a, int n)

```

```

{
    if (fastPow(a, n - 1) % n != 1)
        return false;

    for (auto e : prime_dividers)
        if (fastPow(a, (n - 1) / e) % n == 1)
            return false;

    return true;
}

```

```

pair<int, int> genPrime(int n)

```

```

{
    int a = n - 1;

    n = fastPow(2, n) - 1;

    // printf("\nNew n = %d\n", n);
    fact(n - 1);

    while (!lucasTest(a, n) && a > 1)
        a--;

    return make_pair(a, n);
}

```

```

int main()

```

```

{
    setlocale(0, "");

    int n;

```

```

pair<int, int> res;

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

n = 3;
res = genPrime(n);
printf("\n\nПростое число для n = %3d: %7d | Свидетель: %4d\n\n",
      n, res.second, res.first);

n = 5;
res = genPrime(n);
printf("\n\nПростое число для n = %3d: %7d | Свидетель: %4d\n\n",
      n, res.second, res.first);

n = 7;
res = genPrime(n);
printf("\n\nПростое число для n = %3d: %7d | Свидетель: %4d\n\n",
      n, res.second, res.first);

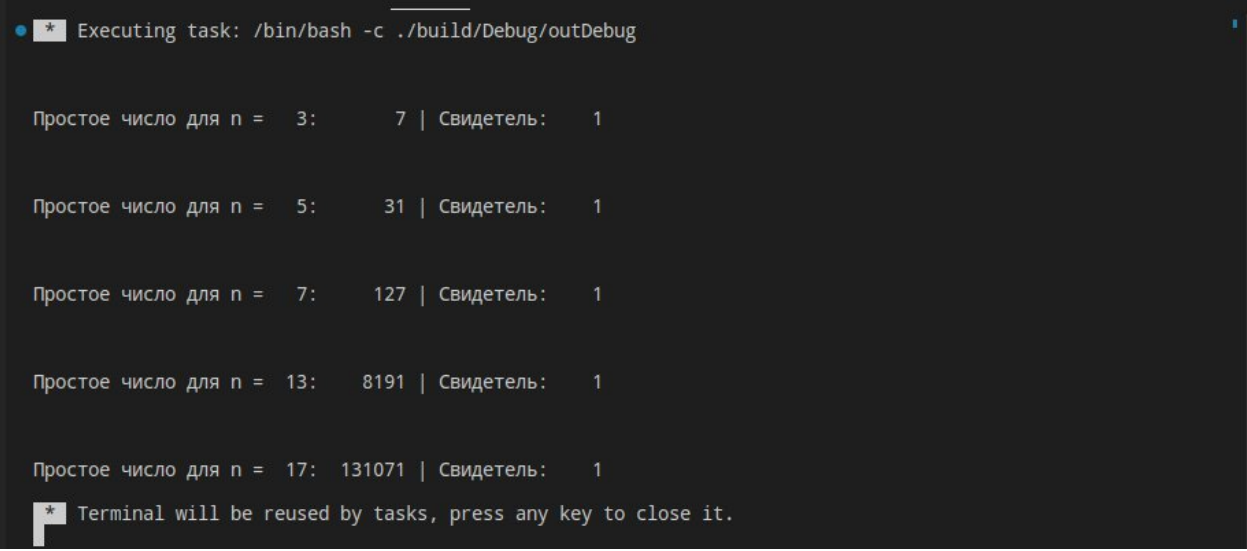
n = 13;
res = genPrime(n);
printf("\n\nПростое число для n = %3d: %7d | Свидетель: %4d\n\n",
      n, res.second, res.first);

n = 17;
res = genPrime(n);
printf("\n\nПростое число для n = %3d: %7d | Свидетель: %4d\n\n",
      n, res.second, res.first);

return 0;
}

```

Пример запуска программы



```

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

Простое число для n =   3:      7 | Свидетель:    1

Простое число для n =   5:     31 | Свидетель:    1

Простое число для n =   7:    127 | Свидетель:    1

Простое число для n =  13:   8191 | Свидетель:    1

Простое число для n =  17: 131071 | Свидетель:    1

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

```