

# Лабораторная работа №5

---

## Лабораторная работа №5

Цель работы

Задание

Теоретическое введение

Оборудование

## Выполнение лабораторной работы

Выводы

Список литературы

## Цель работы

---

Реализовать с помощью программирования программы определяющие отношение к составным или простым числам, методами, описанными в задании к лабораторной работе №5.

## Задание

---

Разработать программы, которые будут представлять из себя:

1. Программа повторяющая тест Ферма.
2. Программа повторяющая тест Соловея-Штрассена (алгоритм Якоби будет его частью)
3. Тест Миллера-Рабина

## Теоретическое введение

---

Поскольку определение простоты чисел является актуальной задачей криптографии, математиками разработано большое количество алгоритмов, которые с высокой эффективностью за ограниченное время позволяют проверить число на простоту: *тест Миллера, Миллера-Рабина, Люка-Лемера, Пепина, Агравала-Каяла-Саксены, Соловея-Штрассена* и другие [1].

При проверке числа на простоту тестом Ферма выбирают несколько чисел  $a$ . Чем больше количество  $a$ , для которых утверждение истинно, тем больше вероятность, что число  $n$  простое. Однако существуют составные числа, для которых данное равенство выполняется для всех  $a$  взаимно простых с  $n$  - это числа Кармайкла. Чисел Кармайкла — бесконечное множество, наименьшее число Кармайкла — 561. Тем не менее, тест Ферма довольно эффективен для обнаружения составных чисел.[2]

Тест Миллера — Рабина — вероятностный полиномиальный тест простоты. Тест Миллера — Рабина позволяет эффективно определять, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера — Рабина часто используется в криптографии для получения больших случайных простых чисел.[3].

## Оборудование

---

Лабораторная работа выполнялась дома со следующими характеристиками техники:

- Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81GHz
- ОС Майкрософт Windows 10
- VirtualBox верс. 6.1.26

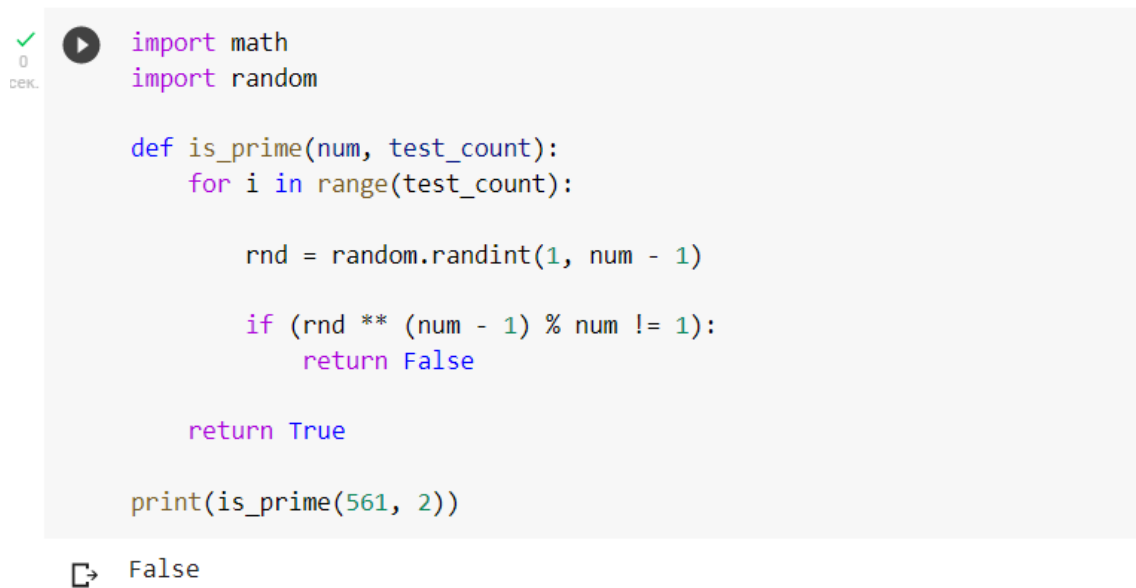
# Выполнение лабораторной работы

---

1. Реализовала программу определения простоты чисел по алгоритму Теста Ферма.

(рис. -@fig:001)

## ▼ Тест Ферма



```
import math
import random

def is_prime(num, test_count):
    for i in range(test_count):

        rnd = random.randint(1, num - 1)

        if (rnd ** (num - 1) % num != 1):
            return False

    return True

print(is_prime(561, 2))
```

False

Основная суть алгоритма лежит в том, чтобы сравнить число  $n$  (определяемое), с произвольным числом  $a$  ( $1 < a < n-1$ ).

Как видно на слайде, алгоритм в данном случае верно определяет, что число составное.

2. Реализовала программу Миллера-Рабина.

(рис. -@fig:003)

```
# factor n - 1 as 2^(r)*s
while r % 2 == 0:
    s = s + 1
    r = r // 2 # floor

# k = accuracy
for i in range(k):
    a = random.randrange(1, n)

    # a^(s) mod n = 1?
    if pow(a, s, n) == 1:
        return True

    # a^(2^(j) * s) mod n = -1 mod n?
    for j in range(r):
        if pow(a, 2**j*s, n) == -1 % n:
            return True

    return False

print(RabinMiller(15, 10))
```

➞ False

Тест Миллера — Рабина, наряду с тестом Ферма и тестом Соловея — Штрассена, позволяет эффективно определить, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера — Рабина часто используется в криптографии для получения больших случайных простых чисел.

Как мы видим, здесь алгоритм также верно определил, что цифра 15 является составным числом.

3. Реализовала алгоритм Соловея-Штрассена. (рис. -@fig:005)

```
if (n == 1):
    return ans;

return 0;

# To perform the Solovay- Strassen
# Primality Test
def solovoyStrassen(p, iterations):

    if (p < 2):
        return False;
    if (p != 2 and p % 2 == 0):
        return False;

    for i in range(iterations):

        # Generate a random number a
        a = random.randrange(p - 1) + 1;
        jacobian = (p + calculateJacobian(a, p)) % p;
        mod = modulo(a, (p - 1) / 2, p);

        if (jacobian == 0 or mod != jacobian):
            return False;

solovoyStrassen(25,2)
```

False

Тест всегда корректно определяет, что простое число является простым, но для составных чисел с некоторой вероятностью он может дать неверный ответ. Основное преимущество теста заключается в том, что он, в отличие от теста Ферма, распознает числа Кармайкла как составные. Также в этом алгоритме рассчитывается число Якоби, как часть программы.

Как мы видим, тут также алгоритм верно определил, что число является составным.

## Выводы

Освоила на практике написание алгоритмов проверки чисел на простоту.

## Список литературы

1. Алгоритм Ферма // Wikipedia URL: [https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\\_%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%B0](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%B0) (дата обращения: 24.12.2021).
2. Метода нахождения простых чисел // ФоксФОРД URL: [https://foxford.ru/wiki/matematika/pr\\_ostye\\_chisla](https://foxford.ru/wiki/matematika/pr_ostye_chisla) (дата обращения: 25.12.2021).

3. Тест Миллера-Рабина // Наука клуб URL: <https://nauka.club/matematika/test-Miller.html>  
(дата обращения: 27.10.2021).