

# Вероятностные алгоритмы проверки чисел на простоту

---

Андрианова Марина Георгиевна

8 ноября 2025

РУДН, Москва, Россия

## Лабораторная работа 5

---

Реализовать три вероятностных алгоритма проверки чисел на простоту и алгоритм вычисления символа Якоби.

### Задание

*Программно реализовать на языке Julia следующие алгоритмы:*

1. Алгоритм, реализующий тест Ферма
2. Алгоритм вычисления символа Якоби
3. Алгоритм, реализующий тест Соловья-Штрассена
4. Алгоритм, реализующий тест Миллера-Рабина

## 1. Тест Ферма - Введение обозначений и проверка условий

```
1  n = 17
2  a = 3
```

```
4  # Проверка условий
5
6  if (n % 2 == 0) | (n < 5)
7  |   println("Введите другое n")
8  else
9  |   println("Всё отлично продолжаем работу")
10 end
11
12
13 if (a < 2) | (a > n - 2)
14 |   println("Введите другое a")
15 else
16 |   println("Всё отлично продолжаем работу")
17 end
```

## 1. Тест Ферма - Реализация

```
19  # Тест Ферма
20
21  r = a^(n-1) % n
22
23  ✓ if r == 1
24      |   println("Число n, вероятно, простое")
25  ✓ else
26      |   println("Число n составное")
27  end
```

Рис. 1: Реализация

## 2. Символ Якоби - Введение обозначений и проверка условий

```
1  n = 15
2  a = 9
3
4  # Проверка условий
5
6  if (n % 2 == 0) | (n < 3)
7      println("Введите другое n")
8  else
9      println("Всё отлично продолжаем работу")
10 end
11
12
13 if (a < 0) | (a >= n)
14     println("Введите другое a")
15 else
16     println("Всё отлично продолжаем работу")
17 end
```

## 2. Символ Якоби - Дополнительная функция

```
19  # функция для приведения a к виду  $2^k \cdot a_1$ 
20
21  function devider(a)
22      k = 0
23      while a % 2 == 0
24          k += 1
25          a = Int(a / 2)
26      end
27      return a, k
28  end
```

Рис. 3: Функция для выделения четной части

## 2. Символ Якоби - Реализация

```
33 function jacoby(a, n, g = 1)
34
35 while a >= 0
36     if a == 0
37         return 0
38     elseif a == 1
39         return g
40     end
41
42     a1, k = devide(a)
43
44     if (k % 2 == 0)
45         s = 1
46     else
47         if ((n-1) % 8 == 0) || ((n + 1) % 8 == 0)
48             s = 1
49         elseif ((n - 3) % 8 == 0) || ((n + 3) % 8 == 0)
50             s = -1
51         end
52     end
53
54     if a1 == 1
55         result = g*s
56         return result
57     end
58
59     if ((n - 3) % 4 == 0) && ((a1 - 3) % 4 == 0)
60         s = s * (-1)
61     end
62
63     a = n % a1
64     n = a1
65     g = g * s
66 end
67 end
```



### 3. Тест Соловья-Штрассена - Введение обозначений и проверка условий

```
1  n = 17|
2  a = 3
3
4  # Проверка условий
5
6  if (n % 2 == 0) || (n < 5)
7  |   println("Введите другое n")
8  else
9  |   println("Всё отлично продолжаем работу")
10 end
11
12
13 if (a < 2) || (a > n - 2)
14 |   println("Введите другое a")
15 else
16 |   println("Всё отлично продолжаем работу")
17 end
```

### 3. Тест Соловья-Штрассена - Реализация

```
57   r = (a^((n-1)/2)) % n
58
59   function solovey_shtrassen(r, n, a)
60   if r != 1 && r != n - 1
61       return("число $n составное")
62   end
63
64   s = jacoby(a, n)
65
66   if s < 0
67       s += n
68   end
69
70   if (r - s) % n == 0
71       return("число $n, вероятно, простое")
72   else
73       return("число $n составное")
74   end
75   end
76
77   res = solovey_shtrassen(r, n, a)
78   println(res)
```

#### 4. Тест Миллера-Рабина - Введение обозначений и проверка условий

```
1  n = 15|
2  a = 2
3
4  # Проверка условий
5
6  if (n % 2 == 0) | (n < 5)
7      println("Введите другое n")
8  else
9      println("Всё отлично продолжаем работу")
10 end
11
12 if (a < 2) | (a > n - 2)
13     println("Введите другое a")
14 else
15     println("Всё отлично продолжаем работу")
16 end
17
```

## 4. Тест Миллера-Рабина - Реализация

```
30 y = (a ^ r) % n
31
32 function miller_rabin(y, n, s)
33     if y == 1 || y == n - 1
34         return "Число $n, вероятно, простое"
35     end
36
37     for j in 1:(s - 1)
38         y = (y * y) % n
39
40         if y == n - 1
41             return "Число $n, вероятно, простое"
42         end
43
44         if y == 1
45             return "Число $n составное"
46         end
47     end
48
49     # Если ни одно из условий не выполнено, то n составное
50     return "Число $n составное"
51 end
52
```

В процессе выполнения работы я реализовала разные виды вероятностных алгоритмов проверки чисел на простоту на языке программирования Julia.

Спасибо за внимание!