Лабораторная работа №5

Математические основы защиты информации и информационной безопасности

Николаев Дмитрий Иванович, НПМмд-02-24

Содержание

# 1 Цель работы

Изучить работу алгоритмов проверки чисел на простоту: алгоритм, реализующий тест Ферма; алгоритм вычисления символа Якоби; алгоритм, реализующий тест Соловэя-Штрассена; алгоритм, реализующий тест Миллера-Рабина; а также реализовать их программно.

# 2 Теоретическое введение

## 2.1 Вероятностные алгоритмы проверки чисел на простоту

Пусть — целое число. Числа $ , a$ называются *тривиальными делителями* числа .

Целое число называется *простым*, если оно не является делителем единицы и не имеет других делителей, кроме тривиальных. В противном случае число называется *составным*.

Например, числа являются простыми.

Пусть . Целые числа и называются сравнимыми по модулю (обозначается ), если разность делится на . Также эта процедура называется нахождением остатка от целочисленного деления на .

Проверка чисел на простоту является составной частью алгоритмов генерации простых чисел, применяемых в криптографии с открытым ключом. Алгоритмы проверки на простоту можно разделить на вероятностные и детерминированные.

*Детерминированный* алгоритм всегда действует по одной и той же схеме и гарантированно решает поставленную задачу (или не дает никакого ответа).

*Вероятностный* алгоритм использует генератор случайных чисел и дает негарантированно точный ответ. Вероятностные алгоритмы в общем случае не менее эффективны, чем детерминированные (если используемый генератор случайных чисел всегда дает набор одних и тех же чисел, зависящих от входных данных, то вероятностный алгоритм становится детерминированным).

Для проверки на простоту числа вероятностным алгоритмом выбирают случайное число () и проверяют условия алгоритма. Если число не проходит тест по основанию , то алгоритм выдает результат «Число составное», и число действительно является составным.

Если же проходит тест по основанию , ничего нельзя сказать о том, действительно ли число является простым. Последовательно проведя ряд проверок таким тестом для разных и получив для каждого из них ответ «Число вероятно, простое», можно утверждать, что число является простым с вероятностью, близкой к 1. После независимых выполнений теста вероятность того, что составное число будет раз объявлено простым (вероятность ошибки), не превосходит .

### 2.1.1 Алгоритм, реализующий тест Ферма

Тест Ферма основан на малой теореме Ферма: для простого числа и произвольного числа , , выполняется сравнение

Следовательно, если для нечётного существует такое целое , что , и , то число составное. Отсюда получаем следующий вероятностный алгоритм проверки числа на простоту.

**Вход**: Нечётное целое число .  
**Выход**: «Число , вероятно, простое» или «Число составное».

1. Выбрать случайное целое число , .
2. Вычислить .
3. При результат: «Число , вероятно, простое». В противном случае результат: «Число составное».

На шаге 1 мы не рассматривали числа и , поскольку для любого целого и для любого нечётного .

### 2.1.2 Алгоритм вычисления символа Якоби

**Вход**: Нечётное целое число , целое число , .  
**Выход**: Символ Якоби .

1. Положить .
2. При результат: .
3. При результат: .
4. Представить в виде , где число нечётное.
5. При четном положить , при нечётном положить , если ; положить , если .
6. При результат: .
7. Если и , то .
8. Положить , , и вернуться на шаг 2.

### 2.1.3 Алгоритм, реализующий тест Соловэя-Штрассена

*Тест Соловэя-Штрассена* основан на критерии Эйлера: нечётное число является простым тогда и только тогда, когда для любого целого числа , , взаимно простого с , выполняется сравнение:

где — *символ Якоби*.

Пусть , где и числа простые (не обязательно различные). *Символ Якоби* определяется равенством

**Вход**: Нечётное целое число .  
**Выход**: «Число , вероятно, простое» или «Число составное».

1. Выбрать случайное целое число , .
2. Вычислить .
3. При и результат: «Число составное».
4. Вычислить символ Якоби .
5. При результат: «Число составное». В противном случае результат: «Число вероятно, простое».

### 2.1.4 Алгоритм, реализующий тест Миллера-Рабина

На сегодняшний день для проверки чисел на простоту чаще всего используется тест Миллера-Рабина, основанный на следующем наблюдении. Пусть число нечётное и , где — нечётное. Если простое, то для любого , взаимно простого с , выполняется условие .

**Вход**: Нечётное целое число .  
**Выход**: «Число , вероятно, простое» или «Число составное».

1. Представить в виде , где число нечётное.
2. Выбрать случайное целое число , .
3. Вычислить .
4. При и выполнить следующие действия:
   1. Положить .
   2. Если и , то:
      1. Положить .
      2. При результат: «Число составное».
      3. Положить .
   3. При результат: «Число составное».
5. Результат: «Число , вероятно, простое».

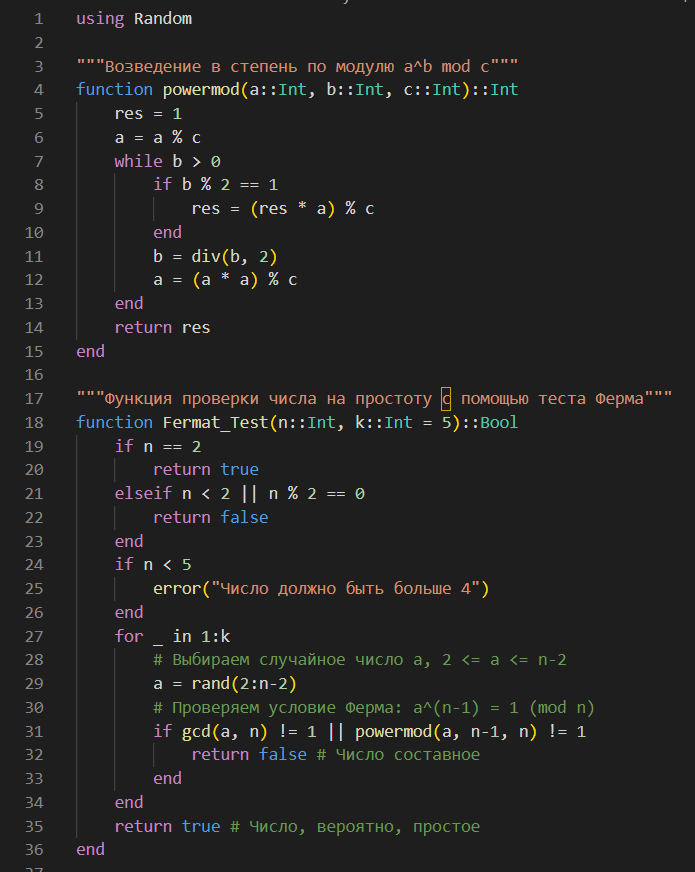
## 2.2 Резюме алгоритмов

1. **Тест Ферма**: Этот алгоритм проверяет, удовлетворяет ли случайное число соотношению , где . Если это условие не выполняется, число составное.
2. **Символ Якоби**: Функция вычисляет символ Якоби рекурсивно, используя свойства сравнения и перестановки чисел по модулю.
3. **Тест Соловэя-Штрассена**: Использует символ Якоби для проверки условия . Если это условие не выполняется для какого-либо , число составное.
4. **Тест Миллера-Рабина**: Использует разложение на и проверяет условия для случайных , используя возведение в степень по модулю. Этот тест считается более надежным и чаще используется.

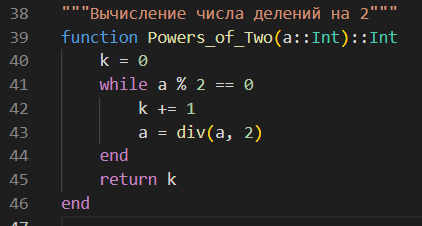
# 3 Выполнение лабораторной работы

Действуя согласно [1], реализуем все описанные алгоритмы проверки чисел на простоту на языке Julia.

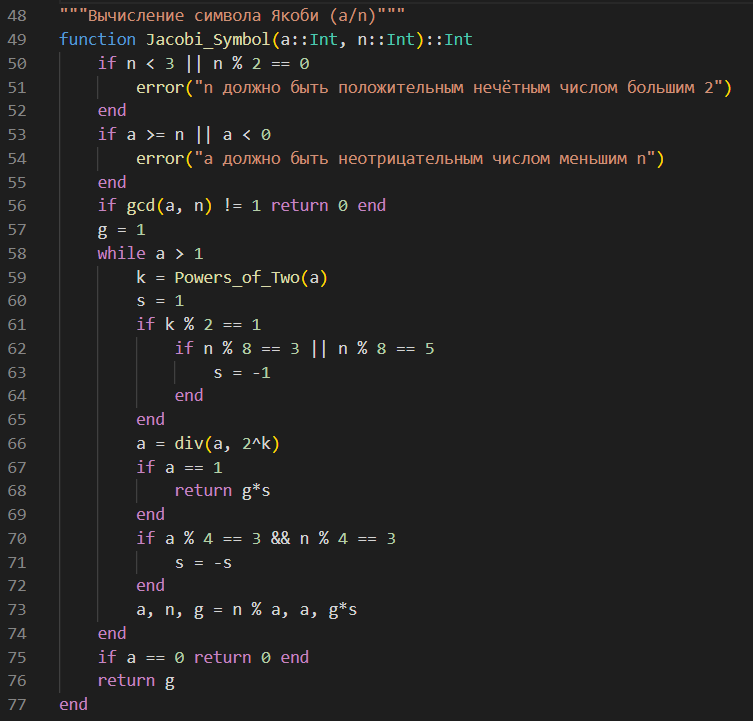
Сначала реализуем функцию модульного экспоненциирования и алгоритм теста Ферма (Рис.[??]). Для реализации алгоритма теста Соловэя-Штрассена (Рис.[??]) предварительно определим функции вычисления числа делений некоторого заданного числа на 2 (Рис.[??]) и символа Якоби (Рис.[??]). После чего реализуем алгоритм Миллера-Рабина (Рис.[??]). Для следующего набора чисел: 13 - простое, 15 - составное, 17 - простое, 19 - простое, 561 - составное, 1105 - составное, 1729 - составное, 2143 - простое, 2399 - простое (Рис.[??]), проверим работу данных алгоритмов, в результате чего получим следующий вывод, представленный на Рис.[??].



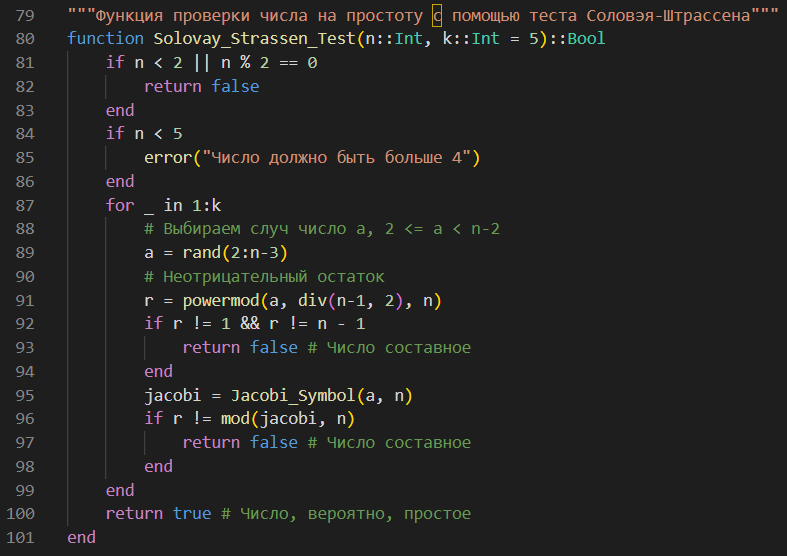
Код алгоритма теста Ферма на Julia



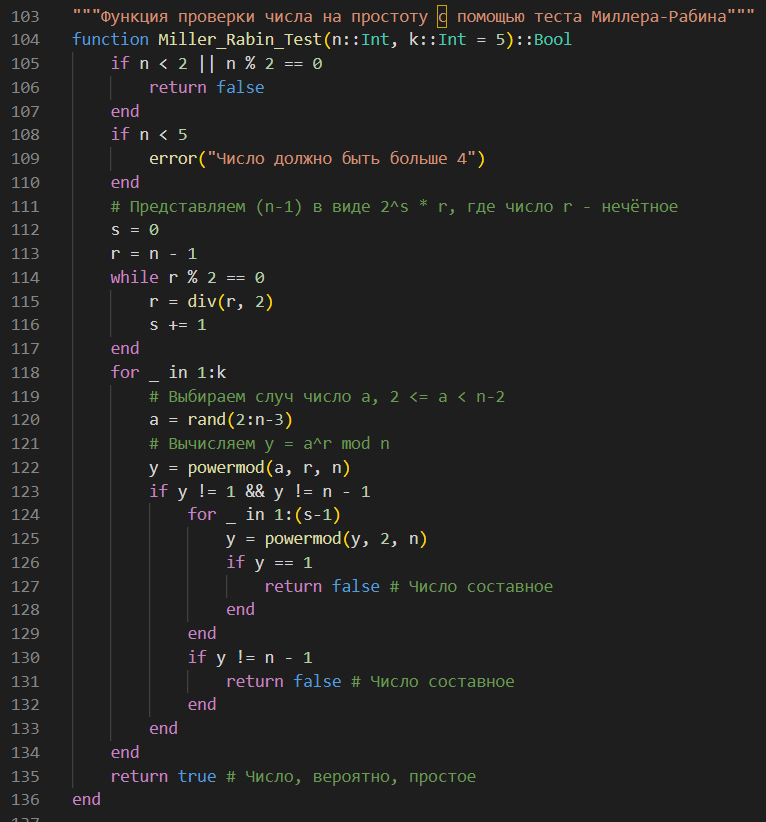
Код вычисления количества делений некоторого числа на 2 на Julia



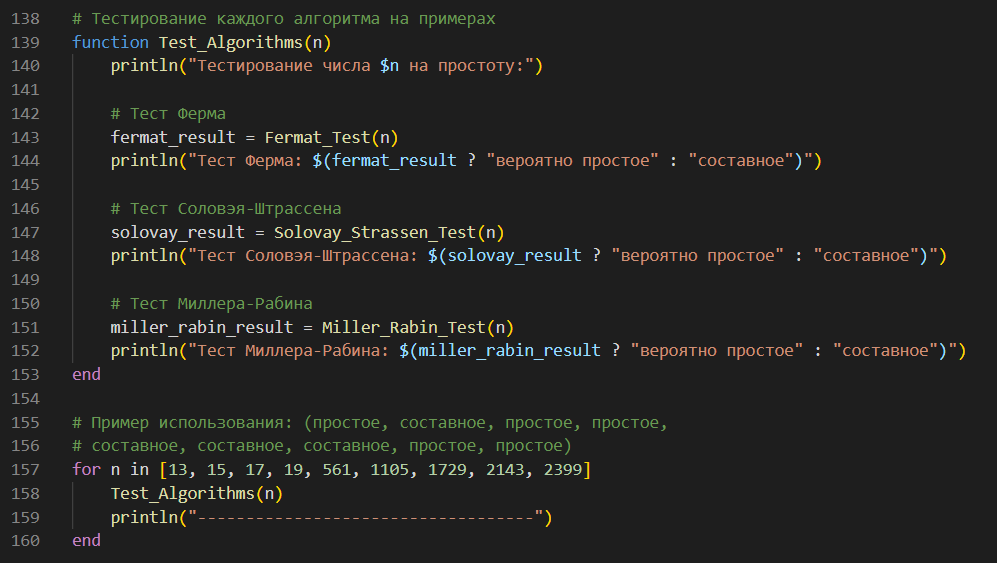
Код вычисления символа Якоби на Julia



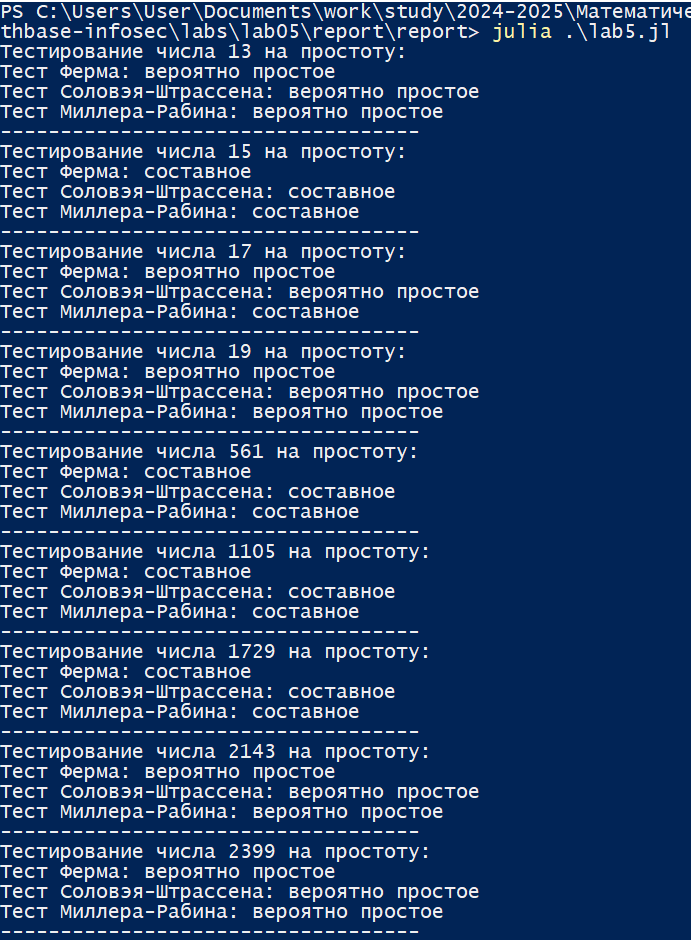
Код алгоритма теста Соловэя-Штрассена на Julia



Код алгоритма теста Миллера-Рабина на Julia



Начальные данные для сравнения алгоритмов проверки чисел на простоту на Julia



Результат выполнения кода и сравнения алгоритмов проверки чисел на простоту на Julia

# 4 Выводы

В ходе выполнения лабораторной работы я изучил работу алгоритмов проверки чисел на простоту: алгоритма, реализующего тест Ферма; алгоритма вычисления символа Якоби; алгоритма, реализующего тест Соловэя-Штрассена; алгоритма, реализующего тест Миллера-Рабина; а также реализовал их программно на языке Julia.

# Список литературы

1. Лабораторная работа № 5. Вероятностные алгоритмы проверки чисел на простоту [Электронный ресурс]. Саратовский государственный университет имени Н.Г. Чернышевского, 2024.