Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

[Кафедра информационных](https://www.belstu.by/fakultety/fit/vm) систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №3**

по дисциплине Информационная безопасность

Тема: Основы теории чисел и их использование в криптографии

Исполнитель:

Студент 3 курса группы 6

Руководитель:

Ассистент Нистюк О. А.

Минск, 2024

**Лабораторная работа №3**

**Цель:** приобретение практических навыков выполнения операций с числами для решения задач в области криптографии и разработка приложений для автоматизации этих операций.

**Задачи:**

1. Закрепить теоретические знания по высшей арифметике.

2. Научиться практически решать задачи с использованием простых и взаимно простых чисел, вычислений по правилам модулярной арифметики и нахождению обратных чисел по модулю.

3. Ознакомиться с особенностями реализации готового программного средства L\_PROST и особенностями выполнения с его помощью операций над простыми числами.

4. Разработать приложение для реализации указанных преподавателем операций с числами.

5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения эксперимента с использованием приложения и результатов эксперимент.

# Теоретические сведения

Множество всех целых чисел (обозначим буквой Z) есть набор всех действительных чисел без дробной части: {..., –3, –2, –1, 0, 1, 2, 3, ...}.

Натуральные числа являются подмножеством целых чисел и образуют множество N: {1, 2, 3, ...}.

Делимость – одно из основных понятий теории чисел. Если для некоторого целого числа a и натурального числа b существует целое число q, при котором bq = a, то говорят, что число a делится на b. В этом случае b называется делителем числа a, а a называется кратным числу b.

Делитель a называется собственным делителем числа b, если 1 < |a| < |b|, и несобственным – в противном случае.

Всякое целое число а можно представить с помощью положительного целого числа b равенством вида а = bq + r, 0 ≤ r ≤ b. Число q называется неполным частным, а число r – остатком отделения а на b.

Натуральное число n называется простым, если n > 1 и не имеет положительных делителей, отличных от 1 и n.

Любое составное число представляется уникальным образом в виде произведения простых чисел; иначе еще говорят, что разложение числа на простые множители однозначно.

Всякое натуральное число n, кроме 1, можно представить как произведение простых множителей: n = p1p2p3...pz, z > 1.

Простых чисел бесконечно много, причем существует примерно n / ln(n) простых чисел, меньших числа n.

Наименьший простой делитель составного числа n не превышает √n, поэтому для проверки простоты числа достаточно проверить его делимость на 2 и на все нечетные (а еще лучше простые) числа, не превосходящие √n.

Любое четное число, большее 2, представимо в виде суммы двух простых чисел, а любое нечетное, большее 5, представимо в виде суммы трех простых чисел.

Для любого натурального n, большего 1, существует хотя бы одно простое число на интервале от n до 2n.

Натуральное число n называется составным, если n > 1 и имеет по крайней мере один положительный делитель, отличный от 1 и n.

Положительный наименьший собственный делитель составного числа n есть простое число.

Если два простых числа отличаются на 2, то их называют числами-близнецами.

Первый алгоритм нахождения простых чисел, не превышающих n, был придуман Эратосфеном во II в. до н. э. и известен сейчас как «решето Эратосфена». Его суть в последовательном исключении из списка целых чисел от 1 до n (или из сокращенного диапазона, например от m до n, 1 < m ≤ n) чисел, кратных 2, 3, 5 и другим простым числам, уже найденным «решетом».

Наибольшее целое число, которое делит без остатка числа a и b, называется наибольшим общим делителем этих чисел – НОД (a, b). Простым и эффективным средством вычисления НОД (a, b) является алгоритм Евклида.

Взаимно простыми являются целые числа, наибольший общий делитель которых равен 1.

Целые числа a и b взаимно просты тогда и только тогда, когда существуют такие целые числа u и v, что выполняется равенство:

аu + bv = 1.

Если НОД (a, b) = d, то справедливо следующее соотношение (соотношение Безу):

аu + bv = d.

Количество натуральных чисел, не превосходящих n и взаимно простых с n, называется функцией Эйлера и обозначается φ(n).

Если p – простое число, то φ(p) = p – 1, если числа p и q являются простыми и p ≠ q, то φ(p) = (p – 1)(q – 1).

Модулярная арифметика так же коммутативна, ассоциативна и дистрибутивна, как и обычная арифметика. В силу этих свойств сравнения можно почленно складывать, вычитать, умножать, возводить в степень.

Малая теорема Ферма. Если n – простое число, а число а не кратно n, то справедливо:

an ≡ 1 mod n.

В соответствии с обобщением Эйлера приведенной теоремы, если НОД (а, n) = 1, то справедливо:

aφ(n) mod n ≡ 1

Последнее выражение можно переписать в следующем виде:

а–1 mod n ≡ aφ(n) – 1 mod n.

# Ход работы

Целью работы была разработка приложения, позволяющего вычислять НОД двух или трех чисел и выполнять поиска простых чисел.

Для вычисления НОД была разработана функция gcd, принимающая три числа (последнее число является необязательным параметром). Код функции представлен на рисунке 2.1.

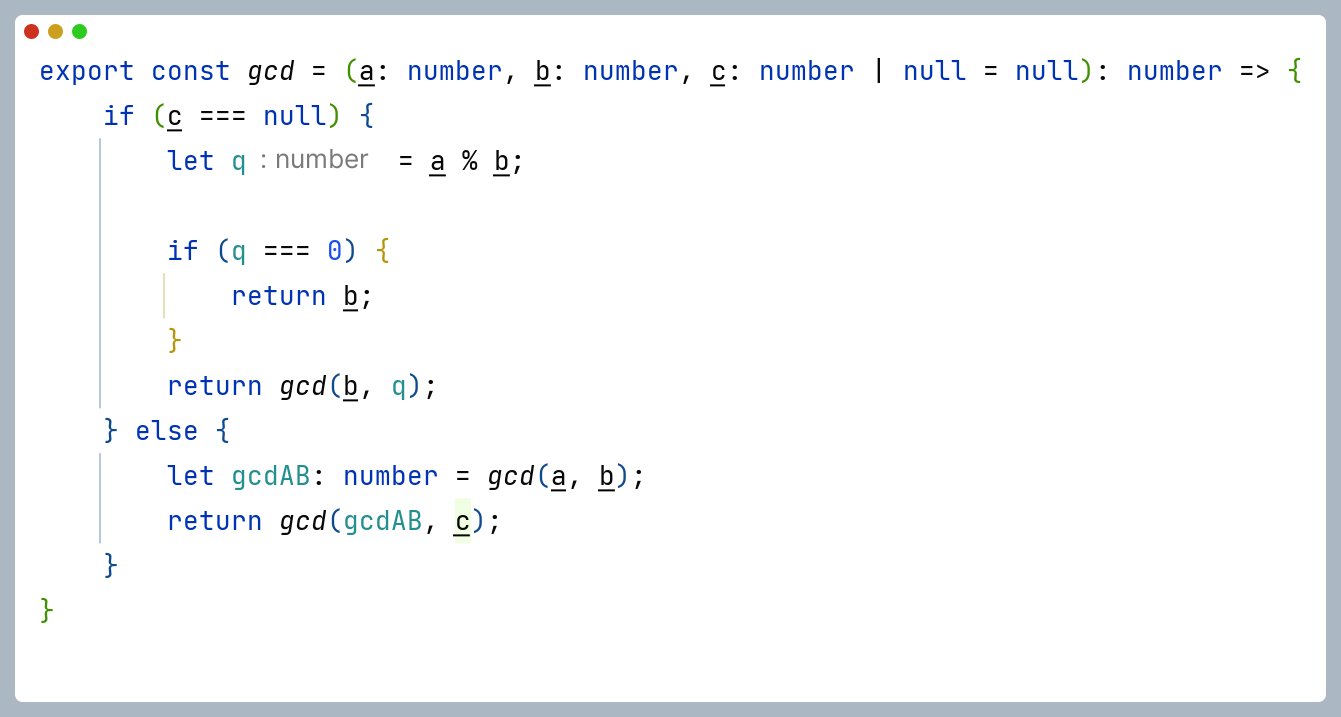


Рисунок 2.1 – Функция для подсчета НОД по алгоритму Евклида

Результат работы функции представлен на рисунке 2.2.

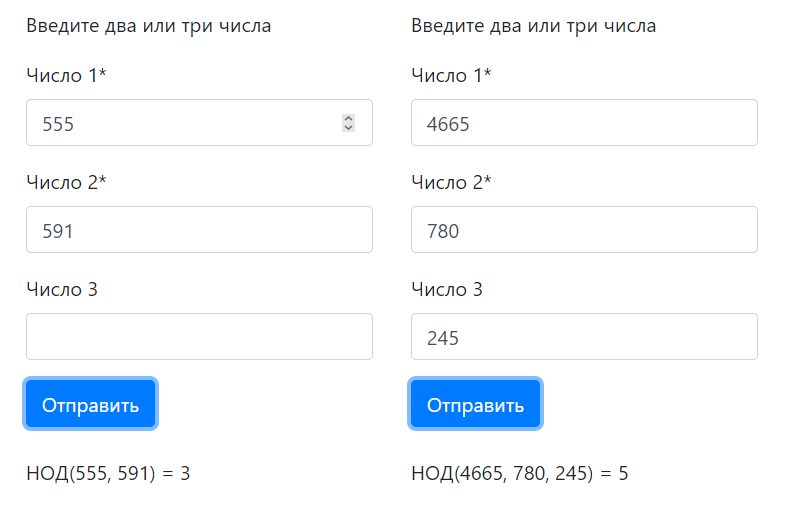


Рисунок 2.2 – Результат подсчета НОД для двух и трех чисел

Кроме того, была разработана функция isPrime, которая принимает одно число и проверяет, является ли оно простым. Код функции представлен на рисунке 2.3.

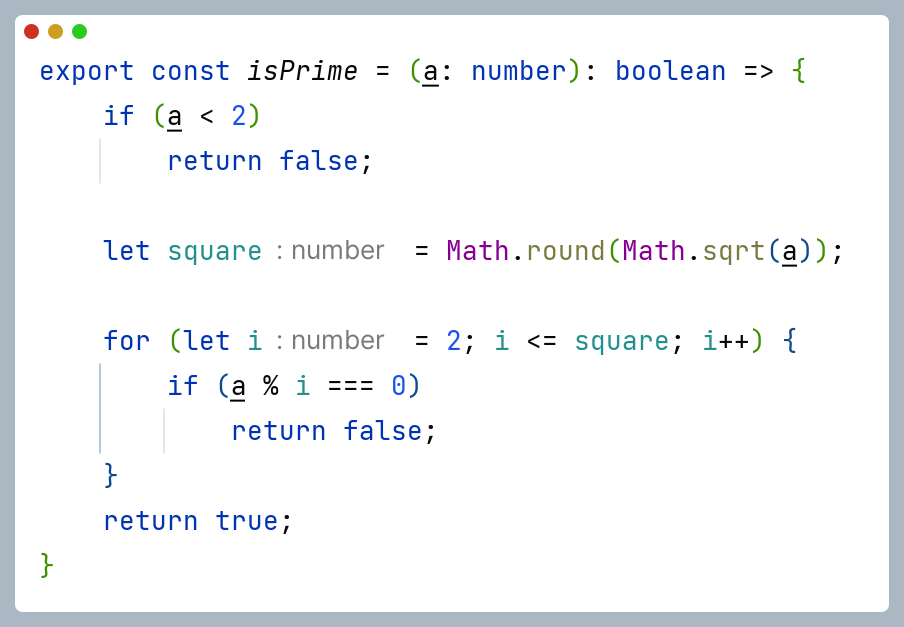


Рисунок 2.3 – Функция для определения простоты числа

С помощью функции isPrime было определено, является ли конкатенация чисел m и n простым числом. Результат работы функции представлен на рисунке 2.4.

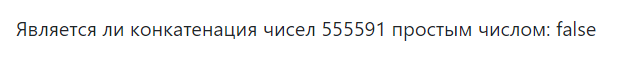


Рисунок 2.4 – Результат работы функции isPrime

Также была разработана функция primeFactors, которая позволяет записать число в виде произведения простых множителей в канонической форме. Код функции представлен на рисунке 2.5.

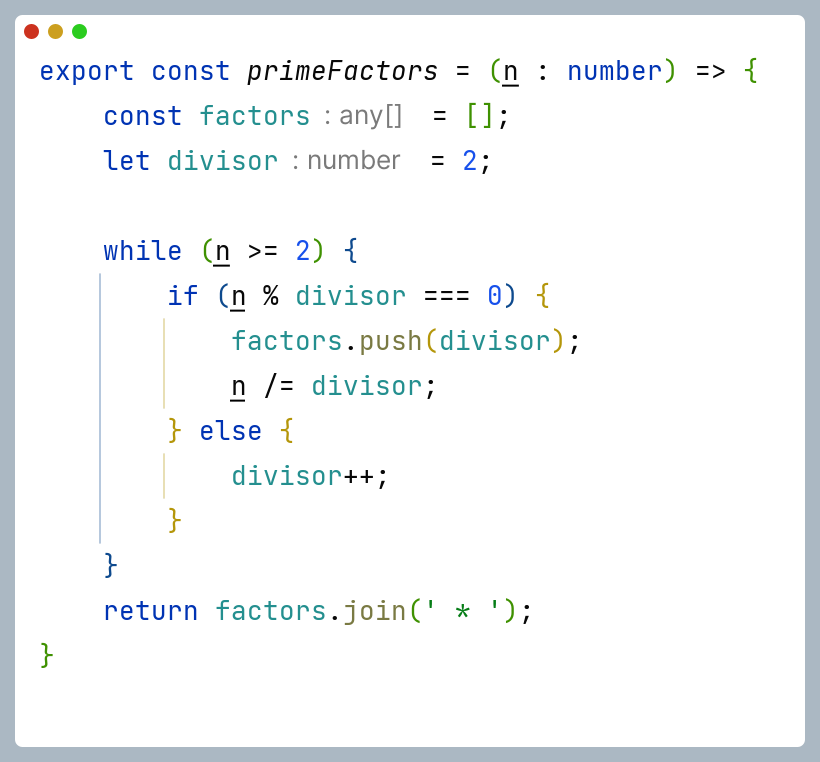


Рисунок 2.5 – Код функции primeFactors

Результат работы функции представлен на рисунке 2.6.

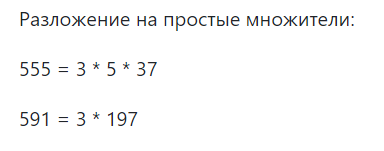


Рисунок 2.6 – Результат работы функции primeFactors

Далее была разработана функция sieveOfEratosthenes, которая по алгоритму «решето Эратосфена» находит все простые числа в интервале от m до n. Код функции представлен на рисунке 2.7.

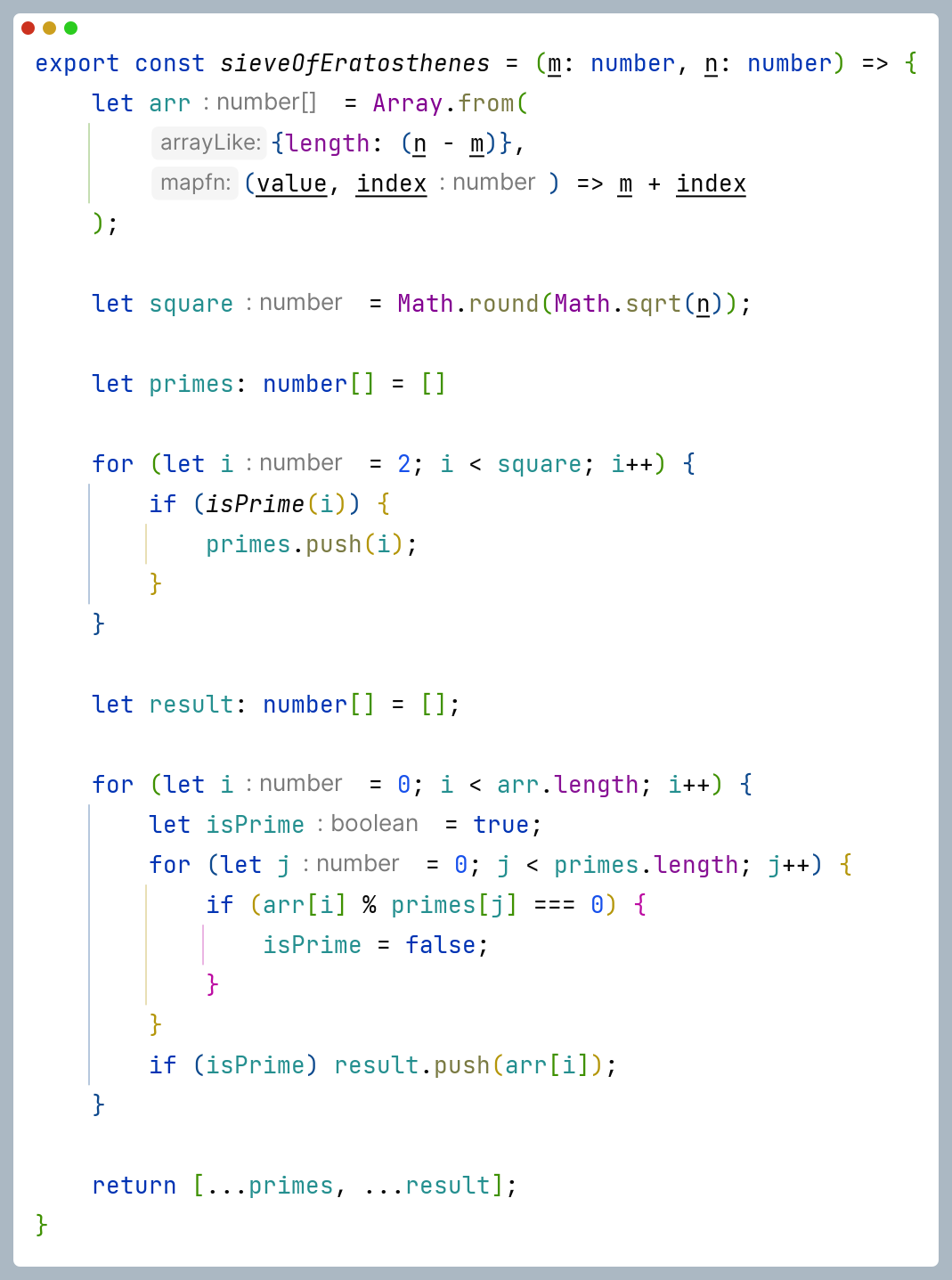


Рисунок 2.7 – Функция для нахождения простых чисел в заданном интервале

Результат работы функции представлен на рисунке 2.8.

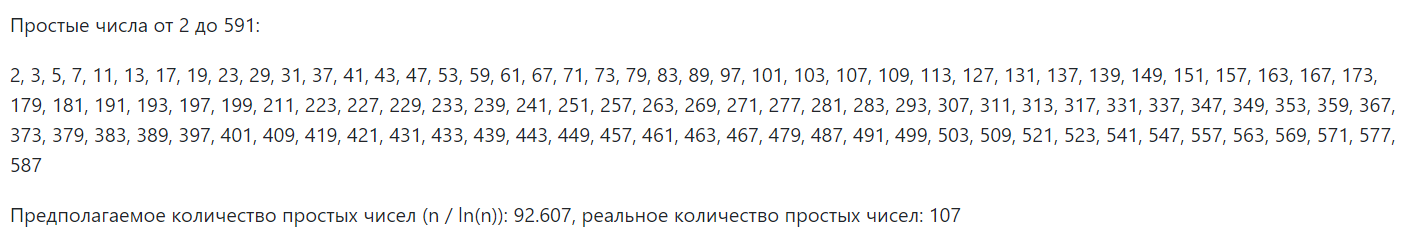


Рисунок 2.8 – Результат работы функции для нахождения простых чисел для m = 555, n = 591

Результат работы функции нужно было сравнить с ручными вычислениями, используя алгоритм «решето Эратосфена».

Порядок выполнения:

1. Исходный массив чисел: 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591.
2. Вычисление Значит для проверки простоты числа, достаточно проверить его делимость на 2 и на все простые числа, не превосходящие √n. Значит проверять делимость нужно будет на следующие числа: 2, 3, 5, 7, 11, 13, 17, 19, 23.
3. Удаление чисел из массива с учётом s = 2: 555, 557, 559, 561, 563, 565, 567, 569, 571, 573, 575, 577, 579, 581, 583, 585, 587, 589, 591.
4. Удаление чисел из массива с учётом s = 3: 557, 559, 563, 565, 569, 571, 575, 577, 581, 583, 587, 589.
5. Удаление чисел из массива с учётом s = 5: 557, 559, 563, 569, 571, 577, 581, 583, 587, 589.
6. Удаление чисел из массива с учётом s = 7: 557, 559, 563, 569, 571, 577, 583, 587, 589.
7. Удаление чисел из массива с учётом s = 11: 557, 559, 563, 569, 571, 577, 587, 589.
8. Удаление чисел из массива с учётом s = 13: 557, 563, 569, 571, 577, 587, 589.
9. Удаление чисел из массива с учётом s = 17: 557, 563, 569, 571, 577, 587, 589.
10. Удаление чисел из массива с учётом s = 19: 557, 563, 569, 571, 577, 587.
11. Удаление чисел из массива с учётом s = 23: 557, 563, 569, 571, 577, 587.

Итоговый массив: [557, 563, 569, 571, 577, 587], что совпадает с массивом чисел, полученном при работе приложения.

**Вывод:** в ходе выполнения лабораторной работы были освоены основы теории чисел, необходимые для применения в криптографии. Это включало в себя изучение свойств простых и составных чисел, взаимной простоты чисел, а также критериев делимости. Целью работы было укрепление теоретических знаний в области высшей арифметики и приобретение навыков решения практических задач, включающих простые и взаимно простые числа, а также операции модулярной арифметики и нахождение обратных чисел по модулю. В результате работы было разработано приложение, позволяющее выполнять такие операции с числами как вычисление НОД и нахождение простых чисел.