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

Лабораторная работа №3

**Основы теории чисел и их использование в криптографии**

Студент: Корнелюк. В. В.

ФИТ 3 курс 4 группа

Преподаватель: Нистюк О.А.

Минск 2025

**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

Перед выполнением основного задания было рекомендовано выполнить следующие задания.

Записать числа m и n (по вариантам) в виде произведения простых множителей.

m = 399 = 1 \* 3 \* 7 \* 19

n = 433 = 1 \* 433

Проверить, является ли число, состоящее из конкатенации цифр m ǀǀ n простым.

Для решения этой задачи была разработана функция isPrime, которая проверяет является ли число простым. Код функции представлен на рисунке 2.1.

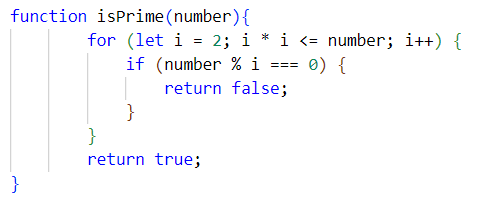


Рисунок 2.1 – Функции для проверки является ли число простым

Результат работы функции представлен на рисунке 2.2



Рисунок 2.1 – Результат работы функции isPrime

Найти НОД(m, n). Используя алгоритм Евклида найдем НОД(433, 399).

433 = 399 \* 1 + 43

399 = 34 \* 11 + 25

34 = 25 \* 1 + 9

25 = 9 \* 2 + 7

9 = 7 \* 1 + 2

7 = 2 \* 3 + 1

2 = 1 \* 2 + 0

Таким Образом НОД(433, 399) = 1.

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

Для вычисления НОД была разработана функция NOD принимающая два числа, и функция MODMultiple, принимающая переменное число параметров. Код функций представлен на рисунке 2.1.

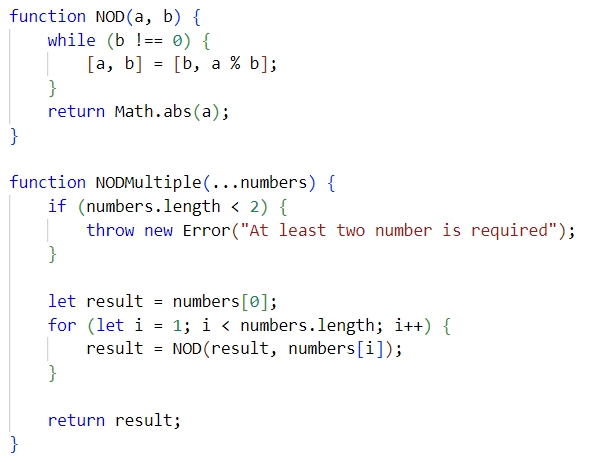


Рисунок 2.1 – Функции для подсчета НОД по алгоритму Евклида

Результат работы функции представлен на рисунке 2.2.

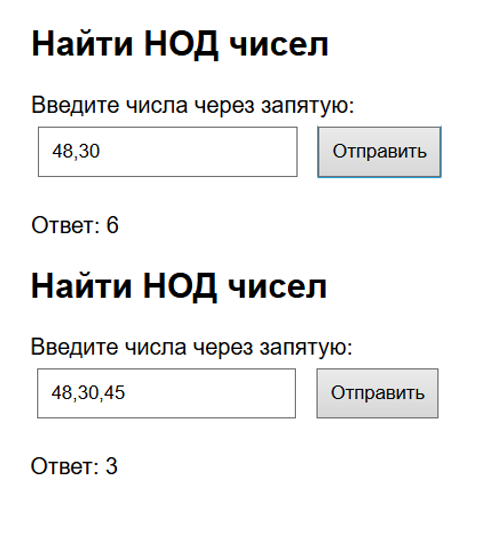


Рисунок 2.2 – Результат подсчета НОД для двух и трех чисел

Далее была разработана функция findPrimesInRange, которая по алгоритму «решето Эратосфена» находит все простые числа в интервале от m до n. Код функции представлен на рисунке 2.3.

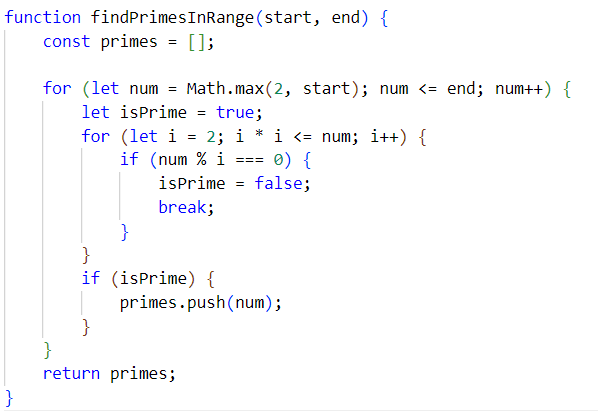


Рисунок 2.3 – Функция для нахождения простых чисел в заданном интервале

Результат работы функции представлен на рисунке 2.4.

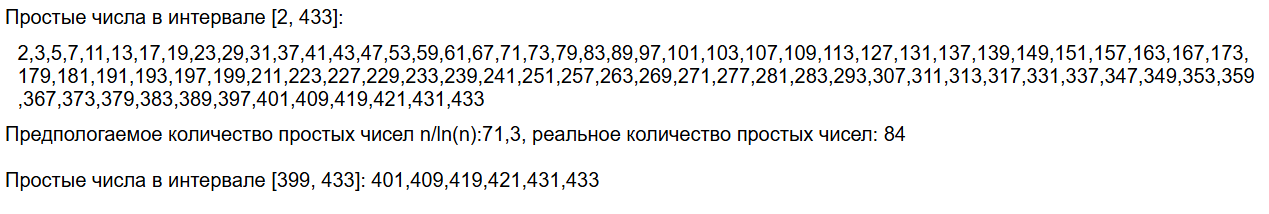


Рисунок 2.4 – Результат работы функции для нахождения простых чисел для m = 399, n = 433

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

Порядок выполнения:

1. Исходный массив чисел: 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433.
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.
3. Удаление чисел из массива с учётом s = 2: 399, 401, 403, 405, 407, 409, 411, 413, 415, 417, 419, 421, 423, 425, 427, 429, 431, 433.
4. Удаление чисел из массива с учётом s = 3: 401, 403, 405, 407, 409, 413, 415, 419, 421, 425, 427, 431, 433
5. Удаление чисел из массива с учётом s = 5: 401, 403, 407, 409, 413, 419, 421, 427, 431, 433.
6. Удаление чисел из массива с учётом s = 7: 401, 403, 407, 409, 419, 421, 431, 433.
7. Удаление чисел из массива с учётом s = 11: 401, 403, 409, 419, 421, 431, 433.
8. Удаление чисел из массива с учётом s = 13: 401, 409, 419, 421, 431, 433.
9. Удаление чисел из массива с учётом s = 17: 401, 409, 419, 421, 431, 433.
10. Удаление чисел из массива с учётом s = 19: 401, 409, 419, 421, 431, 433.
11. Удаление чисел из массива с учётом s = 23: 401, 409, 419, 421, 431, 433.

Итоговый массив: [401, 409, 419, 421, 431, 433.], 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.