МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего профессионального образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

Разработка программной системы для вычисления и анализа степеней примитивных корней для начальных диапазонов простых чисел

Научный руководитель:
Профессор кафедры
СП, д.ф.-м.н., доцент
Р.Ж. Алеев

Автор:
студент группы КЭ-433
Д.А. Манов

АКТУАЛЬНОСТЬ

Примитивные корни имеют важное значение в криптографии, так как они лежат в основе многих алгоритмов шифрования, таких как RSA и эллиптические кривые. Автоматизация вычисления примитивных корней позволяет быстро и эффективно находить необходимые параметры для криптографических систем, что особенно важно при работе с большими числами.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель:

Разработка программной системы для вычисления и анализа степеней примитивных корней для начальных диапазонов простых чисел.

Задачи:

1. Составить список простых чисел из диапазона [2..P].
2. Разработка функции для вычисления примитивных корней по модулю простых чисел из диапазона [2..P].
3. Разработка функции, которая будет находить для каждого примитивного корня количества простых чисел из диапазона [2..P]
4. Создать графическое приложение, в которое будут включены все указанные функции.

ПОНЯТИЕ «ПРИМИТИВНЫЙ КОРЕНЬ»

Первообразным корнем по модулю n (primitive root mood n) называется такое число g, что все его степени по модулю n пробегают по всем числам, взаимно простым с n. Математически это формулируется таким образом: для любого a взаимно простого с n есть такое целое k, что $g^k \equiv a(mod\ n)$.

Например: возьмем число 2 по модулю 5 и проверим его на примитивный корень.

$2^1 = 2 \equiv 2(mod\ 5)$

$2^2 = 4 \equiv 4(mod\ 5)$

$2^3 = 8 \equiv 3(mod\ 5)$

$2^4 = 16 \equiv 1(mod\ 5)$

Таким образом, 2 это примитивный корень, поскольку полученные результаты можно составить в цепочку от 1 до 4.

ТЕОРЕМА

Пусть p – простое нечётное число и α– натуральное число. Количество примитивных корней по модулю p равно $\varphi(p-1)$, где φ – функция Эйлера. Пусть g – примитивный корень по модулю p. Если p^2 не делит $g^{p-1}-1 \leftrightarrow g^{p-1}-$ $-1 \neq 0(mod\ p^2)$, то g – примитивный корень по модулю p^{α}.

Если g не является примитивным корнем по модулю p^{α} , то $g+p$ – примитивный корень по модулю p^{α}.

Например: возьмем число 29.

Тогда примитивные корни: 2, 3, 8, 10, 11, 14, 15, 18, 19, 20, 21 в ходе всех вычислений будут выстроены в цепочку чисел, где на рисунке 1 число 14 не удовлетворяет условиям теоремы и соответственно не будет являться примитивным корнем по модулю числа 29.

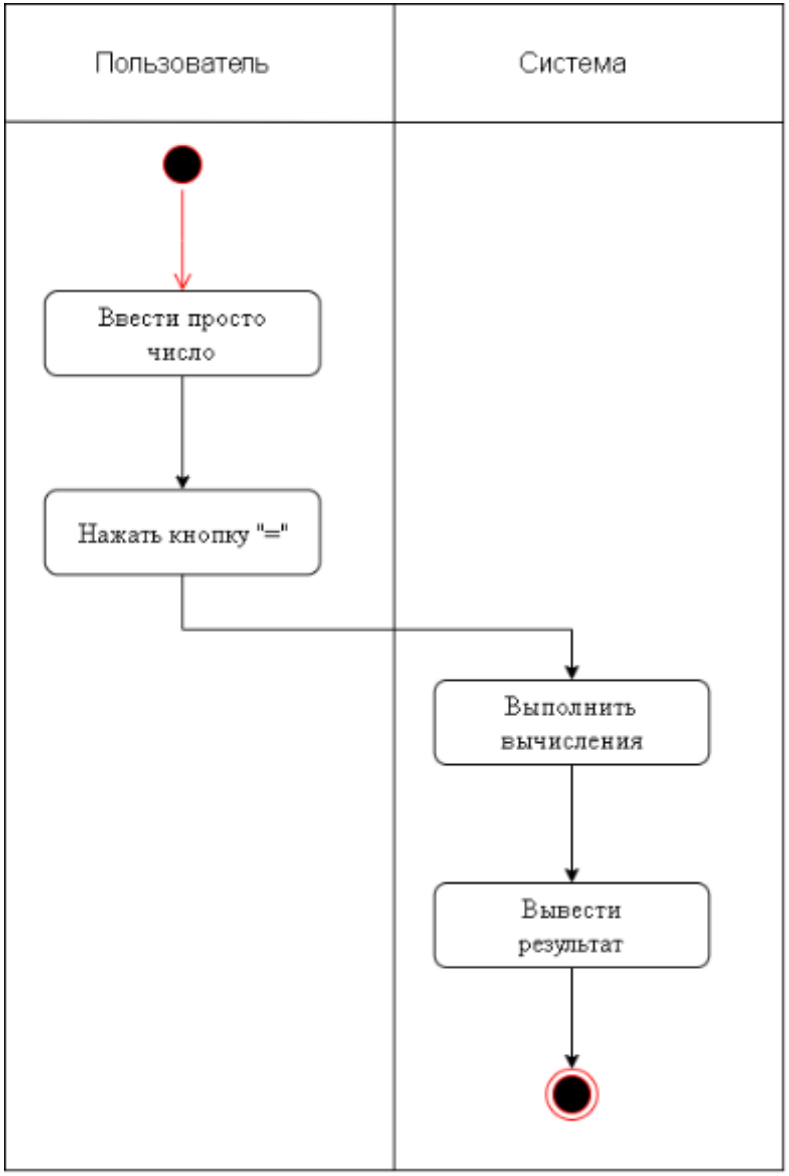
Подробнее о теореме можно прочитать в статье по ссылке: https://kaf401.rloc.ru/Criptfiles/primroots.htm

$$2^{29-1} - 1 \neq 0(mod 29^2)$$
$$3^{29-1} - 1 \neq 0(mod 29^2)$$
$$8^{29-1} - 1 \neq 0(mod 29^2)$$
$$10^{29-1} - 1 \neq 0(mod 29^2)$$
$$11^{29-1} - 1 \neq 0(mod 29^2)$$
$$14^{29-1} - 1 \equiv 0(mod 29^2)$$
$$15^{29-1} - 1 \neq 0(mod 29^2)$$
$$18^{29-1} - 1 \neq 0(mod 29^2)$$
$$19^{29-1} - 1 \neq 0(mod 29^2)$$
$$20^{29-1} - 1 \neq 0(mod 29^2)$$
$$21^{29-1} - 1 \neq 0(mod 29^2)$$

Рисунок 1 – Проверка числа 29 по теореме 2.

Таким образом: 2, 3, 8, 10, 11, 15, 18, 19, 20, 21– примитивные корни по модулю 29^{α} для любого натурального числа α.

ДИАГРАММА ДЕЯТЕЛЬНОСТИ



СРЕДСТВА РАЗРАБОТКИ

Язык программирования:

Python версии 3.12.2

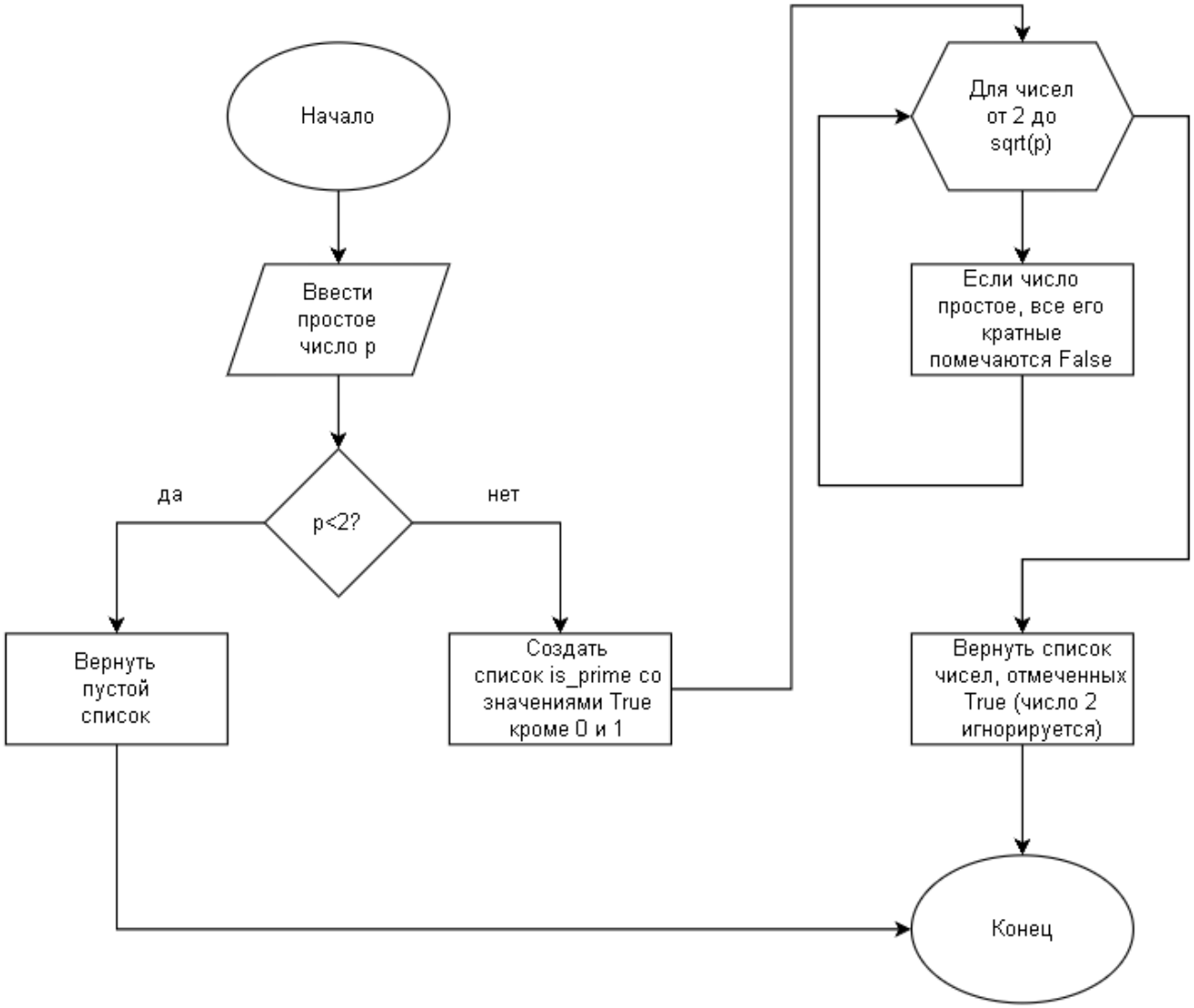
Среда разработки:

PyCharm Community Edition 2020.1

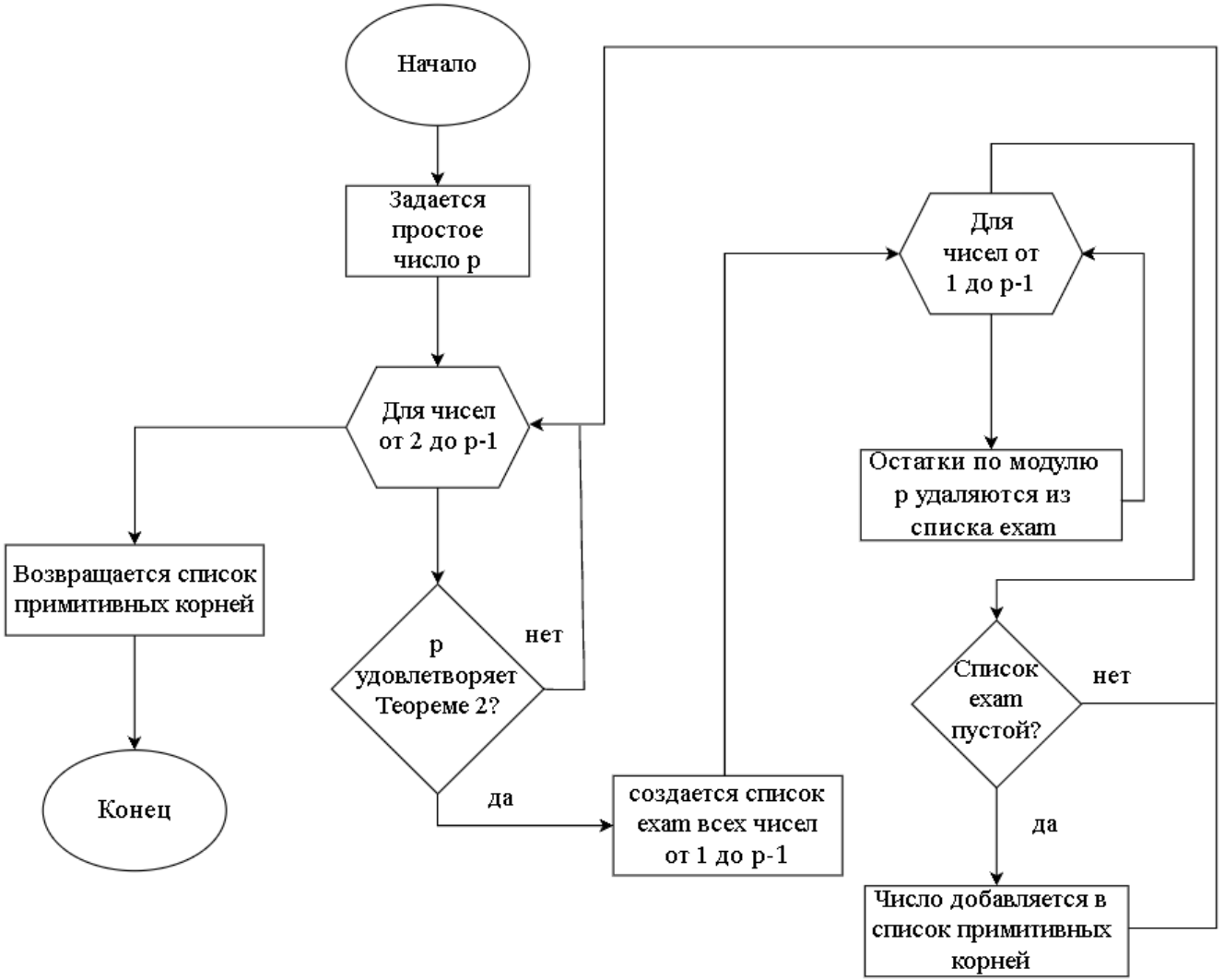
Используемые библиотеки:

Tkinter 8.6.13

АЛГОРИТМ НАХОЖДЕНИЯ ВСЕХ ПРОСТЫХ ЧИСЕЛ ИЗ ДИАПАЗОНА

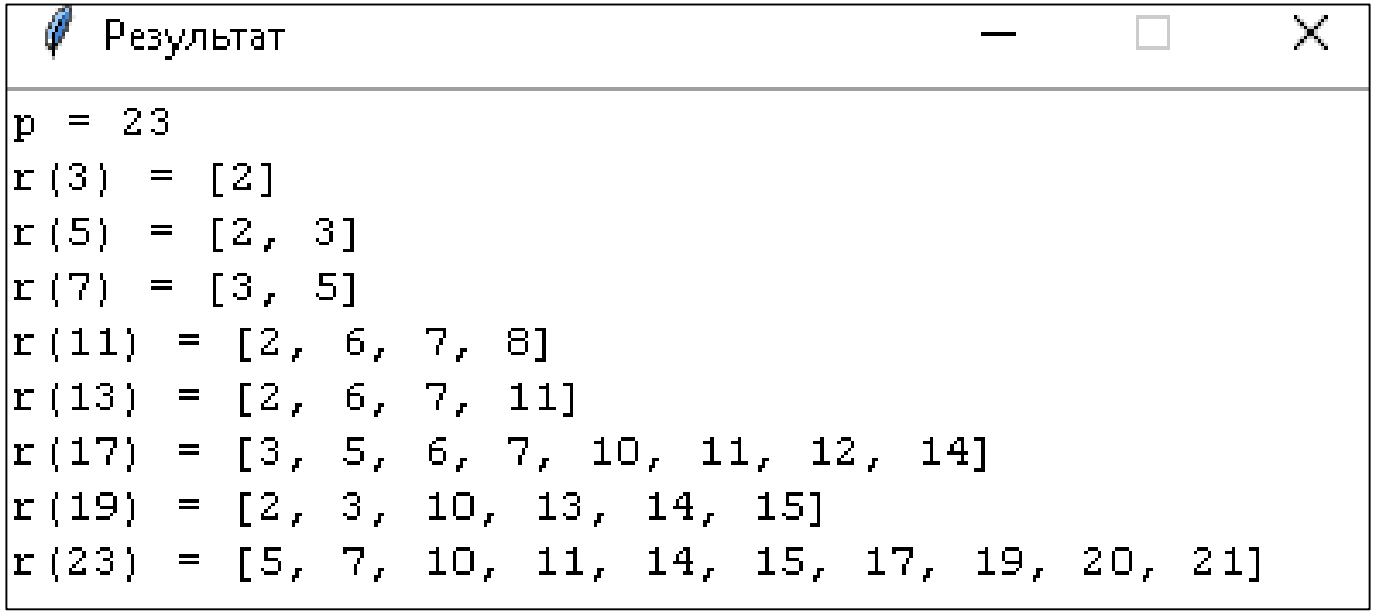


АЛГОРИТМ НАХОЖДЕНИЯ ПРИМИТИВНЫХ КОРНЕЙ ДЛЯ ПРОСТЫХ ЧИСЕЛ ИЗ ДИАПАЗОНА



ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ГЛАВНОГО ЭКРАНА СИСТЕМЫ

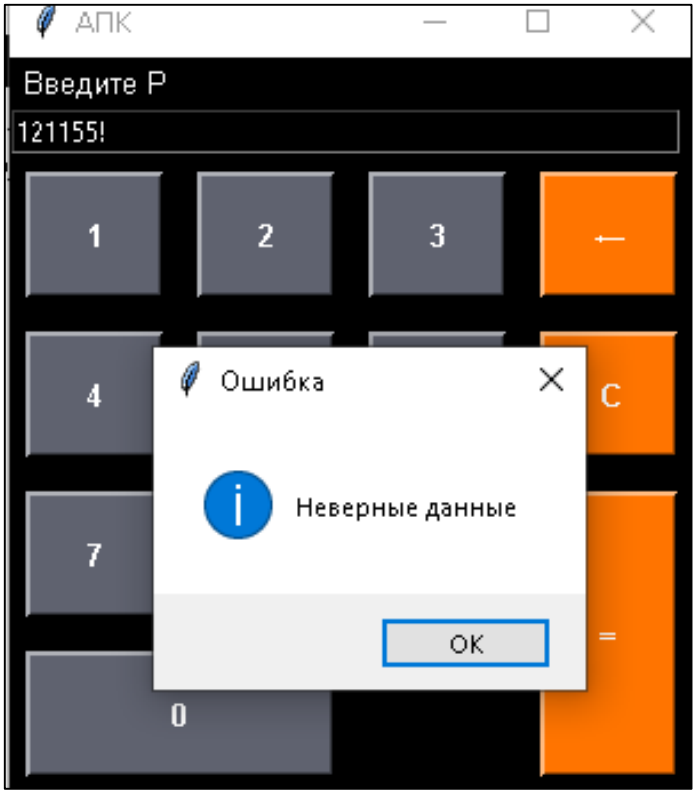
ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ЭКРАНА ПОСЛЕ ВЫЧИСЛЕНИЙ



ТЕСТИРОВАНИЕ

1. Было проведено 5 функциональных тестов для проверки программной системы функциональным требованиям.
2. Также были проведены тесты на корректность входных и выходных данных.

ТЕСТИРОВАНИЕ НА КОРРЕКТНОСТЬ ВХОДНЫХ ДАННЫХ



ТЕСТИРОВАНИЕ НА КОРРЕКТНОСТЬ ВЫХОДНЫХ ДАННЫХ

№	Вывод системы	Ожидаемый результат	Тест
1	Ввод: [23] Вывод: r(3) = [2] r(5) = [2, 3] r(7) = [3, 5] r(11) = [2, 6, 7, 8] r(13) = [2, 6, 7, 11] r(17) = [3, 5, 6, 7, 10, 11, 12, 14] r(19) = [2, 3, 10, 13, 14, 15] r(23) = [5, 7, 10, 11, 14, 15, 17, 19, 20, 21]	Ввод: [23] Вывод: r(3) = [2] r(5) = [2, 3] r(7) = [3, 5] r(11) = [2, 6, 7, 8] r(13) = [2, 6, 7, 11] r(17) = [3, 5, 6, 7, 10, 11, 12, 14] r(19) = [2, 3, 10, 13, 14, 15] r(23) = [5, 7, 10, 11, 14, 15, 17, 19, 20, 21]	Пройден
2	Ввод: [17] Вывод: r(3) = [2] r(5) = [2, 3] r(7) = [3, 5] r(11) = [2, 6, 7, 8] r(13) = [2, 6, 7, 11] r(17) = [3, 5, 6, 7, 10, 11, 12, 14]	Ввод: [17] Вывод: r(3) = [2] r(5) = [2, 3] r(7) = [3, 5] r(11) = [2, 6, 7, 8] r(13) = [2, 6, 7, 11] r(17) = [3, 5, 6, 7, 10, 11, 12, 14]	Пройден
3	Ввод: [7] Вывод: r(3) = [2] r(5) = [2, 3] r(7) = [3, 5]	Ввод: [7] Вывод: r(3) = [2] r(5) = [2, 3] r(7) = [3, 5]	Пройден

ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

№	Цель теста	Действия	Ожидаемый результат	Тест пройден?
1	Запустить приложение	Ввести в окошке число P и нажать «=»	Открылось окно с результатами числа P	Да
2	Проверить функционал кнопок калькулятора	Нажать на кнопки от 1 до 9, кнопку «C» и «←»	При нажатии цифр в окно вводятся числа, а при нажатии «C» удаляются все числа и «←» удаляется последнее число.	Да
3	Ввести число, у которого нету примитивных корней	Ввести в окошке число P и нажать «=»	Открылось окно для заданного числа, где написано нету примитивных корней	Да
4	Ввести большое простое число	Ввести в окошке большое простое число P и нажать «=»	Благополучно открылось окно с результатами числа P	Да
5	Ввести буквы, знаки препинания, восклицания и др.	Ввести в окошке буквы, знаки препинания, восклицания и др.	Открылось окно ошибка введите число	Да

АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

1. Были проведены вычисления для диапазонов с большими значениями.
2. Найдены все примитивные корни простых чисел до 1000.
3. На основе полученных данных составлены таблицы и построены графики.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Составлен список простых чисел из диапазона [2..P], где P - натуральное число.

2. Разработана функция для вычисления примитивных корней по модулю простых чисел из диапазона [2..P] для заданного числа P.

3. Разработана функция, которая будет находить для каждого примитивного корня количества простых чисел из диапазона [2..P], для которых это число будет примитивным корнем.

4. Создано графическое приложение, в которое включены все указанные функции.