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

РАБОТА ПРОВЕРЕНА

Рецензент
Доцент кафедры компьютерной
топологии и алгебры
ФГБОУ ВО «ЧелГУ», к.ф.-м.н.

_____ О.В. Митина

«___»_____ 2024 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«___»_____ 2024 г.

Разработка программной системы для изучения круговых единиц для простых чисел

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.04.02.2024.308-1398.ВКР

Научный руководитель,
профессор кафедры СП, д.ф.-м.н.,
доцент
_____ Р.Ж. Алеев

Автор работы,
студент группы КЭ-220
_____ Я.П. Романов

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
«___»_____ 2024 г.

Челябинск, 2024 г.

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

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
29.01.2024 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы магистранта

студенту группы КЭ-220
Романову Ярославу Петровичу,
обучающемуся по направлению
02.04.02 «Фундаментальная информатика и информационные технологии»
(магистерская программа «Машинное обучение и анализ больших данных»)

1. **Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка программной системы для изучения круговых единиц
для простых чисел.

2. **Срок сдачи студентом законченной работы:** 20.05.2024 г.

3. **Исходные данные к работе**

3.1. Виноградов, И. М. Основы теории чисел: учебное пособие. // Санкт-Петербург: Лань, 2009. – 176 с.

3.2. Нестеренко, Ю.В. Теория чисел: учебник для студ. высш. учеб. заведений. // М.: Издательский центр «Академия», 2008 – 292 с.

3.3. The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.12.2. [Электронный ресурс] URL: https://www.gap-system.org (дата обращения: 26.02.2024 г.).

4. Перечень подлежащих разработке вопросов

4.1. Изучить предметную область.

4.2. Провести вычисления с круговыми единицами.

4.3. Определить функциональные и нефункциональные требования к системе.

4.4. Разработать и реализовать алгоритмы расчетов.

4.5. Определить способ вывода данных.

4.6. Тестирование системы.

5. Дата выдачи задания: 29.01.2024 г.

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

Задание принял к исполнению Я.П. Романов

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность

Теория чисел является одной из старейших и наиболее фундаментальных областей математики. Она изучает свойства целых чисел, простых чисел, арифметические операции и их взаимосвязи. Теория чисел имеет широкий спектр прикладных применений и является ключевой для многих областей науки и техники.

Далее перечислены несколько областей, в которых теория чисел актуальна.

1. Криптография. Теория чисел используется в криптографии для шифрования информации, защиты данных и обеспечения конфиденциальности. Например, RSA-алгоритм основан на теории чисел и используется для создания криптосистем с открытым ключом.
2. Компьютерная безопасность. Теория чисел играет важную роль в области компьютерной безопасности, особенно при разработке алгоритмов шифрования, аутентификации и защиты информации.
3. Математическое моделирование. Теория чисел используется для создания математических моделей и алгоритмов, которые широко применяются в научных и инженерных расчетах.
4. Финансовая математика. Теория чисел играет важную роль в финансовой математике при анализе финансовых рынков, оценке рисков и разработке финансовых инструментов.
5. Машинное обучение и искусственный интеллект. Теория чисел используется в разработке алгоритмов машинного обучения и искусственного интеллекта для обработки данных, классификации и прогнозирования.

Таким образом, изучение и исследование теории чисел в рамках выпускной квалификационной работы может быть полезно как для академического роста, так и для развития профессиональных навыков в областях современной науки и техники.

Постановка задачи

Целью выпускной квалификационной работы является разработка программной системы для изучения круговых единиц для простых чисел. Для которой необходимо выполнить вычисления с простыми числами. Во-первых, найти следы степеней круговых единиц, которые сравнимы с 1 по модулю простого числа. Во-вторых, вычислить следы степеней круговых единиц по модулям 2 и 3. И наконец, найти степени, которые удовлетворяют предыдущим двум условиям. Для достижения поставленной цели необходимо решить следующие задачи:

1) изучение предметной области;
2) провести вычисления;
3) определение функциональных и нефункциональных требований к системе;
4) разработка и реализация алгоритмов расчетов;
5) определение способа вывода данных;
6) тестирование системы.

Структура и содержание работы

Работа состоит из введения, семи глав, заключения и списка литературы. Объем работы составляет 40 страниц, объем списка литературы – 20 источников.

В первой главе проводится обзор предметной области.

Во второй главе описываются основные теоретические сведения.

Третья глава посвящена основным функциям GAP.

В четвертой главе описаны основные вычисления.

Пятая глава показывает процесс проектирования системы и разработки алгоритмов, функциональные и нефункциональные требования.

Шестая глава посвящена реализацию алгоритмов.

В седьмой главе проходит тестирование системы.

В приложениях содержатся блок-схемы алгоритмов и результаты расчетов.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Учебное пособие «Основы теории чисел» И.М. Виноградова [1] является классическим и авторитетным изданием в области теории чисел. В книге автор представляет основные принципы и понятия теории чисел, написанные доступным и понятным языком для студентов и любителей математики.

В учебнике И.М. Виноградова рассматриваются следующие темы.

1. Основные понятия и определения в теории чисел.
2. Теоремы о делении и остатках.
3. Основные теоремы арифметики.
4. Теорема о простых числах.
5. Теория делимости и членство в множествах.
6. Решение диофантовых уравнений.

В книге также присутствуют примеры решения задач, упражнения для самостоятельной работы и проверки знаний, что делает материал более доступным и интересным для читателя. И.М. Виноградов в своем пособии стремится обеспечить читателя полным и систематическим изложением информации по теории чисел, что позволяет использовать книгу как один из основных источников знаний по этой теме.

Учебник «Теория чисел» под редакцией Ю.В. Нестеренко [2] является важным источником информации для студентов высших учебных заведений, изучающих эту область математики. Он издан в 2008 году и содержит обширный материал по основам теории чисел, включая такие темы, как простые числа, арифметические функции, кольца целых чисел, диофантовые уравнения и другие.

В учебнике представлены теоретические сведения и практические задачи, которые помогут студентам углубить свои знания в данной области математики и научиться применять их на практике. Книга также может быть полезна для преподавателей, исследователей и всех, кто интересуется теорией чисел.

Ю.В. Нестеренко известен своими работами в области теории чисел, и его учебник является авторитетным источником информации по данной теме.

The GAP Group предоставляет доступ к программе GAP (Groups, Algorithms, and Programming) [3], которая является мощным инструментом для исследования структуры алгебраических объектов, таких как группы, кольца и поля. На сайте представлена информация о версиях программы, а также документация, примеры использования и ссылки на дополнительные ресурсы. GAP распространяется свободно, и вся поддержка пользователей осуществляется авторами программы.

Кроме того, на сайте представлены различные руководства и учебные материалы, которые помогут пользователям лучше понять функциональность программы и научиться использовать ее в своей работе. Также на сайте доступны ссылки на сообщество пользователей, форумы обсуждений, а также возможность задать вопросы и получить помощь от других пользователей или разработчиков программы.

Для студентов, исследователей и преподавателей, занимающихся математикой и работающих с группами и алгоритмами, программа GAP может быть ценным инструментом для проведения исследований, выполнения вычислений и анализа алгебраических объектов.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Определение. Целые числа k и l сравнимы по натуральному модулю m, если $k - l$ делится на m. Сравнимость чисел обозначается, как число $k \equiv l \ (mod\ m)$. Сравнимость по модулю 2 означает, что, либо оба числа четные, либо оба нечетные.

Если n и m – взаимно простые натуральные числа, то мультипликативный порядок числа n по модулю m – это такое наименьшее натуральное число i, при котором выполняется условие (1):

$$n^i \equiv 1 \ (mod\ m). \tag{1}$$

Пусть n и m – натуральные числа. Число n называется примитивным первообразным корнем по модулю m, если для любого натурального числа k, взаимно простого с m, существует такое натуральное число j_k, что выполнятся условие (2):

$$n^{j_k} \equiv k \ (mod\ m). \tag{2}$$

Теорема 1 (Гаусс). Примитивные корни существуют для любого простого модуля [4].

Пусть n – натуральное число.

1. Комплексное число α, называется корнем из единицы степени n, если $\alpha^n = 1$.

2. Корень из единицы степени n называется примитивным (или первообразным), если $\alpha^k \neq 1$ для любого $k \in \{1, 2, \dots, n-1\}$.

Примеры.

1. Все корни из единицы степени 4 – $\{1, i, -1, i\}$. Примитивные корни только – $\{i, -i\}$.

2. Для любого натурального числа n множество всех корней из единицы степени n равно формуле (3):

$$\left\{ e^{\frac{2k\pi i}{n}} = \cos\frac{2k\pi}{n} + i\sin\frac{2k\pi}{n} \,\middle|\, k \in \{0,1,2, \dots, n-1\} \right\}. \tag{3}$$

Любой примитивный корень степени n соответствует формуле (4):

$$e^{\frac{2\pi i}{n}} = \cos\frac{2\pi}{n} + i\sin\frac{2\pi}{n}. \quad (4)$$

Пусть p – простое число, g – примитивный корень по модулю p и α – примитивный корень из единицы степени p. Круговой единицей для примитивного корня g называется комплексное число, показанное на формуле (5):

$$1 + \alpha + \cdots + \alpha^{g-1}. \quad (5)$$

Примеры.

1. Круговой единицей для примитивного корня 2 будет: $1 + \alpha$.
2. Круговой единицей для примитивного корня 3 будет: $1 + \alpha + \alpha^2$.

Пусть p – простое число и α – примитивный корень из единицы степени p. Также пусть λ соответствует условию (6):

$$\lambda = l_0 * 1 + l_1\alpha + l_2\alpha^2 + \cdots + l_{p-1}\alpha^{p-1}, \quad (6)$$

где для любого $j \in \{0,1,\dots,p-1\}$ коэффициент l_j – рациональное число.

Следом комплексного числа λ называется рациональное число, показанное на формуле (7):

$$l_0(p-1) - l_1 - l_2 - \cdots - l_{p-1}. \quad (7)$$

Примеры. Пусть, g – примитивный корень по модулю p.

1. След круговой единицы для примитивного корня g рассчитывается с помощью формулы (8):

$$(p-1) - 1 - 1 - \cdots - 1 = (p-1) - (g-1) = p - g\,. \quad (8)$$

2. След круговой единицы для примитивного корня 2 будет: $p - 2$.
3. След круговой единицы для примитивного корня 3 будет: $p - 3$.

3. СВЕДЕНИЯ ИЗ GAP

GAP – это система компьютерной дискретной алгебры, которая акцентирует внимание на теории вычислительных групп. GAP предлагает язык программирования и библиотеку из множества функций, которые выполняют алгебраические алгоритмы, написанные на языке GAP, а также содержит обширные библиотеки данных алгебраических объектов. GAP используется в исследованиях и обучении для изучения групп, их представлений, колец, векторных пространств, алгебр, комбинаторных структур и многого другого. Система, включая исходный код, доступна бесплатно. Вы можете изучать, легко изменять или расширять ее в соответствии с вашими конкретными целями.

GAP имеет ядро, написанное на C. Оно реализует:

1) язык GAP;

2) интерактивную среду для разработки и использования программ GAP;

3) управление памятью и быстрые версии критичных ко времени операций для различных типов данных.

Остальные функции библиотеки написаны на языке GAP. Большинство пакетов также написаны на GAP, но некоторые из них имеют автономные версии. Некоторые пакеты содержат ссылки на другие системы [5].

При вычислениях были использованы встроенные функции языка GAP.

`LcmInt(m, n)` (функция) возвращает наименьшее общее кратное целых чисел m и n.

`OrderMod(n, m[,bound])` (функция) возвращает порядок умножения целого числа n по модулю положительное целое число m. Если n и m не взаимно просты, порядок n не равен определено, а `OrderMod` вернет 0. Если задана неверная граница (bound), результат будет неверным.

`PrimitiveRootMod(m, [,start])` – функция возвращает наименьший примитивный корень по модулю целого числа m и ошибку (`fail`), если такого первообразного корня не существует. Если задается необязательный второй целочисленный аргумент `start`, `PrimitiveRootMod` возвращает наименьший примитивный корень, строго превышающий `start`.

`IsPrimitiveRootMod(r, m)` – функция возвращает `true`, если целое число r является примитивным корнем по модулю положительного целого числа m и `false` в противном случае. Если p меньше 0 или больше, чем m, он заменяется своим остатком.

`E(n)` (операция) возвращает примитивный n-й корень из единицы $e_n = \exp(\frac{2\pi i}{n})$.

`IsIntegralCyclotomic(obj)` – принадлежность циклотомические целые числа – это те круговые многочлены, для которых внешнее представление представляет собой список целых чисел.

`Conductor(cyc)` – атрибут: для элемента `cyc` кругового многочлена `Conductor` возвращает наименьшее целое число n такое, что `cyc` содержится в n-м круговом поле.

`CoeffsCyc(cyc,N)` – функция, возвращающая коэффициенты. Пусть `cyc` – круговой многочлен с корнем n. Если N не кратно n, то `CoeffsCyc` возвращает ошибку, потому что `cyc` не может быть выражен через корни N-й степени из единицы. В противном случае `CoeffsCyc` возвращает список длиной N.

4. ОСНОВНЫЕ ВЫЧИСЛЕНИЯ С ПОМОЩЬЮ GAP

Пример вычислений будет рассматриваться на примере простого числа 11. Первым действием вычисляется наименьший примитивный корень при помощи функции `PrimitiveRootMod`. После чего вычисляется степени круговых единиц сравнимые с единицей по модулю p с помощью функции `E`. Вышеупомянутые вычисления продемонстрированы на рисунке 1.

```
gap> p:=11;
11
gap> g:=PrimitiveRootMod(p);
2
gap> c:=1+E(p);
-E(11)^2-E(11)^3-E(11)^4-E(11)^5-E(11)^6-E(11)^7-E(11)^8-E(11)^9-E(11)^10
```

Рисунок 1 – Создание переменных

Проверка, что след числа $-c^{(p-1)/2}$ сравним с $p-1$ по модулю p с помощью функции `Trace`, приведена на рисунке 2.

```
gap> c^((p-1)/2);
4*E(11)+9*E(11)^2+9*E(11)^3+4*E(11)^4-E(11)^6-E(11)^7-E(11)^8-E(11)^9-
E(11)^10
gap> Trace(c^((p-1)/2));
-21
gap> Trace(c^((p-1)/2)) mod p;
1
```

Рисунок 2 – Проверка

Порядки 2 и 3 по модулю p, рассчитанные с помощью функции `OrderMod`, показаны на рисунке 3.

```
gap> f2:=OrderMod(2,p);
10
gap> f3:=OrderMod(3,p);
5
```

Рисунок 3 – Порядки 2 и 3

Эти числа обеспечивают делимость на 2 и 3 коэффициентов чисел $\pm c^k - 1$. Далее необходимо найти следы степеней круговых единиц по модулям 2 и 3. После чего проверить степени, соответствующие следам сте-

пени круговых единиц сравнимые с 1 по модулю p и следам степеней круговых единиц по модулям 2 и 3.

Сначала рассчитываются следы степеней круговых единиц по модулю 2. Для этого необходимо вычислить $2^{f2} - 1$ и рассчитать коэффициенты кругового многочлена с помощью функции `CoeffsCyc` (рисунок 4).

```
gap> st:=2^f2-1;
1023
gap> lu2:=CoeffsCyc(c^st-1,p) mod 2;
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

Рисунок 4 – Коэффициенты кругового многочлена

Необходимо уменьшить число st. Для этого его нужно разложить на простые множители при помощи функции `FactorsInt` и проверить без какого из них коэффициенты проходят проверку (рисунок 5).

```
gap> FactorsInt(st);
[ 3, 11, 31 ]
gap> st3:=st/3;
341
gap> st11:=st/11;
93
gap> st31:=st/31;
33
gap> lu2:=CoeffsCyc(c^st2-1,p) mod 2;
Error, Variable: 'st2' must have a value
not in any function at *stdin*:15
gap> lu2:=CoeffsCyc(c^st3-1,p) mod 2;
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> lu2:=CoeffsCyc(c^st11-1,p) mod 2;
[ 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1 ]
gap> lu2:=CoeffsCyc(c^st31-1,p) mod 2;
[ 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0 ]
```

Рисунок 5 – Уменьшение числа st

Из полученных результатов очевидно, что подходит только $st3$. Далее вычисляется наименьшее общее кратное чисел $(p-1)/2$ и $st3$ с помощью функции `LcmInt` и проводится проверка (рисунок 6).

```
gap> ep2:=LcmInt((p-1)/2,st3);
1705
gap> tr:=Trace(c^ep2) mod p;
1
gap> tr:=Trace(-c^ep2) mod p;
10
gap> l2:=CoeffsCyc(c^ep2-1,p) mod 2;
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> l2:=CoeffsCyc(-c^ep2-1,p) mod 2;
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

Рисунок 6 – Наименьшее общее кратное и проверка

Подходит только $-c^{ep2}$. Далее вычисляются следы степеней круговых единиц для порядка 3. Для этого сначала вычисляется $3^{f3}-1$ коэффициенты кругового многочлена (рисунок 7).

```
gap> pow:=3^f3-1;
242
gap> lu3:=CoeffsCyc(c^pow-1,p) mod 3;
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> lu3:=CoeffsCyc(-c^pow-1,p) mod 3;
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
gap> lu3:=CoeffsCyc(-c^pow-1,p) mod 3;
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

Рисунок 7 – Вычисления для порядка 3

Для уменьшения числа pow, его необходимо разложить на простые множители, после чего pow нужно разделить на каждый уникальный простой множитель и проверить какое новое число проходит проверку (рисунок 8).

```
gap> FactorsInt(pow);
[ 2, 11, 11 ]
gap> pow2:=pow/2;
121
gap> pow11:=pow/11;
22
gap> lu3:=CoeffsCyc(c^pow2-1,p) mod 3;
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
gap> lu3:=CoeffsCyc(-c^pow2-1,p) mod 3;
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> lu3:=CoeffsCyc(c^pow11-1,p) mod 3;
[ 0, 2, 1, 0, 0, 2, 2, 0, 0, 1, 2 ]
gap> lu3:=CoeffsCyc(-c^pow11-1,p) mod 3;
[ 0, 0, 1, 2, 2, 0, 0, 2, 2, 1, 0 ]
```

Рисунок 8 – Уменьшение числа pow

Из полученных результатов очевидно, что подходит число $pow2$. Вычисляется наименьшее общее кратное чисел $(p-1)/2$ и $pow2$ и проводится проверка на рисунке 9.

```
gap> ep3:=LcmInt((p-1)/2,pow2);
605
gap> tr3:=Trace(-c^ep3) mod p;
10
gap> tr3:=Trace(c^ep3) mod p;
1
gap> l3:=CoeffsCyc(-c^ep3-1,p) mod 3;
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> l3:=CoeffsCyc(c^ep3-1,p) mod 3;
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

Рисунок 9 – Проверка

Полученные результаты показывают, что подходит число $-c^{ep3}$. Часть расчетов приведена в таблице 1, где p – это простое число, g – примитивный корень, f2 и f3 – порядки 2 и 3 по модулю p соответственно, колонки 2 и 3 – результат для порядков 2 и 3 соответственно.

Таблица 1 – Часть расчетов

p	**g**	**след**	**f2**	**f3**	**F2**	**F3**	**ep2**	**ep3**	**2**	**3**
5	2	1	4	4	15	40	30	40	-c	c
7	3	6	3	6	7	364	21	1092	c	c
11	2	1	10	5	341	121	1705	605	-c	-c
13	2	1	12	13	819	26	1638	78	-c	-c
17	3	16	8	16	255	223040	2040	223040	c	c
19	2	1	18	18	87381	387420488	87381	3486784392	-c	-c
23	5	22	11	11	2047	88573	22517	974303	c	c
29	2	1	28	28	2375535	1387060720	33257490	9709425040	-c	-c

5. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

5.1. Анализ требований

Программная система – это приложение, которые получает на вход простое положительное число, и исходя из этого пользователь может предоставить результаты вычисления примитивного корня, круговой единицы и свойства порядков 2 и 3 для этого числа. В результате проведенного анализа для системы были выделены следующие функциональные требования.

1. Пользователю доступен ввод простого числа.
2. Система должна проверить корректность ввода данных.
3. Система должна выводить результаты вычислений, исходя из полученного числа.
4. Пользователь может сохранить полученные результаты.

Также были выделены следующие нефункциональные требования.

1. Система должна поддерживаться ОС Windows.
2. Данные вычислений должны храниться в формате .csv.
3. Система должна быть реализована с помощью языка Python.
4. Система должна сохранять данные.

5.2. Диаграмма вариантов использования

На рисунке 10 представлена диаграмма вариантов использования проектируемой программной системы. Представление системы в виде диаграммы вариантов использования позволяет понять, как будет она функционировать [6].

Диаграмма вариантов использования показывает, как будет происходить взаимодействие между системой и пользователем, и какие функции в ней реализованы.

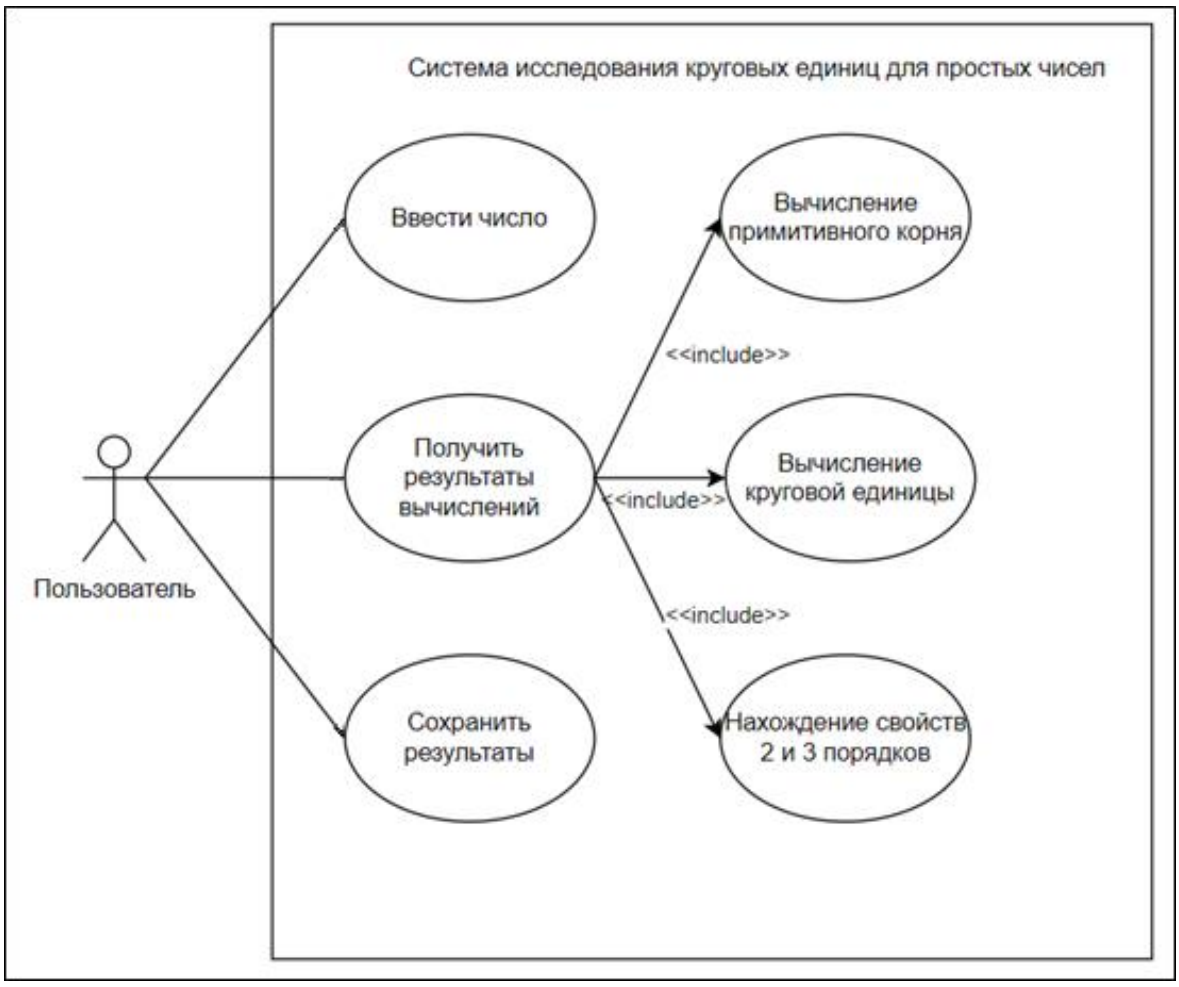

Рисунок 10 – Диаграмма вариантов использования

Актером системы является пользователь. Пользователь может ввести число и получить соответсвующие ему результаты. Пользователь может выбрать какие результаты ему необходимы. Также ему доступно сохранение полученных вычислений.

5.3. Алгоритмы вычислений

Для разработки алгоритма вычисления примитивного корня g по модулю числа p, указанного в формуле (2), необходимо найти такое число, что все его степени по указанному модулю пробегают по всем числам взаимно простыми с этим модулем. В случае простого числа пробегают по всем числам от 1 до $p-1$.

Согласно теореме Гаусса [7] первообразный корень по модулю p существует для любого простого модуля.

Для нахождения корня будет использоваться функция Эйлера. Функция Эйлера [8] – это количество чисел от 1 до p, взаимно простых с p, то есть количество таких чисел в промежутке $[1; p]$, наибольший общий делитель которых с p равен одному. С помощью этой функции можно показать, что для примитивного корня по модулю p наименьшее i из формулы (1) равно значению функции Эйлера от числа p ($\varphi(p)$).

Из теоремы Лагранжа [9] следует, что показатель любого числа по модулю p является делителем $\varphi(p)$. Таким образом достаточно проверить, что для всех собственных делителей $d|\varphi(p)$ выполняется условие (9):

$$g^d ! \equiv 1 \ (mod\ p). \tag{9}$$

Факторизируется число, как показано на формуле (10):

$$\varphi(p) = p_1^{\alpha_1} \dots p_s^{\alpha_s}. \tag{10}$$

Можно доказать, что достаточно рассматривать в качестве d лишь те числа, вид которых $\frac{\varphi(p)}{p_i}$. Пусть d – произвольный собственный делитель $\varphi(p)$. Тогда найдется такое j, при котором выполняется условие (11):

$$d * k = \frac{\varphi(p)}{p_j}. \tag{11}$$

Однако, если выполняется формула (12):

$$g^d \equiv 1 \ (mod\ p), \tag{12}$$

то выходит формула (13):

$$g^{\frac{\varphi(p)}{p_j}} \equiv (g^d)^k \equiv 1^k \equiv 1 \ (mod\ p), \tag{13}$$

то есть все равно среди чисел вида $\frac{\varphi(p)}{p_i}$ нашлось бы такое, что условие бы не выполнилось, что и требовалось доказать.

Таким образом, составляется алгоритм нахождения примитивного корня. Находится число $\varphi(p)$, после чего оно факторизируется. После чего

перебираются все числа g от 1 до p, и для каждого считаются величины, как показано на формуле (14):

$$g^{\frac{\varphi(p)}{p_j}} \ (mod\ p). \tag{14}$$

Если для текущего g все эти числа оказались отличными от 1, то это число и есть искомый примитивный корень.

На рисунке 1 приложения А изображен алгоритм нахождения примитивного корня в виде блок-схемы.

Для вычисления корня было использовано бинарное возведение в степень по модулю [10]. Такой прием позволяет уменьшить количество умножений с $O(n)$ до $O(logn)$.

Для возведения любого числа n в четную степень p, достаточно возвести в степень $p/2$ и возвести полученное число в квадрат. Если же p не четно, то степень уменьшается на единицу и в конце дополнительно умножается на n.

Чтобы выполнить указанные расчеты, число p представляется в двоичном виде, тогда результат можно представить как произведение числа n в степенях степеней числа два. Для подсчета необходимо пройти по всем битам p. В начале результат равен единице. На каждом шаге необходимо домножить n^{2^b} на текущий результат, где b номер текущей итерации, если b-тый бит. Также нужно возводить в квадрат число n^{2^b}. Алгоритм показан на рисунке 2 приложения А.

Для поиска свойств круговой единицы может потребоваться поиск простых множителей числа [11]. Для поиска списка всех таких множителей, проверяется делимость данного числа на натуральные числа, начиная с 2. Если находится такой делитель, то данное число делится на него, пока его можно разделить целочисленно и добавлять его в список. Данный алгоритм представлен на рисунке 3 приложения А.

6. РЕАЛИЗАЦИЯ ПРОГРАММНОЙ СИСТЕМЫ

Реализация системы будет реализован на языке Python. Python – это высокоуровневый язык программирования, который отличается простотой и чистотой синтаксиса [12]. Он подходит для разработки различных приложений, начиная от веб-сервисов и мобильных приложений, до научных вычислений и машинного обучения.

Python поддерживает различные парадигмы программирования, включая объектно-ориентированное, функциональное и процедурное программирование. Благодаря богатым библиотекам и фреймворкам, Python стал популярным выбором для разработки приложений в различных областях [13].

Основные преимущества использования Python [14].

1. Простота и удобство использования. Python имеет понятный и лаконичный синтаксис, что делает его отличным выбором для начинающих программистов.

2. Большое количество библиотек. Python имеет огромное количество библиотек, которые позволяют расширить его функциональность и упростить разработку приложений.

3. Поддержка широкого спектра задач. Python поддерживает различные области программирования, начиная от разработки веб-приложений до научных исследований и анализа данных.

4. Активное сообщество. Python имеет большое и активное сообщество разработчиков, которые готовы помочь другим и делиться знаниями.

У языка Python есть различные инструменты и технологии в зависимости от поставленных целей и задач. Например, для разработки веб-приложений можно использовать фреймворки Django или Flask, для работы с данными – библиотеки pandas, NumPy и matplotlib [15].

В целом, использование Python позволяет создать качественное и функциональное приложение с минимальными затратами времени и усилий.

6.1. Реализация алгоритма разбиения на простые множители

Алгоритм реализован на языке Python. Для реализации были использованы такие типы данных, как int32 и set (целые числа и множества) [16].

Для раскладывания числа на множители необходимо последовательно проверять его делимость на натуральные числа, начиная с числа 2. Если число делится на какой-либо делитель, то его нужно поделить на этот делитель и добавить его в список простых множителей.

Если после выполнения этого алгоритма число остается не равным 1, то оставшееся значение также является простым множителем и его необходимо добавить в список. Реализация этого алгоритма представлена на рисунке 11 и соответствует функции Эйлера.

```
def findPrimefactors(s, n):
    while (n % 2 == 0):
        s.add(2)
        n = n // 2
    for i in range(3, int(sqrt(n)), 2):
        while (n % i == 0):
            s.add(i)
            n = n // i
    if (n > 2):
              s.add(n)
```

Рисунок 11 – Реализация алгоритма разбиения на простые множители

Функция findPrimefactors принимает два аргумента: множество s и число n. Внутри функции происходит поиск простых множителей числа n.

Сначала проверяется делится ли число n на 2 без остатка. Если да, то число 2 добавляется во множество s, и n делится на 2.

Затем в цикле происходит проверка деления числа n на нечетные числа от 3 до корня из n. Если число n делится на i без остатка, то число i добавляется во множество s и n делится на i.

Если после прохода цикла надо остались простые множители, то они также добавляются во множество s.

Таким образом, во множество s добавляются все простые множители числа n.

Результат вывода реализации алгоритма показан на рисунке 12.

```
C:\Users\User\AppData\Local\Program
Prime factors of  88
{2, 11}

Process finished with exit code 0
```

Рисунок 12 – Результат алгоритма разбиения на простые множители

6.2. Реализация алгоритм вычисления примитивного корня

Алгоритм реализован на языке Python. Для реализации были использованы такие типы данных, как int32, bool и set.

Для данный алгоритма требуется проверить является ли число простым. Данное действие выделено в отдельную функцию, реализация которой представлена на рисунке 13. В начале проверяется, является ли число меньше 1, чтобы сразу вычеркнуть все отрицательные чилсла, ноль и единицу. Сначала проверяются числа 2 и 3. Далее проверяется число на делимость на 2 и 3, чтобы пропускать их. У каждого составного числа есть собственный делитель, не превосходящий квадратного корня этого числа.

```
def isPrime(n):
    if (n <= 1):
        return False
    if (n <= 3):
        return True
    if (n % 2 == 0 or n % 3 == 0):
        return False
    i = 5
    while (i * i <= n):
        if (n % i == 0 or n % (i + 2) == 0):
            return False
        i = i + 6
    return True
```

Рисунок 13 – Реализация функции isPrime

Функция isPrime принимает на вход число n и проверяет, является ли оно простым числом. Если число меньше или равно 1, то функция

возвращает False. Если число меньше или равно 3, то возвращает True. Затем проверяется делится ли число на 2 или 3 без остатка, и если да, то возвращается False.

Далее устанавливается начальное значение переменной i равное 5 и в цикле while проверяется условие, пока i в квадрате меньше или равно n. Внутри цикла проверяется делится ли число на i и $i + 2$ без остатка, и если да, то возвращается False.

Если после прохождения всех проверок число не делится на какие-либо числа от 5 до корня из n, то возвращается True, что означает, что число является простым. Результат выполнения функции isPrime для разных чисел показан на рисунке 14.

```
Is 5 prime: True
Is 10 prime: False
Is 12 prime: False
Is 13 prime: True
Is 103 prime: True
Is 88 prime: False

Process finished with exit code 0
```

Рисунок 14 – Результат выполнения функции isPrime

Также необходима функция возведения в степень по модулю. Реализация показана на рисунке 15. Создается переменная для хранения результата и считаем модуль числа. Далее побитно перемножается число.

```
def power(x, y, p):
    res = 1
    x = x % p
    while (y > 0):
        if (y & 1):
            res = (res * x) % p
        y = y >> 1
        x = (x * x) % p
    return res
```

Рисунок 15 – Реализация функции power

Функция power принимает три аргумента: x, y и p. Внутри функции создается переменная res, которая устанавливается в 1. Затем переменная x приводится к остатку от деления на p.

Далее начинается цикл, который выполняется пока y больше 0. В каждом шаге цикла проверяется, является ли младший бит y равным 1. Если да, то переменная res умножается на x и результат приводится к остатку от деления на p.

Затем y сдвигается на один разряд вправо (эквивалентно делению на 2), а переменная x возводится в квадрат и результат приводится к остатку от деления на p.

После завершения цикла функция возвращает переменную res. Результат функции power показан на рисунке 16.

```
12 in power 5 mod 2 is: 0
15 in power 2 mod 4: 1
88 in power 23 mod 10: 2

Process finished with exit code 0
```

Рисунок 16 – Результат функции power

Для нахождения примитивного корня требуется отдельная функция. В начале необходимо проверить является ли число простым. После чего разбиваем число, уменьшеное на единицу, на простые множители.

Чтобы найти корень используется цикл от 2 до передоваемого числа. Тело цикла будет проверять является ли число текущий итерации примитивный корнем, при успешной проверке цикл прерываем, иначе продолжаем искать. Реализации функции представленая на рисунке 17.

```
def findPrimitive(n):
    s = set()
    if (isPrime(n) == False):
        return -1
    phi = n - 1
    findPrimefactors(s, phi)
    for r in range(2, phi + 1):
        flag = False
        for it in s:
            if (power(r, phi // it, n) == 1):
                flag = True
                break
        if (flag == False):
            return r
    return -1
```

Рисунок 17 – Реализация функции findPrimitive

Функция findPrimitive ищет первообразный корень по модулю n. Сначала создается пустое множество s, затем проверяется, является ли n простым числом. Если нет, возвращается -1.

Далее вычисляется значение функции Эйлера от n (phi), и находятся простые множители числа phi. Затем происходит цикл от 2 до $phi + 1$, в котором для каждого числа r проверяется, обладает ли оно свойством первообразного корня.

Если все проверки прошли успешно, то возвращается значение r. Если ни для одного r не выполняется условие, возвращается -1.

Результат выполнения функции показан на рисунке 18.

```
Smallest primitive root of 11 is 2
Smallest primitive root of 31 is 3
Smallest primitive root of 17 is 3
Smallest primitive root of 19 is 2
Smallest primitive root of 103 is 5
Smallest primitive root of 79 is 3

Process finished with exit code 0
```

Рисунок 18 – Результат нахождения первообрзного корня

6.3. Реализация системы

Для начало необходимо считать данный вычеслений (таблица 1 приложение Б) из файла формата csv [17]. Для этого будем использовать библиотеку pandas. Фрагмент полученных данных показан на рисунке 19.

Библиотека pandas – это инструмент для работы с данными, представленными в виде таблицы [18]. Она предоставляет удобные и эффективные средства для работы с различными типами данных, включая числовые, текстовые, временные ряды и другие.

С помощью pandas можно выполнять множество операций над данными, такие как фильтрация, сортировка, группировка, агрегирование, объединение и многое другое. Библиотека также обладает мощными средствами для чтения и записи данных из различных источников, таких как файлы CSV.

Важными компонентами библиотеки pandas являются два основных типа данных – Series и DataFrame. Series представляет собой одномерный массив данных, а DataFrame – двумерную таблицу данных, состоящую из строк и столбцов. Эти типы данных позволяют удобно и эффективно работать с различными видами информации.

```
0   p  g  trace  f2  f3   F2   F3   ep2   ep3    for2    for3
1   5  2      1   4   4   15   40    30    40  -c^ep2   c^ep3
2   7  3      6   3   6    7  364    21  1092   c^ep2   c^ep3
3  11  2      1  10   5  341  121  1705   605  -c^ep2  -c^ep3
4  13  2      1  12  13  819   26  1638    78  -c^ep2  -c^ep3

Process finished with exit code 0
```

Рисунок 19 – Фрагмент данных считанных из файла

Для реализации системы использовался фреймворк Flask [19]. Фреймворк Flask - это микрофреймворк для веб-приложений на языке программирования Python. Он предоставляет минималистичный набор инструментов для создания веб-приложений.

Flask имеет следующие основные преимущества использования [20].

1. Простота использования. Flask имеет простой и понятный синтаксис, что упрощает процесс разработки веб-приложения для студента без опыта.

2. Гибкость. Flask позволяет выбирать нужные компоненты для вашего приложения и не навязывает жестких правил. Вы можете легко настроить приложение под свои потребности.

3. Расширяемость. В Flask есть множество расширений (extensions), которые добавляют дополнительные функциональные возможности. Это позволяет быстро и легко расширять функциональность вашего веб-приложения.

4. Отличное сообщество. У Flask большое и активное сообщество разработчиков, что обеспечивает удобный доступ к документации, обучающим материалам и поддержке.

5. Подходит для небольших проектов. Flask идеально подходит для создания небольших веб-приложений.

Для сохранения данных будет использоваться база данных. Класс модели базы данных покан на рисунке 20.

```
class PrimeNumber(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    p = db.Column(db.Integer, nullable=False)
    g = db.Column(db.Integer, nullable=False)
    trace = db.Column(db.Integer, nullable=False)
    f2 = db.Column(db.Integer, nullable=False)
    f3 = db.Column(db.Integer, nullable=False)
    F2 = db.Column(db.String(1000), nullable=False)
    F3 = db.Column(db.String(1000), nullable=False)
    ep2 = db.Column(db.String(1000), nullable=False)
    ep3 = db.Column(db.String(1000), nullable=False)
    def __repr__(self):
        return '<PrimeNumber %r>' % self.id
```

Рисунок 20 – Класс модели базы данных

Для более удобного создания веб-приложения используются шаблоны. Шаблоны в фреймворке Flask представляют собой отдельные файлы HTML, которые содержат макеты и структуру визуального

представления веб-страницы. Они позволяют разделять логику и представление, упрощая процесс разработки и облегчая поддержку кода.

Шаблоны Flask используют специальный синтаксис Jinja2 для вставки переменных, условных операторов, циклов и других элементов программирования. Они могут быть подключены к основным Python-скриптам приложения через метод `render_template` и передавать данные из Python-кода в HTML-шаблоны.

Преимущества использования шаблонов на фреймворке Flask включают удобство разделения бизнес-логики и представления, повышение уровня абстракции кода, улучшение читаемости и поддерживаемости сайта, а также возможность быстрого создания дизайна веб-страниц.

Для системы используем шаблон, показанный на рисунке 21.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/main.css') }}">
    {% block head %}{% endblock %}
</head>
<body>
    {% block body %}{% endblock %}
</body>
</html>
```

Рисунок 21 – Шаблон веб-приложения

Подсчет и вывод системы продемонстрирован на рисунке 22.

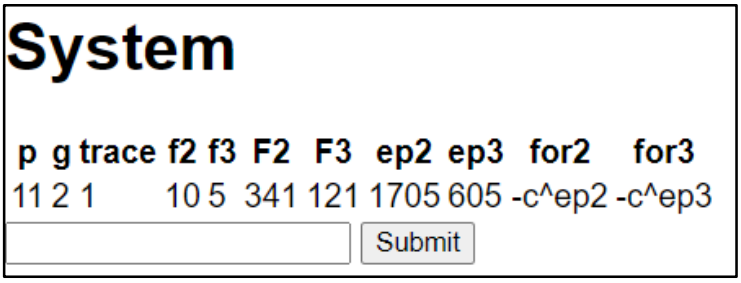

Рисунок 22 – Демонстрация системы

7. ТЕСТИРОВАНИЕ

После разработки приложения было проведено функциональное тестирование в целях проверки степени реализации функциональных требований. Набор функциональных тестов представлен в таблице 2.

Таблица 2 – Тестирование разработанной системы

№	Название теста	Шаги	Ожидаемый результат	Тест пройден
1	Запуск программы без ввода данных	1. Запустить систему. 2. Нажать на кнопку «Submit»	Программа ничего не делает	Да
2	Некорректный ввод данных	1. Запуск программы 2. Ввести текст вместо числа 3. Нажать на кнопку «Submit»	Программа выдает ошибку	Да
3	Ввод отрицательного числа	1. Запуск программы 2. Ввести отрицательное число 3. Нажать на кнопку «Submit»	Программа выдает ошибку	Да
4	Ввод непростого числа	1. Запуск программы 2. Ввести непростое число 3. Нажать на кнопку «Submit»	Программа выдает ошибку	Да
5	Корректный ввод	1. Запуск программы 2. Ввести текст вместо числа 3. Нажать на кнопку «Submit»	Программа корректно работает	Да
6	Сохранение данных	1. Запуск программы 2. Работа с программой 3. Закрытие программы 4. Открытие файла с данными	Данные сохранены в файл	Да

ЗАКЛЮЧЕНИЕ

В рамках данной работы были проведены расчеты с простыми числами с помощью программы GAP, найдены следы степеней круговых единиц, которые сравнимы с 1 по модулю простого числа, вычислены следы степеней круговых единиц по модулям 2 и 3, найдены степени, которые удовлетворяют предыдущим двум условиям, разработаны алгоритмы расчетов и разработана программная система для изучения круговых единиц для простых чисел. При этом были решены следующие задачи:

1) изучены предметной области;
2) определены функциональных и нефункциональных требований к системе;
3) разработаны и реализованы алгоритмы расчетов;
4) определен способа вывода данных;
5) проведено тестирование системы.

В рамках проведенных исследований были изучены основные принципы работы с круговыми единицами для простых чисел. Была проведена детальная аналитика предметной области, что позволило определить ключевые аспекты, влияющие на работу системы.

После проведения анализа были выделены функциональные и нефункциональные требования к системе. Спроектированные алгоритмы были успешно реализованы и использованы в программной системе.

Особое внимание было уделено способу вывода данных, чтобы обеспечить удобство использования системы и доступность результатов исследований. Благодаря тестированию системы удалось проверить ее работоспособность и убедиться в правильности проведенных расчетов.

Таким образом, в результате работы была разработана и успешно протестирована программная система, позволяющая изучать круговые единицы для простых чисел. Полученные результаты могут быть использованы в дальнейших исследованиях в области теории чисел и вычислительной математики.

ЛИТЕРАТУРА

1. Виноградов, И.М. Основы теории чисел: учебное пособие. // Санкт-Петербург: Лань, 2009. – 176 с.

2. Нестеренко, Ю.В. Теория чисел: учебник для студ. высш. учеб. заведений. // М.: Издательский центр «Академия», 2008 – 292 с.

3. The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.12.2. [Электронный ресурс] URL: https://www.gap-system.org (дата обращения: 26.02.2024 г.).

4. Cohen, H. A. Course in computational algebraic number theory Springer-Verlag. // Graduate Texts in Mathematics, 1993. – 534 p.

5. GAP – Reference Manual. [Электронный ресурс] URL: https://docs.gap-system.org/doc/ref/chap0_mj.html (дата обращения: 01.03.2024 г.).

6. IBM Documentation – Use-case diagrams. [Электронный ресурс] URL: https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case (дата обращения: 05.03.2024 г.).

7. Гиндикин С. Дебют Гаусса // Квант, №1, 1972. С. 2–11

8. Кириллов А.А. О правильных многоугольниках, функции Эйлера и числах Ферма // Квант, №7, 1977. С. 2–9. Квант, №6, 1994. С. 15–18.

9. Лагранжа теорема // Математическая энциклопедия (в 5 томах). – М.: Советская Энциклопедия, 1982. –174 с.

10. Хабр – Алгоритмы быстрого возведения в степень. [Электронный ресурс] URL: https://habr.com/ru/companies/otus/articles/779396/ (дата обращения: 07.03.2024 г.).

11. Метод факторизации Л. Инфельд и Т.Е. Халл. [Электронный ресурс] URL: https://ega-math.narod.ru/Nquant/Infeld.htm (дата обращения: 09.03.2024 г.).

12. Python. [Электронный ресурс] URL: https://www.python.org/ (дата обращения: 01.04.2024 г.).

13. Python 3.9.19 documentation. [Электронный ресурс] URL: https://docs.python.org/3.9/ (дата обращения: 01.04.2024 г.).

14. Почему стоит учить Python – Mate academy. [Электронный ресурс] URL: https://mate.academy/blog/ru/python-ru/python-2023 (дата обращения: 01.04.2024 г.).

15. Что такое Python? – Описание языка программирования Python – AWS. [Электронный ресурс] URL: https://aws.amazon.com/ru/what-is/python/ (дата обращения: 01.04.2024 г.).

16. Python Data Types – DigitalOcean. [Электронный ресурс] URL: https://www.digitalocean.com/community/tutorials/python-data-types (дата обращения: 05.04.2024 г.).

17. Shafranovich Y. Common Format and MIME Type for Comma-Separated Values (CSV) Files. // SolidMatrix Technologies, Inc., 2005 – P. 8

18. Pandas - Python Data Analysis Library. [Электронный ресурс] URL: https://pandas.pydata.org (дата обращения: 20.04.2024 г.).

19. Welcome to Flask – Flask Documentation. [Электронный ресурс] URL: https://flask.palletsprojects.com/en/3.0.x/ (дата обращения: 22.04.2024 г.).

20. Flask Vs Django: Which Python Framework to Choose? – Great Learning. [Электронный ресурс] URL: https://www.mygreatlearning.com/ (дата обращения: 23.04.2024 г.).

ПРИЛОЖЕНИЯ

Приложение А. Блок-схемы алгоритмов

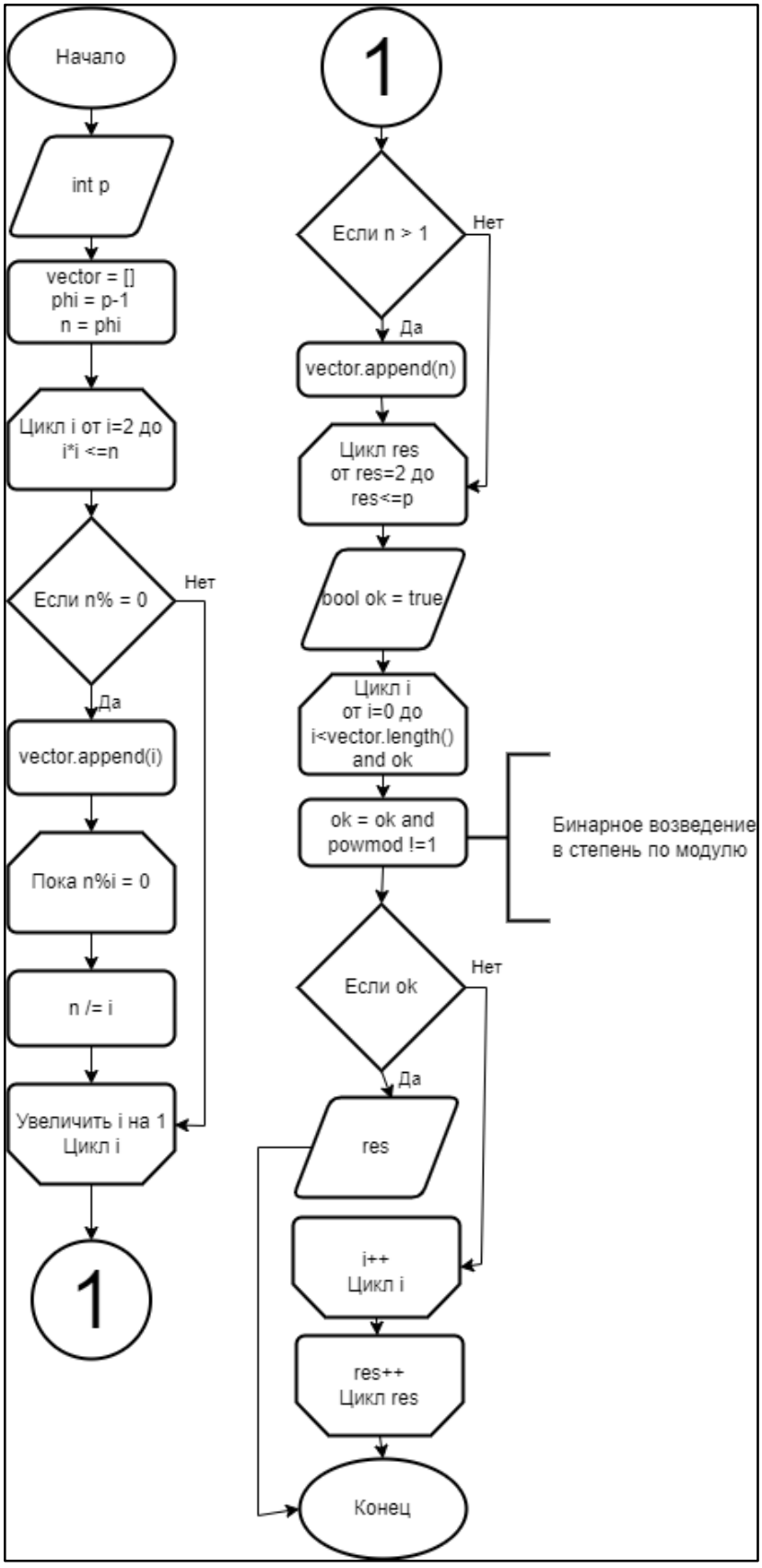

Рисунок 1 – Алгоритм вычисления примитивного корня

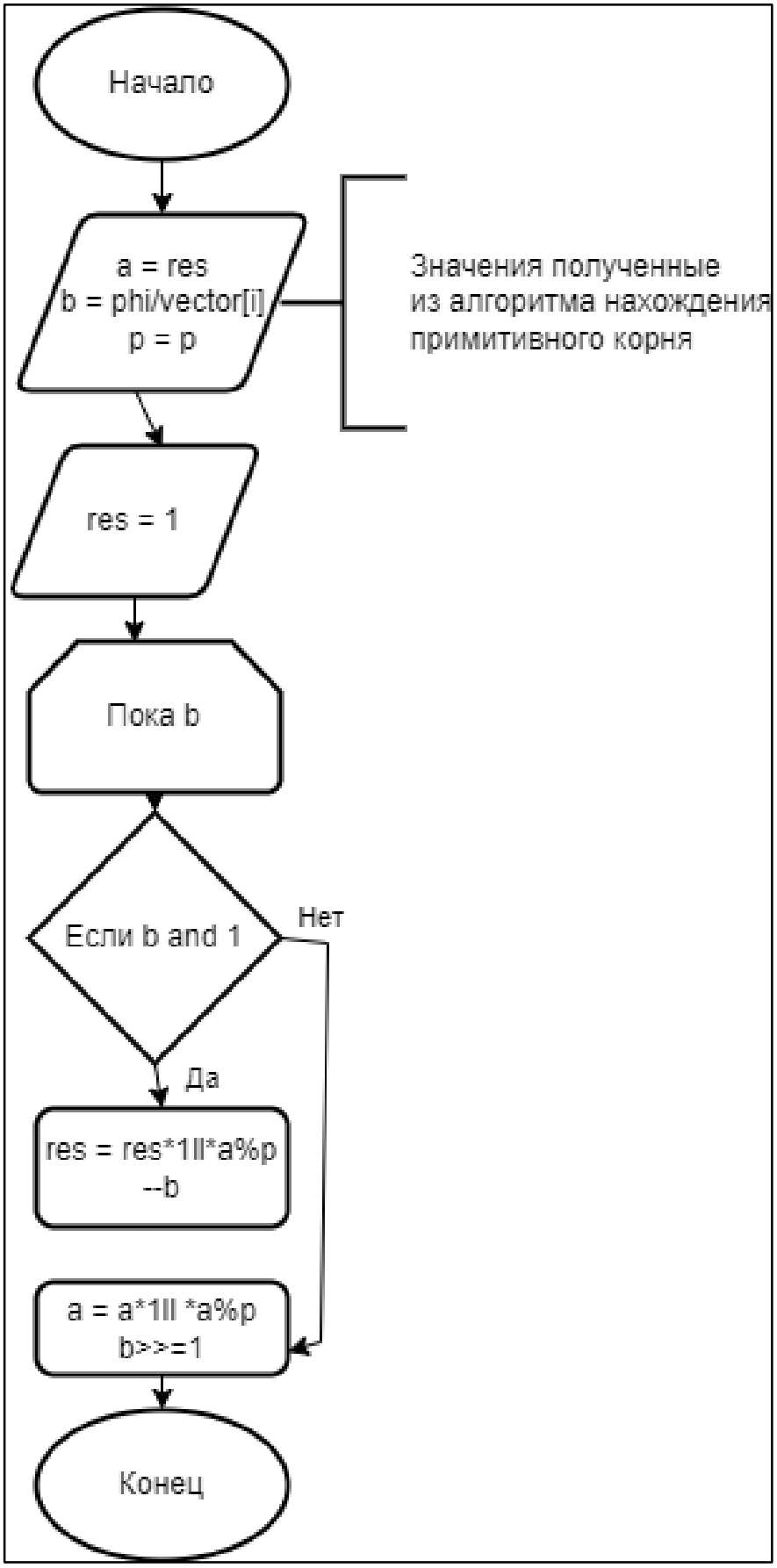

Рисунок 2 – Алгоритм бинарного возведения в степень по модулю

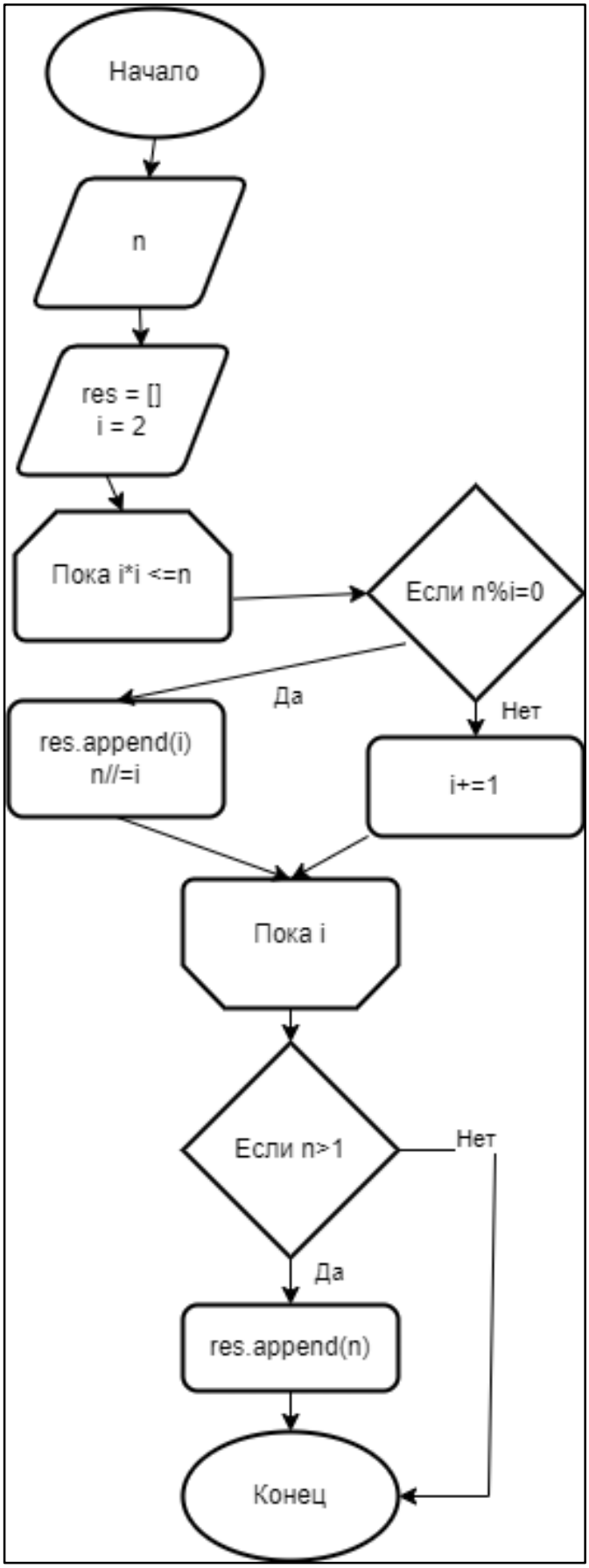

Рисунок 3 – Алгоритм разбития на простые множители

Приложение Б. Результаты вычислений

Таблица 1 – Результаты расчетов

p	g	след	f2	f2	F2	F3	ep2	ep3	для 2	для 3
5	2	1	4	4	15	40	30	40	–c^ep2	c^ep3
7	3	6	3	6	7	364	21	1092	c^ep2	c^ep3
11	2	1	10	5	341	121	1705	605	–c^ep2	–c^ep3
13	2	1	12	13	819	26	1638	78	–c^ep2	–c^ep3
17	3	16	8	16	255	223040	2040	223040	c^ep2	c^ep3
19	2	1	18	18	87381	387420488	87381	3486784392	–c^ep2	–c^ep3
23	5	22	11	11	2047	88573	22517	974303	c^ep2	c^ep3
29	2	1	28	28	2375535	1387060720	33257490	9709425040	–c^ep2	–c^ep3
31	3	30	5	30	31	45141664568	465	677124968520	c^ep2	c^ep3
37	2	1	36	18	3	193710244	1260907830	1743392196	–c^ep2	–c^ep3
41	6	40	20	8	209715	3280	838860	3280	c^ep2	c^ep3
43	3	1	14	42	5461	297262317351272	114681	891786952053816	–c^ep2	–c^ep3
47	5	46	23	23	178481	94143178826	4105063	1082646556499	c^ep2	c^ep3
53	2	1	52	52	1501199875790165	3230540944613336649466120	39031196770544290	41997032279973376443059560	c^ep2	c^ep3
59	2	1	58	29	96076792050570581	34315188682441	2786226969466546849	995140471790789	c^ep2	c^ep3
61	2	1	60	10	384307168202282325	29524	768614336404564650	442860	c^ep2	c^ep3
67	2	1	66	22	24595658764946068821	15690529804	270552246414406757031	517787483532	–c^ep2	c^ep3
71	7	70	35	35	1108378657	25015772549499853	38793252995	875552039232494855	c^ep2	c^ep3
73	5	72	9	12	511	265720	18396	2391480	c^ep2	c^ep3
79	3	78	39	78	549755813887	821160163413032907311573390035462764	21440476741593	24634804902390987219342701701063882932	–c^ep2	–c^ep3

Продолжение таблицы 1 приложения Б

p	g	след	f2	f3	F2	F3	ep2	ep3	для 2	для 3
83	2	1	82	41	16119010928195 05566274901	182364981885853932 01	660879448055997282 17270941	7476964257320011212 41	–c^ep2	–c^ep3
89	3	88	11	88	2047	171259449368480011 63166740682720	90068	1883853943053280127 94834147509920	c^ep2	c^ep3
101	2	1	100	100	42255020007607 64671655677351 25	257688760366005665 518230564882810636 351053761000	845100400152152934 331135470250	2576887603660056655 182305648828106363 51053761000	c^ep2	c^ep3
103	5	102	51	34	17180000257	8338590849833284	876180013107	425268133341497484	–c^ep2	c^ep3
107	2	1	106	53	27043212804868 89389859633504 8021	969162283384000994 8398361	143329027865805137 6625605757545113	5136560101935205272 65113133	c^ep2	c^ep3
109	6	1	36	27	22906492245	3812798742493	137438953470	205891132094622	c^ep2	c^ep3
113	3	112	28	112	89478485	756300677501419560 285974296360319968 0	5010795160	5294104742509936922 0018200745222397760	–c^ep2	c^ep3
127	3	126	7	126	127	906319399233301609 598513021660405922 6179192	8001	8156874593099714486 3866171949436533035 612728	c^ep2	c^ep3
131	2	1	130	65	45370982256125 12846178328099 09024281941	515052573043876872 6986773633921	294911384664813335 001591326440865783 26165	3347841724785199672 54140286204865	–c^ep2	–c^ep3
137	3	136	68	136	98382635059784 275285	386777005072714220 941742234218637674 827573731284608967 58392580560	669001918406533071 9380	6575209086236141756 0096179817168404720 6875343183835244892 673869520	c^ep2	–c^ep3

Продолжение таблицы 1 приложения Б

p	g	след	f2	f3	F2	F3	ep2	ep3	для 2	для 3
139	2	1	138	138	11614971457568 03288621651993 36710216176981	348099304565442798 847568010796773907 344816358156148070 825533225044	267144343524064756 382979958474433497 2070563	2401885201501555312 0482192744977399606 7923287127742168869 61792528036	c^ep2	c^ep3
149	2	1	148	148	11893730772549 66567548571641 20791261365228 885	205549158352848318 291500434695387024 548040611327623874 34176911405652680	880136077168675259 985943014493855334 1026937490	7605318859055387776 7855160837293199082 7750261912208335064 545722009149160	–c^ep2	c^ep3
151	6	150	15	50	32767	717897987691852588 770248	2457525	2692117453844447207 8884300	–c^ep2	c^ep3
157	5	1	52	78	15011998757901 65	821160163413032907 311573390035462764 4	117093590311632870	2463480490239098721 9347201701063882932	c^ep2	–c^ep3
163	2	1	162	162	19486688497745 37224271579776 95504402620791 0057301	983135252377764568 090379542634560581 415517254721073834 636577077689831955 98404	194866884977453722 427157977695504402 6207910057301	7963395544259893001 5320742953399407094 6568976324069806055 6274329287638843470 724	–c^ep2	–c^ep3
167	5	166	83	83	96714065569170 33397649407	199541919709366996 476712333778617451 7613	802726744224113772 004900781	1656197933587746070 7567123703625248496 1879	c^ep2	c^ep3
173	2	1	172	172	19954369021691 26117654097691 60196508283689 9898676565	580531535176546199 811688216130281677 720048783740246888 614552398605068871 6890187320	171607573586544846 118252401477768997 123973391286184590	2496285601259148659 1902593293602112141 9620977008306162104 2575314001796148262 78054760	c^ep2	c^ep3

Окончание таблицы 1 приложения Б

p	g	след	f2	f3	F2	F3	ep2	ep3	для 2	для 3
179	2	1	178	89	12770796173882 40715298622522 62525765301561 593515300181	145466059468128540 431523291324612122 3340241	113660085947553423 661577404513647931 118389818228617161 09	1294647929266344009 8405572927890478887 7281449	–c^ep2	c^ep3
181	2	1	180	45	51083184695529 62861194490090 50103061206246 374061200725	147715635327541684 9321	102166369391059257 223889801810020612 241249274812240145 0	1329440717947875164 38890	–c^ep2	–c^ep3
191	19	190	95	95	39614081257132 16879677197516 7	106044757352265705 974580479375642237 1815036053	376333771942755603 5693337640865	1007425194846524206 7585145540686012532 2428425035	c^ep2	–c^ep3
193	5	192	96	16	26409387504754 77919784798344 5	21523360	845100400152152934 331135470240	64570080	–c^ep2	–c^ep3
197	2	1	196	196	33477875922062 29740712421025 71075542192125 62370474850735 445	163959252392515289 715297112367765201 746440402872465738 255910470209316352 408211224070607838 9160	328083184036210514 589817260519654031 348283111230653537 2073610	8034003367233249196 0495585060204948855 7557974075082117453 9613040256501268002 3499794597841068840	c^ep2	c^ep3
199	3	198	99	198	63382530011411 47007483516026 87	147563327153263760 743767401130988681 571796362585219164 430319423188384717 167390101663547055 02444	627487047112973553 74086808666013	1460876938817311231 3632972711967879475 6078398959366972786 0162289565008699957 1620064691158447419 56	c^ep2	–c^ep3