

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра информационной безопасности**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Программирование»**  
**Тема: Приложение «Калькулятор»**

Студент гр. 3364		Юсфи.А.
Преподаватель		Халиуллин Р.А.

Санкт-Петербург  
2023

## ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Юсфи.А.
Группа 3364
Тема проекта: Приложение «Калькулятор»
<p>Исходные данные:</p> <p>Разработать на языке программирования С или С++ (по выбору студента) приложение для выполнения простых расчетов с использованием основных арифметических операций (сложение, вычитание, умножение, деление). Значения для расчета (не менее двух значений) и арифметическая операция вводятся пользователем. Поддержка вещественных значений (значений с плавающей точкой) не обязательна. Допускается возникновение целочисленного переполнения в результате выполнения арифметической операции. Ввод/вывод значений должен осуществляться в десятичной системе счисления. Приложение должно иметь консольный или графический интерфейс (по выбору студента). В интерфейсе приложения допускается использовать буквы латинского алфавита для транслитерации букв алфавита русского языка. Интерфейс приложения должен быть интуитивно понятным и содержать подсказки для пользователя. В исходном коде приложения должны быть реализованы функции. В исходном коде приложения должны быть реализованы проверки аргументов реализованных функций и проверки возвращаемых функциями значений (для всех функций — как сторонних, так и реализованных). Приложение должно корректно обрабатывать ошибки, в том числе ошибки ввода/вывода, выделения/освобождения памяти и т. д.</p>

<p>Содержание пояснительной записки:</p> <ul style="list-style-type: none"> <li>• АННОТАЦИЯ.</li> <li>• SUMMARY.</li> <li>• СОДЕРЖАНИЕ.</li> <li>• ВВЕДЕНИЕ</li> <li>• ЗАКЛЮЧЕНИЕ</li> <li>• СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</li> </ul>		
<p>Предполагаемый объем пояснительной записки: Не менее 25 страниц.</p>		
<p>Дата выдачи задания: 12.11.2023</p>		
<p>Дата сдачи реферата: 25.12.2023</p>		
<p>Дата защиты реферата: 25.12.2023.</p>		
Студент		Юсфи.А.
Преподаватель		Халиуллин Р.А.

## **АННОТАЦИЯ**

Курсовой проект включает в себя разработку приложения-калькулятора на языке программирования C с основными арифметическими операциями. Методы исследования включают в себя анализ существующих решений, определение функциональных требований и выбор технологий, таких как C. Результаты включают успешную реализацию функций сложения, вычитания, умножения и деления с надежной обработкой ошибок. Приложение поддерживает ввод пользователем двух операндов и выбранной операции. Интерфейс разработан для удобства использования, с интуитивными подсказками. Проект обеспечивает стабильную функциональность, успешно достигая цели предоставления надежного приложения-калькулятора.

## **SUMMARY**

The course project involves developing a C calculator application with basic arithmetic operations. Research methods include analyzing existing solutions, defining functional requirements, and selecting technologies like C. Results include the successful implementation of addition, subtraction, multiplication, and division functions with proper error handling. The application supports user input for two operands and the chosen operation. The interface is designed for ease of use, featuring intuitive prompts. The project ensures stable functionality, effectively meeting the goal of providing a reliable calculator application.

## СОДЕРЖАНИЕ

	<b>Введение</b>	6-8
1.	<b>Теоретическая часть</b>	9-12
1.1.	Описание (определение)	9
1.2.	История калькуляторов	9
1.3	Технологический прогресс и современные калькуляторы	10
1.4	Перспективы развития	11-12
2.	<b>Реализация программы</b>	13-17
2.1.	Основные сведения о программном обеспечении	10
2.2.	Реализованные функции	10-15
3.	<b>Результаты тестирования программы</b>	18-19
3.1.	Тестирование программы с помощью база без проверки на наличие каких-либо ошибок	18
3.2.	Проверка входного значения	18
3.3	Проверка деления на Ноль	19
	<b>Заключение</b>	20
	<b>Список использованных источников</b>	21
	<b>Приложение а Руководство пользователя</b>	22-24
	<b>Приложение А блок схема алгоритма</b>	25-26
	<b>Приложение В исходный код программы</b>	27-32

## ВВЕДЕНИЕ

Цель работы:

Создание простого калькулятора на языке программирования C.

Основные задачи и методы их решения:

Валидация ввода:

Задача:

Гарантировать корректность ввода чисел и операций пользователем.

Метод:

Использование функции `buffer()` для очистки вводного буфера и циклов проверки ввода для чисел и операций.

Арифметические функции:

Задача:

Реализовать базовые арифметические операции (сложение, вычитание, умножение, деление).

Метод:

Определение отдельных функций (`addition`, `subtraction`, `multiplication`, `division`) для каждой операции с проверкой возможных ошибок (например, NaN).

Основной цикл программы:

Задача:

Обеспечить многократное выполнение операций до тех пор, пока пользователь не решит завершить работу.

Метод:

Использование цикла `while` в функции `main()`, который продолжается, пока `continueFlag` равен 'y' или 'Y'.

Обработка ошибок:

Задача:

Обеспечить корректное завершение программы при возникновении ошибок.

Метод:

Вывод соответствующих сообщений об ошибке и использование ненулевых кодов завершения для индикации проблем.

Завершение программы:

Задача:

Предоставить пользователю возможность завершить программу при необходимости.

Метод:

Вывод сообщения о завершении программы при вводе 'q' или отказе от продолжения после каждого расчета.

### **Пользовательский опыт:**

Описание:

Положительный пользовательский опыт является ключевым для удобства использования калькулятора. Четкие инструкции, интуитивно понятные подсказки и прямолинейные операции способствуют бесперебойному взаимодействию с пользователем.

Применение:

Калькулятор предоставляет четкие и лаконичные подсказки, направляя пользователя на каждом этапе процесса вычисления.

Механизмы обратной связи, такие как сообщения об ошибках и отображение результатов, гарантируют, что пользователь всегда в курсе и может легко понять ответы программы.

Пример: после каждого вычисления пользователю предоставляется результат, а затем предлагается возможность

выполнить новое вычисление или завершить программу,  
увеличивая гибкость и вовлеченность пользователя.



## **2.Реализация программы**

### **1.1. ОПИСАНИЕ (ОПРЕДЕЛЕНИЕ):**

Калькулятор — это устройство или инструмент, предназначенный для проведения арифметических и, в некоторых случаях, более сложных математических операций. От простых ручных счетов до сложных электронных устройств, его основная функция остается неизменной: помощь в математических вычислениях. Калькуляторы находят применение в различных сферах, от образования до научных исследований и профессиональной деятельности.

### **1.2. История калькуляторов:**

Калькуляторы имеют долгую историю, эволюционируя от простых счетных инструментов, таких как счеты, до сложных электронных устройств. Ключевые вехи включают:

#### **1.2.1. Счеты (около 2400 г. до н.э.):**

Счеты — одно из первых известных инструментов для арифметических вычислений. Используя счеты, древние цивилизации проводили простые математические операции и вели учет различных величин.

#### **1.2.2. Механические калькуляторы (начиная с 17 века):**

С появлением механических калькуляторов, таких как калькулятор Паскаля и арифмометр Чарльза Ксавье Томаса, возможности для быстрых и точных вычислений значительно расширились.

#### **1.2.3. Электромеханические калькуляторы (середина 20 века):**

С развитием электронных технологий в 1940-х и 1950-х годах появились электромеханические калькуляторы, работающие на основе вакуумных трубок и, позже, транзисторов.

#### **1.2.4. Электронные калькуляторы (1960-е годы и далее):**

Появление карманных электронных калькуляторов в 1960-х и 1970-х годах стало настоящей революцией, делая вычисления доступными и удобными для массового пользователя.

### **1.3. Технологический прогресс и современные калькуляторы:**

В последние десятилетия с развитием микроэлектроники и информационных технологий калькуляторы претерпели значительные изменения и усовершенствования.

#### **1.3.1. Программируемые калькуляторы:**

С появлением возможности программирования стали доступны калькуляторы, позволяющие создавать и хранить пользовательские программы для решения сложных задач.

#### **1.3.2. Графические калькуляторы:**

Современные калькуляторы не только проводят вычисления, но и строят графики функций, предоставляя пользователю визуальное представление математических зависимостей.

#### **1.3.3. Взаимодействие с компьютером:**

Благодаря развитию технологий беспроводной связи и интерфейсов, современные калькуляторы могут взаимодействовать с компьютерами, обмениваться данными и использовать специализированные программы.

#### **1.4. Перспективы развития:**

С учетом текущих тенденций в области технологий, таких как искусственный интеллект и квантовые вычисления, можно ожидать дальнейшего развития и инноваций в области калькуляторов, делая их еще более функциональными и многофункциональными инструментами.

##### **1.4.1. Искусственный интеллект в калькуляторах:**

Внедрение искусственного интеллекта (ИИ) в калькуляторы открывает новые горизонты в автоматизации вычислений. Развитие алгоритмов машинного обучения позволяет калькуляторам адаптироваться к индивидуальным потребностям пользователя, предсказывать следующие шаги и обеспечивать более точные результаты.

##### **1.4.2. Квантовые вычисления:**

Возможное использование квантовых вычислений в калькуляторах предоставляет уникальные преимущества в области обработки данных. Эта технология может значительно ускорить решение сложных математических задач, которые сегодня требуют значительного времени.

##### **1.4.3. Расширенные функциональности:**

Ожидается, что будущие калькуляторы будут обладать расширенными функциональностями, включая интеграцию с облачными вычислениями, совместимость с различными устройствами и возможностью создания пользовательских плагинов и приложений.

##### **1.4.4. Экологическая устойчивость:**

С течением времени становится все более важным обеспечение экологической устойчивости технологий. Будущие калькуляторы, вероятно, будут разрабатываться с учетом энергосбережения и использования экологически чистых материалов.

#### **1.4.5. Образовательные инновации:**

Развитие калькуляторов также будет связано с образовательными инновациями. Интерактивные обучающие программы, поддерживаемые калькуляторами, могут изменить подход к математическому образованию, делая его более доступным и увлекательным.

#### **1.4.6. Мобильные и переносные решения:**

С мобильностью в центре внимания, будущие калькуляторы, вероятно, станут еще более компактными и удобными для использования на мобильных устройствах, что повысит их доступность и практичность.

#### **1.4.7. Этические и безопасные решения:**

С ростом автоматизации в обработке данных и вычислениях возникает необходимость в разработке этических и безопасных решений. Калькуляторы будущего, скорее всего, будут обеспечены средствами защиты данных и механизмами предотвращения ошибок.

## 2. Реализация программы

### 1.1. Основные сведения о программном обеспечении:

Курсовая работа была выполнена мной на языке программирования Си. Использовались операционная система **Windows**, и среда разработки **Clion** от **Jetbrains** Community 2023 версии 2023.3.2 с, и используйте **GitHub**. Протестировано в среде IDE и с использованием (**exe**).

### 1.2. Реализованные функции:

#### Функция **Buffer** (Буфер):

Описание:

Функция **buffer()** необходима для очистки входного буфера. Это гарантирует удаление любых оставшихся или нежелательных символов в буфере перед тем, как программа продолжит выполнение.

Применение:

Когда пользователь вводит символ или строку вместо числа, функция **buffer()** очищает входной буфер. Это предотвращает ошибочную интерпретацию программой недопустимого ввода и позволяет запросить новый ввод.

Пример: если пользователь по ошибке вводит символ "А" вместо числа, функция **buffer** гарантирует, что этот символ будет очищен, и программа сможет запросить действительный числовой ввод.

#### Арифметические функции:

Описание:

Арифметические функции (сложение, вычитание, умножение, деление) являются основой функциональности калькулятора. Они отвечают за выполнение основных математических операций и обеспечивают точность результатов.

Применение:

Сложение:

Проверяет вводные числа и выполняет операцию сложения. Функция возвращает сумму двух чисел.

Вычитание:

Проверяет входные значения и вычисляет разницу между ними, возвращая результат.

Умножение:

Проверяет вводные данные и вычисляет произведение указанных чисел, предоставляя результат.

Деление:

Обеспечивает, что знаменатель не равен нулю, а затем выполняет операцию деления, возвращая частное.

Пример:

Сложение: Ввод чисел 5 и 3. Результат: 8

Вычитание: Ввод чисел 7 и 4. Результат: 3

Умножение: Ввод чисел 6 и 2. Результат: 12

Деление: Ввод чисел 8 и 2. Результат: 4

Главная функция:

Описание:

Главная функция действует как организатор калькулятора, управляя взаимодействием с пользователем, вызовом арифметических функций и отображением результатов.

Применение:

Функция получает ввод от пользователя для чисел и операций.

Она вызывает соответствующую арифметическую функцию на основе выбранной операции.

Отображает вычисленный результат пользователю и предлагает варианты для дальнейших вычислений или завершения работы программы.

Пример:

Пользователь вводит первое число: 10

Пользователь выбирает операцию: '+'

Пользователь вводит второе число: 5

Результат отображается: 15

Пользователь получает запрос на продолжение или завершение.

### **Основная функция(main):**

Функция ``main`` служит основным драйвером программы калькулятора. Она организует взаимодействие с пользователем, вызывает необходимые арифметические функции и управляет общим потоком программы.

#### **1. Инициализация:**

- Переменные, такие как ``Num1``, ``Num2``, ``result``, ``operation`` и ``continueFlag``, объявлены для хранения пользовательских вводов и результатов.

- Пользователю отображается приветственное сообщение.

#### **2. Цикл взаимодействия с пользователем:**

- Цикл ``while`` (``while (continueFlag == 'y' || continueFlag == 'Y')``) гарантирует, что калькулятор продолжит работу до тех пор, пока пользователь не решит выйти.

#### **3. Обработка ввода:**

- В рамках цикла программа:

- Предлагает пользователю ввести первое число.

- Проверяет вводимые данные, чтобы убедиться, что это допустимое числовое значение.

- Просит пользователя выбрать арифметическую операцию (`+`, `-`, `*`, `/`) или завершить (`q`).

- Предлагает пользователю ввести второе число и проверяет правильность ввода аналогично первому числу.

#### **4. Арифметическая операция:**

- На основе выбранной операции программа:
  - Вызывает соответствующую арифметическую функцию (`сложение`, `вычитание`, `умножение`, `деление`), передавая введенные числа.
  - Получает результат от арифметической функции и сохраняет его в переменной `результат`.

#### **5. Отображение результата:**

- Вычисленный результат затем отображается пользователю с помощью `printf`.

#### **6. Обработка ошибок:**

- Если во время проверки ввода или арифметических операций возникают какие-либо ошибки, отображаются соответствующие сообщения об ошибках, и программа может завершиться с ненулевым статусом.

#### **7. Запрос на продолжение:**

- После отображения результата программа спрашивает пользователя, желает ли он выполнить другое вычисление. Если пользователь решает продолжить, цикл повторяется снова; в противном случае программа завершается.

#### **Резюме:**

Функция `main` инкапсулирует основную функциональность калькулятора, обрабатывая взаимодействие с пользователем, проверку ввода,



арифметические операции и отображение результатов. Она предоставляет структурированную и интерактивную среду для выполнения пользователями базовых арифметических вычислений до тех пор, пока они не решат выйти из программы.

### **Обработка ошибок:**

#### **Описание:**

Обработка ошибок неотъемлема для надежности программы. Это гарантирует, что программа может корректно обрабатывать неожиданный ввод или условия, предоставляя обратную связь пользователю при необходимости.

#### **Применение:**

Код включает проверки для валидации ввода пользователем, такие как гарантия того, что знаменатель не равен нулю или обнаружение нечисловых символов.

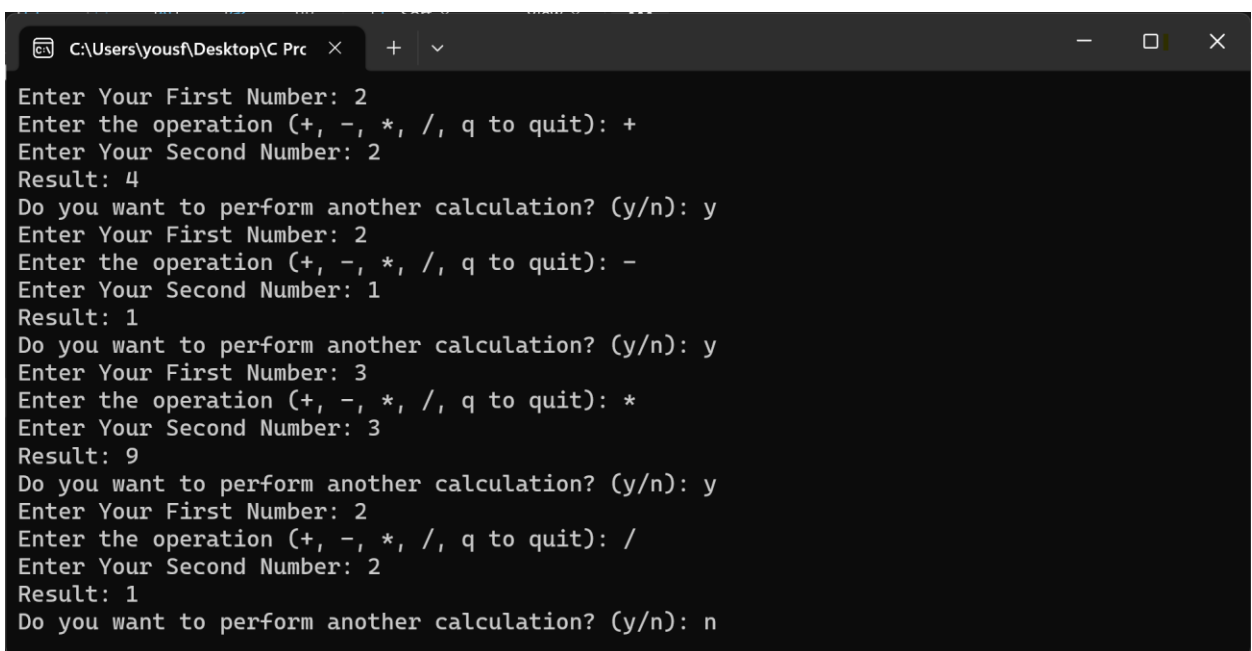
Если обнаружена ошибка, программа отображает описательное сообщение об ошибке, направляя пользователя на корректировку их ввода.

Пример: если пользователь пытается разделить на ноль, программа обнаруживает эту ситуацию, отображает сообщение об ошибке ("Нельзя делить на ноль.") и предлагает пользователю ввести действительный знаменатель.

### 3. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ПРОГРАММЫ

#### 3.1. Тестирование программы с помощью basic без проверки на наличие каких либо ошибок:

- На этом шаге мы протестировали программу, используя (.exe) . и мы протестировали с простым и допустимым значением, чтобы подтвердить, что он работает нормально, в качестве первого шага.

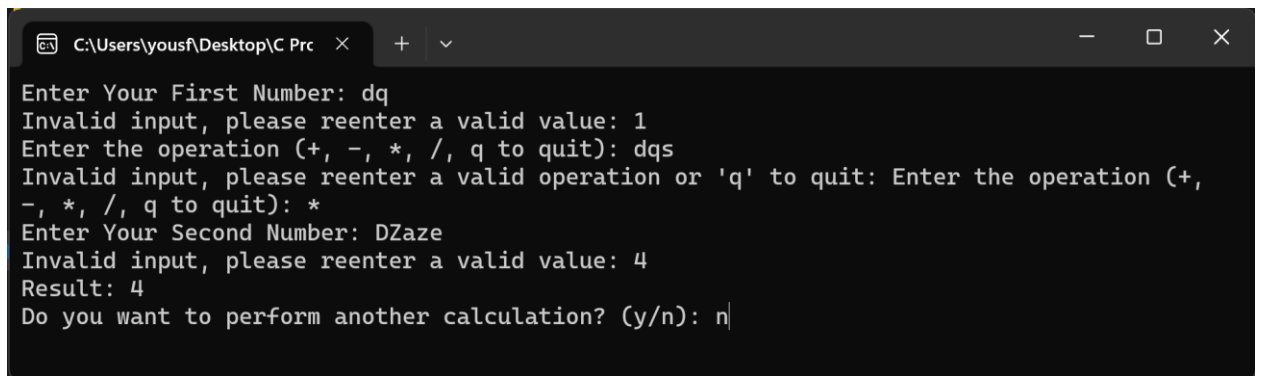


```
C:\Users\yousf\Desktop\C Prc >
Enter Your First Number: 2
Enter the operation (+, -, *, /, q to quit): +
Enter Your Second Number: 2
Result: 4
Do you want to perform another calculation? (y/n): y
Enter Your First Number: 2
Enter the operation (+, -, *, /, q to quit): -
Enter Your Second Number: 1
Result: 1
Do you want to perform another calculation? (y/n): y
Enter Your First Number: 3
Enter the operation (+, -, *, /, q to quit): *
Enter Your Second Number: 3
Result: 9
Do you want to perform another calculation? (y/n): y
Enter Your First Number: 2
Enter the operation (+, -, *, /, q to quit): /
Enter Your Second Number: 2
Result: 1
Do you want to perform another calculation? (y/n): n
```

Рисунок 1: Протестированный подтвержденный ввод.

#### 3.2. Проверка входного значения:

Проверяя допустимый ввод в качестве примера, мы попытались ввести вместо первого числового алфавита, но код проверяется и дает нам возможность ввести еще раз и для второго числа тоже два. и мы проверили арифметическую операцию, если ввод не из допустимого, также проверяем код и возвращаемся к у нас есть шанс внести свой вклад еще раз .

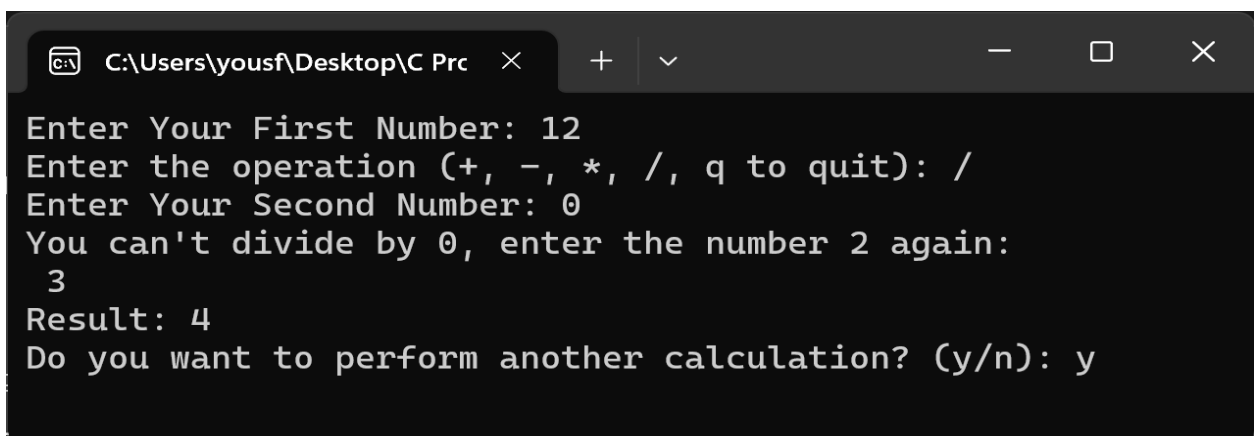


```
C:\Users\yousf\Desktop\C Prc > Enter Your First Number: dq
Invalid input, please reenter a valid value: 1
Enter the operation (+, -, *, /, q to quit): dqs
Invalid input, please reenter a valid operation or 'q' to quit: Enter the operation (+, -, *, /, q to quit): *
Enter Your Second Number: DZaze
Invalid input, please reenter a valid value: 4
Result: 4
Do you want to perform another calculation? (y/n): n
```

**рисунок 2: Проверка входного значения**

### 3.3. Проверка деления на Ноль :

В общем обычный калькулятор, когда мы делим на 0, калькулятор выдает нам ошибку, но в моем коде я проверил и даю возможность пользователю снова ввести второе число .



```
C:\Users\yousf\Desktop\C Prc > Enter Your First Number: 12
Enter the operation (+, -, *, /, q to quit): /
Enter Your Second Number: 0
You can't divide by 0, enter the number 2 again:
3
Result: 4
Do you want to perform another calculation? (y/n): y
```

**Рисунок 3 : Проверка деления на ноль**

Я хочу добавить, что причина, по которой я не запустил опцию n (No), означает, что пользователь не хочет снова использовать калькулятор, поэтому при запуске программа закрылась .

## ЗАКЛЮЧЕНИЕ

Проект целостно выполнил свою цель — создание приложения-калькулятора на языке программирования C, предоставляющего базовый функционал арифметических операций. В процессе анализа альтернативных решений и выбора технологий, включая использование C, были сделаны обоснованные решения. Результаты работы включают успешную реализацию операций сложения, вычитания, умножения и деления с учетом различных сценариев исключительных ситуаций.

Приложение обеспечивает ввод пользователем двух операндов и выбора операции, а также аккуратно обрабатывает возможные ошибки, такие как деление на ноль или неверный ввод. Интуитивно понятный интерфейс и простота использования соответствуют поставленной цели обеспечения удобства для конечного пользователя.

В итоге, проект демонстрирует успешное соответствие своей цели, предоставляя стабильное, надежное и функциональное приложение-калькулятор.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

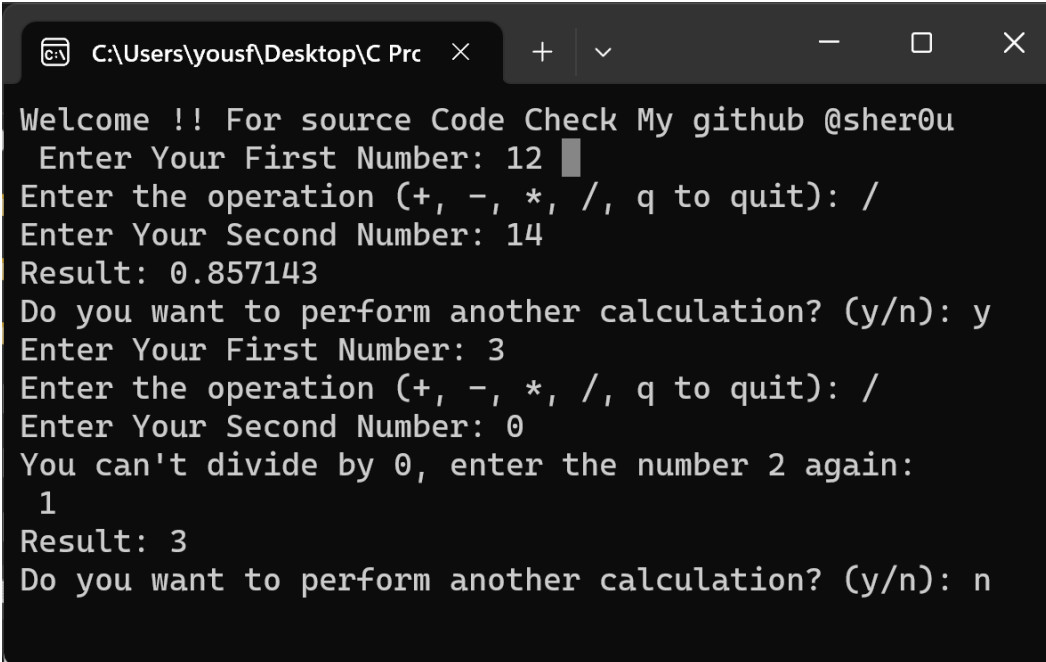
1. "Calculating Machines and Instruments: History, Technology, and Applications" by Geoffrey W.A. Dummer.
2. Standard C Library Functions Table, By Name [Электронный ресурс]. – URL:[https://www.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i\\_74/rtref/stalib.htm](https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_74/rtref/stalib.htm) (дата обращения: 21.12.2020).
3. C/C++ Refences – All C Functions [Электронный ресурс]. – URL: [https://doc.bccnsoft.com/docs/cppreference\\_en/all\\_c\\_functions.html](https://doc.bccnsoft.com/docs/cppreference_en/all_c_functions.html) (дата обращения: 13.12.2020).
4. geeksforgeeks -[Электронный ресурс].URL:<https://www.geeksforgeeks.org/c-programming-language/>

## ПРИЛОЖЕНИЕ А

### РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Эта программа предназначена для выполнения базовых арифметических вычислений. Для оптимальной производительности рекомендуется использовать компьютер с минимум 1 гб оперативной памяти и функциональным процессором. Хотя программа разработана и тщательно протестирована на платформе **Windows**, обеспечивая стабильную работу на этой системе, она также совместима с операционными системами **linux** и **MacOS**.

Для отображения подойдёт любой экран, так как программа будет запускаться в терминале.



```
C:\Users\yousf\Desktop\C Prc x + v - □ x
Welcome !! For source Code Check My github @sher0u
Enter Your First Number: 12
Enter the operation (+, -, *, /, q to quit): /
Enter Your Second Number: 14
Result: 0.857143
Do you want to perform another calculation? (y/n): y
Enter Your First Number: 3
Enter the operation (+, -, *, /, q to quit): /
Enter Your Second Number: 0
You can't divide by 0, enter the number 2 again:
1
Result: 3
Do you want to perform another calculation? (y/n): n
```

ИЗОБРАЖЕНИЕ : ИНТЕРФЕЙС ПРОГРАММЫ

**Шаги:****Выбор плана:**

При запросе введите число 1, чтобы продолжить. Если вы по ошибке введете любое другое число или символ, программа автоматически предложит вам ввести данные снова.

**Выбор операции:**

После ввода 1 для плана программа попросит вас выбрать арифметическую операцию (+, -, \*, /).

Если вы по ошибке введете любой другой символ или число, программа предоставит вам еще один шанс ввести правильную операцию.

**Ввод двух чисел:**

После выбора операции введите первое число.

Для второго числа, если вы по ошибке введете 0 (для операции деления), программа напомним вам и позволит ввести новое значение.

Если вы введете то же число, что и первое, программа уведомит вас и попросит ввести второе число снова.

**Результат вычисления:**

Затем программа отобразит результат арифметической операции на основе двух чисел и выбранной операции.

**Повторить вычисление (по желанию):**

После просмотра результата программа спросит, хотите ли вы повторить вычисление без повторного запуска всей программы.

Введите да, чтобы повторить вычисление.

Введите нет или любой другой символ, чтобы выйти из программы.

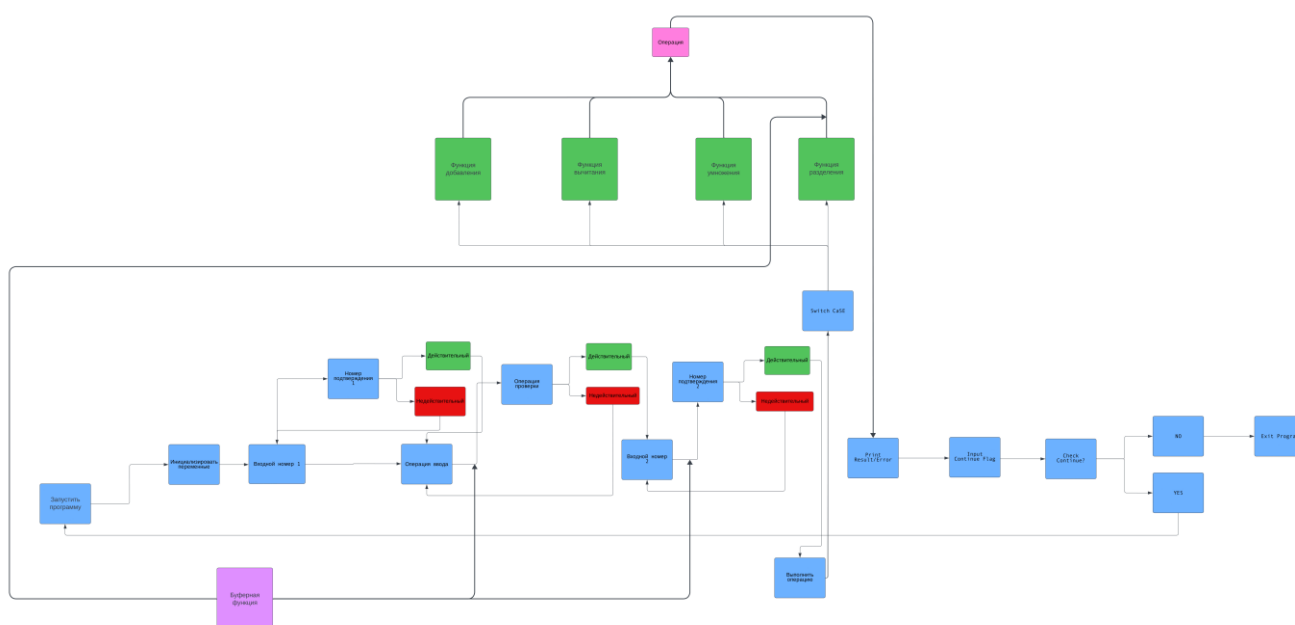
**Заключение:**

Следуйте вышеуказанным шагам, чтобы использовать программу калькулятора и выполнять основные арифметические операции удобно. Убедитесь, что вы вводите правильные значения и операции, чтобы получить точные результаты.



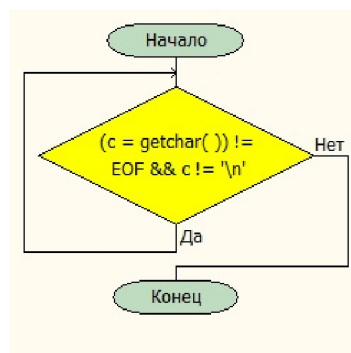
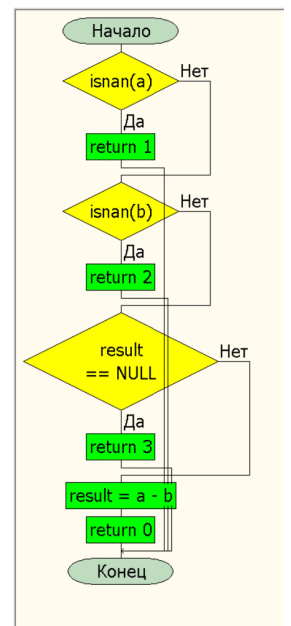
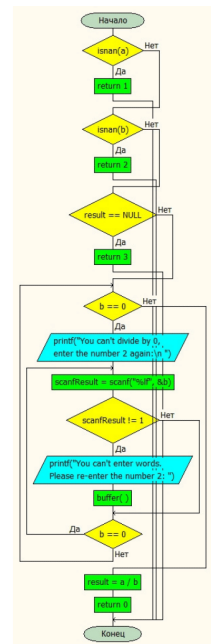
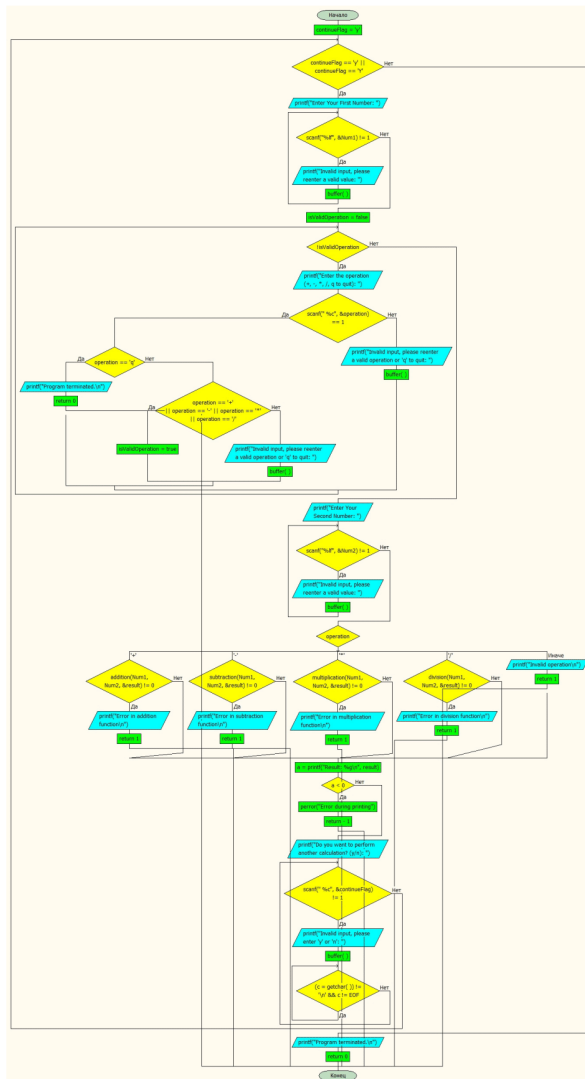
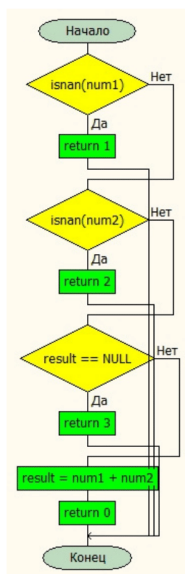
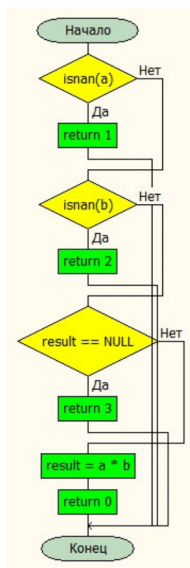
## БЛОК СХЕМА АЛГОРИТМА

Для блока схемы я сделал две схемы, одну для человека, который не разбирается в программировании. а во второй подробно описано, как все работает, и главное, как это работает с деталями.



### Рисунок 1 простая схема блока

## Рисунок 2 детали блока схема



## ПРИЛОЖЕНИЕ С

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <stdio.h>
#include <math.h>

// Function to clear the input buffer
void buffer(void)
{
    int c;
    while ((c = getchar()) != EOF && c != '\n');
}

// Declare the function of addition
double addition(double num1, double num2, double *result)
{
    // Check if num1 is NaN
    if (isnan(num1)) return 1;

    // Check if num2 is NaN
    if (isnan(num2)) return 2;

    // Check if result is NULL
    if (result == NULL) return 3;

    *result = num1 + num2;
    return 0;
}

// Declare the function of subtraction
double subtraction(double a, double b, double *result)
{
    if (isnan(a)) return 1;

    if (isnan(b)) return 2;

    if (result == NULL) return 3;

    *result = a - b;
    return 0;
}

// Declare the function of multiplication
double multiplication(double a, double b, double *result)
{
    if (isnan(a)) return 1;

    if (isnan(b)) return 2;

    if (result == NULL) return 3;
```

```

        *result = a * b;
        return 0;
    }

    // Declare the function of division
    double division(double a, double b, double *result)
    {
        if (isnan(a)) return 1;

        if (isnan(b)) return 2;

        if (result == NULL) return 3;

        // Check for division by zero
        while (b == 0)
        {
            printf("You can't divide by 0, enter the number 2 again:\n
");
            do {
                int scanfResult = scanf("%lf", &b);
                if (scanfResult != 1) {
                    printf("You can't enter words. Please re-enter the
number 2: ");
                    buffer();
                }
            } while (b == 0);
        }

        *result = a / b;
        return 0;
    }

    int main() {
        int check;
        double Num1, Num2, result;
        char operation;
        char continueFlag = 'y';

        check = printf("Welcome !! For source Code Check My github
@sher0u\n ");
        // in the right form
        if (check < 0) {
            perror("Error during printing");
            return 1;
        }

        while (continueFlag == 'y' || continueFlag == 'Y')
        {
            // Get the first number
            check = printf("Enter Your First Number: ");

```

```

        // in the right form
        if (check < 0) {
            perror("Error during printing");
            return 2;
        }
        while (scanf("%lf", &Num1) != 1)
        {
            check =printf("Invalid input, please reenter a valid
value: ");

            // in the right form
            if (check < 0) {
                perror("Error during printing");
                return 3;
            }
            buffer();
        }
        // buffer(); // Uncomment this line if you encounter
issues with subsequent inputs

        // Get the operation from the user
        bool isValidOperation = false;

// Get the operation from the user
        while (!isValidOperation) {
            check =printf("Enter the operation (+, -, *, /, q to
quit): ");
            // in the right form
            if (check < 0) {
                perror("Error during printing");
                return 4;
            }
            if (scanf(" %c", &operation) == 1) {
                if (operation == 'q') {

                    check =printf("Program terminated.\n");
                    // in the right form
                    if (check < 0) {
                        perror("Error during printing");
                        return 5;
                    }
                    return 0;
                } else if (operation == '+' || operation == '-' ||
operation == '*' || operation == '/') {
                    isValidOperation = true; // Exit the loop if a
valid operation is entered
                } else {
                    check = printf("Invalid input, please reenter
a valid operation or 'q' to quit: ");
                    // in the right form

```

```

        if (check < 0) {
            perror("Error during printing");
            return 6;
        }
        buffer();
    }
    } else {
        check = printf("Invalid input, please reenter a
valid operation or 'q' to quit: ");
        // in the right form
        if (check < 0) {
            perror("Error during printing");
            return 7;
        }
        buffer();
    }
}
// buffer(); // Uncomment this line if you encounter
issues with subsequent inputs

// Get the second number
check = printf("Enter Your Second Number: ");
// in the right form
if (check < 0) {
    perror("Error during printing");
    return 8;
}

while (scanf("%lf", &Num2) != 1)
{

    check = printf("Invalid input, please reenter a valid
value: ");
    // in the right form
    if (check < 0) {
        perror("Error during printing");
        return 9;
    }
    buffer();
}
// buffer(); // Uncomment this line if you encounter
issues with subsequent inputs

// now let's make the switch case for the arithmetic
operation
switch (operation)
{
    case '+':
        if (addition(Num1, Num2, &result) != 0)
        {
            printf("Error in addition function\n");

```

```

        return 1;
    }
    break;
case '-':
    if (subtraction(Num1, Num2, &result) != 0)
    {
        printf("Error in subtraction function\n");
        return 1;
    }
    break;
case '*':
    if (multiplication(Num1, Num2, &result) != 0)
    {
        printf("Error in multiplication function\n");
        return 1;
    }
    break;
case '/':
    if (division(Num1, Num2, &result) != 0)
    {
        printf("Error in division function\n");
        return 1;
    }
    break;
default:
    check = printf("Invalid operation\n");
    // in the right form
    if (check < 0) {
        perror("Error during printing");
        return 10;
    }
    return 1; // this will exit with an error code
}

    check = printf("Result: %g\n", result); // we use %g to
print the integer if th ecorrect form and the float
    // in the right form

    if (check < 0) {
        perror("Error during printing");
        return 11;
    }

    // Ask if the user wants to continue

    check =printf("Do you want to perform another calculation?
(y/n): ");
    // in the right form
    if (check < 0) {
        perror("Error during printing");

```

```

        return 12;
    }
    while (scanf(" %c", &continueFlag) != 1) {

        check = printf("Invalid input, please enter 'y' or
'n': ");
        // in the right form
        if (check < 0) {
            perror("Error during printing");
            return 13;
        }
        buffer(); // Clear invalid input

        // Consume any remaining characters in the buffer
including the newline
        int c;
        while ((c = getchar()) != '\n' && c != EOF);
    }

    }

    check =printf("Program terminated.\n");
    // in the right form
    if (check < 0) {
        perror("Error during printing");
        return 13;
    }
    return 0;
}

/*-----
Yousfi abdelakder / GitHub :@sher0u /mail :
abdelkader.yousfi.pr@mail.ru
-----
* Tested to work.
*/

```