|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  **Тема: «Внешние структуры данных: текстовый и двоичный файлы»** | |
|  | |
| Выполнил студент группы ИКБО-11-22 | Гришин А. В. |
| Принял преподаватель | Скворцова Л. А. |

|  |  |  |
| --- | --- | --- |
| Самостоятельная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2023

Содержание

**1.** [**Цель работы** 3](#_Toc147084314)

[**2.** **Задача №1** 3](#_Toc147084315)

[**2.1 Постановка задачи** 3](#_Toc147084316)

[**2.2** **Код программы** 4](#_Toc147084317)

[**2.3** **Выполнение индивидуального задания** 6](#_Toc147084318)

[**2.4** **Тестирование программы** 7](#_Toc147084319)

[**3.** **Задача №2** 8](#_Toc147084320)

[**3.1. Код программы.** 9](#_Toc147084321)

[**3.2. Тестирование программы.** 15](#_Toc147084322)

[**3.3. Входные данные программы.** 17](#_Toc147084323)

# **Цель работы**

Получить навыки применения файловых потоков языка С++ (или файлов языка Си) по управлению текстовым и двоичным файлами

# **Задача №1**

## **2.1 Постановка задачи**

Разработать программу, управления текстовым файлом. Требования:

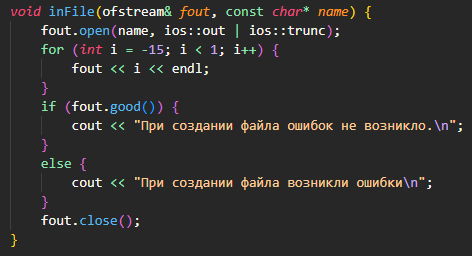
* + - 1. Реализация ввода-вывода на основе файловых потоков С++: ofstream, ifstream.
      2. Имя физического файла вводится пользователем и передается в функции обработки через параметр.
      3. При открытии файла выполнять контроль его существования и открытия.

Примечание. При отладке программы можете имя физического файла определить через константу.

* + - 1. Разработать функции для выполнения операций над текстовым файлом.
         1. создание текстового файла средствами текстового редактора кодировки ASCII, содержащего десятичные числа по несколько чисел на строке;
         2. вывод содержимого текстового файла;
         3. добавление новой записи в конец файла;
         4. прочитать значение числа, указав его порядковый номер в файле, и вернуть его значение;
         5. определить количество чисел в файле.
      2. Разработать приложение и выполнить тестирование всех функций. Приложение должно содержать диалоговый интерфейс на основе текстового меню.
      3. Контроль открытия и существования файла выполнить в основной программе перед вызовом функции. Перед закрытием файла, проверить отсутствие ошибок ввода и вывода (метод good).
      4. Создать модуль и перенести в него все отлаженные функции. Исключить функции из приложения. Отладить приложение, подключив к нему модуль с функциями.
      5. Разработать функции для реализации дополнительных операций, определенных вариантом и сохранить их в модуле с остальными функциями.
      6. Выполнить тестирование приложения в полном объеме.

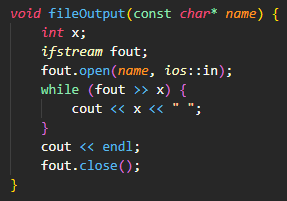
## **Код программы**

Функция inFile выполняет создание текстового файла (рис. 1).



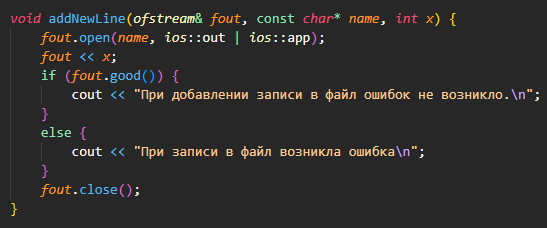
*Рисунок 1 – Функция inFile*

Функция fileOutput выводит содержимое файла на экран (рис. 2).



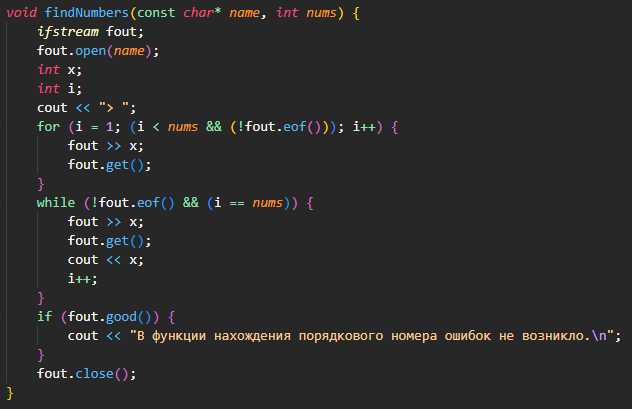
*Рисунок 2 – Функция fileOutput*

Функция addNewLine добавляет новую запись в конец файл (рис. 3).



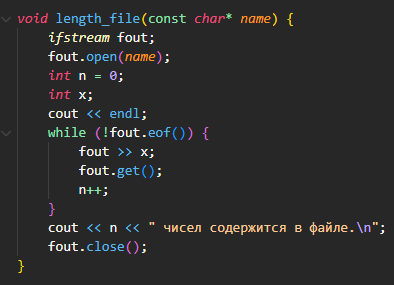
*Рисунок 3 – Функция addNewLine*

Функция findNumbers прочитывает значение числа, указав его порядковый̆ номер в файле, и возвращает его значение (рис. 4).



*Рисунок 4 – Функция findNumbers*

Функция length\_file - определяет количество чисел в файле (рис. 5)



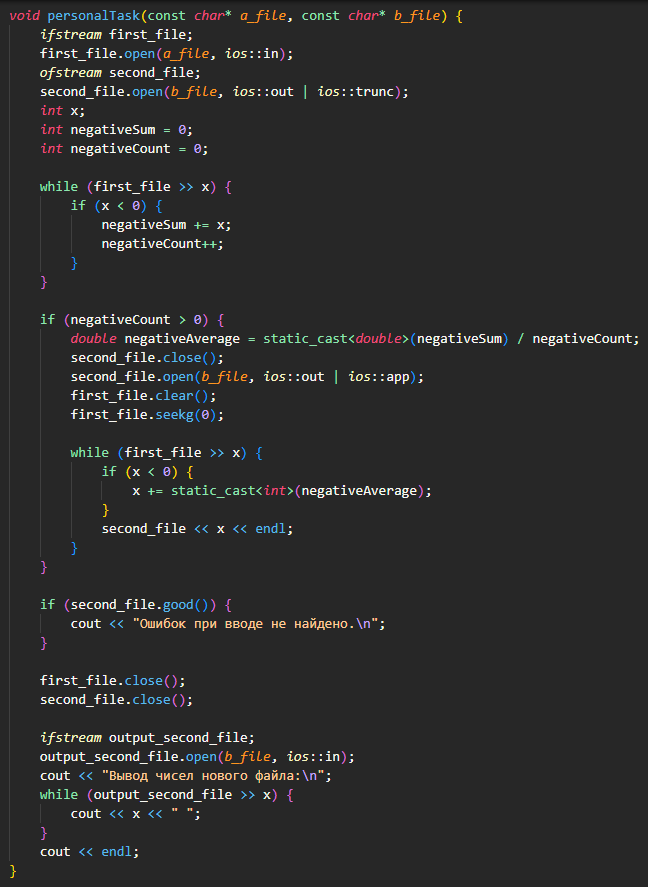
*Рисунок 5 – Функция length\_file*

## **Выполнение индивидуального задания**

Индивидуальный вариант №15

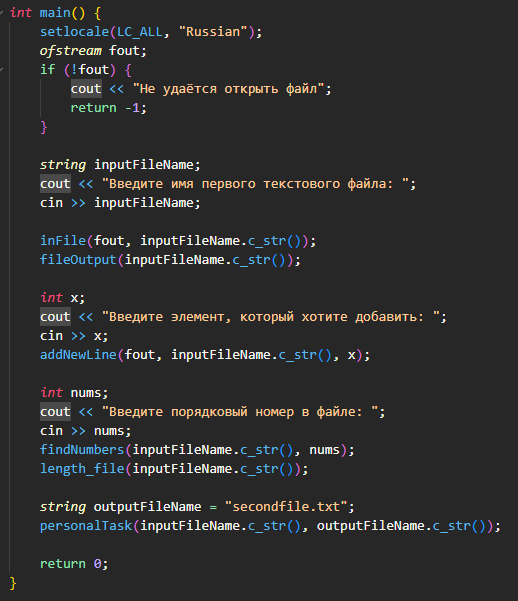
Задание: Создать новый файл из значений исходного, увеличив его на среднее арифметическое всех отрицательных чисел исходного файла.

Функция personalTask создает новый файл из значений исходного, увеличив его на среднее арифметическое всех отрицательных чисел.



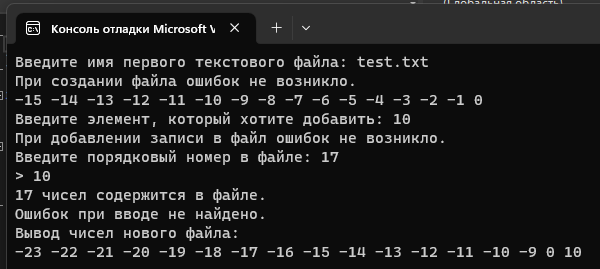
*Рисунок 6 – Функция personalTask*

Функция main — главная функция программы (рис. 7).

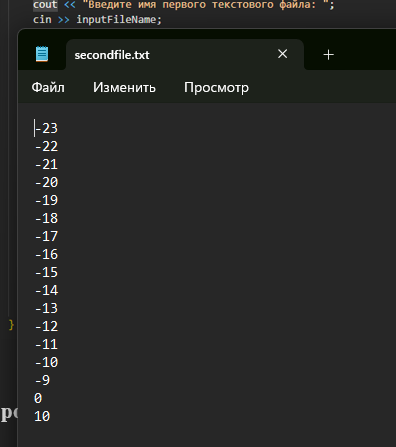


*Рисунок 7 – Функция main*

## **Тестирование программы**



*Рисунок 8 — вывод программы*



*Рисунок 9 — содержание файла*

# **Задача №2**

Разработать программу управление двоичными файлами с записями фиксированной длины

Общие требования: файл состоит из записей определенной структуры, согласно варианту. Записи имеют ключ, уникальный в пределах файла.

Требования к подготовке и выполнению задания

* + - 1. Разработать структуру записи двоичного файла согласно варианту задания.
      2. Подготовить тестовые данные в текстовом файле с кодировкой ASCII, в соответствии со структурой записи варианта. При открытии файла выполнить контроль его существования и открытия. Примечание. Реализация операций по чтению данных из файла будет проще, если значение для каждого поля записи размещать на отдельной строке текстового редактора.
      3. Имя файла вводит пользователь.
      4. При открытии файла обеспечить контроль существования и открытия файла.
      5. При применении механизма прямого доступа к записи файла выполнить контроль присутствия записи с заданным номером в файле.
      6. Разработать функции для выполнения операций: − преобразование тестовых данных из текстового файла в двоичный файл;

− сохранение данных двоичного файла в текстовом, так, чтобы используя их можно было восстановить двоичный файл;

− вывод всех записей двоичного файла;

− доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле;

− удаление записи с заданным значением ключа, выполнить путем замены на последнюю запись.

− манипулирование записями в двоичном файле согласно дополнительным операциям, определенным в варианте;

* + - 1. Сохраните функции в новом модуле.
      2. Разработать приложение, демонстрирующее выполнение всех операций, подключив к нему модуль с функциями.
      3. Выполнить тестирование приложения, продемонстрировав выполнение всех операций.

## **3.1. Код программы.**

Индивидуальный вариант №15.

|  |  |
| --- | --- |
| Структура записи | Найти и поздравить друга: дата рождения, имя |
| Доп. операция | 1. Сформировать в текстовом файле список друзей, родившихся в указанном месяце.  2. Указать имя, родившегося в заданную дату. |

На рисунке 10 показаны структура FriendRecord, имеющая 2 поля (dateOfBirth типа char и name типа char) и функция ConvertToBinary, которая переносит данные из текстового варианта в бинарный.

**

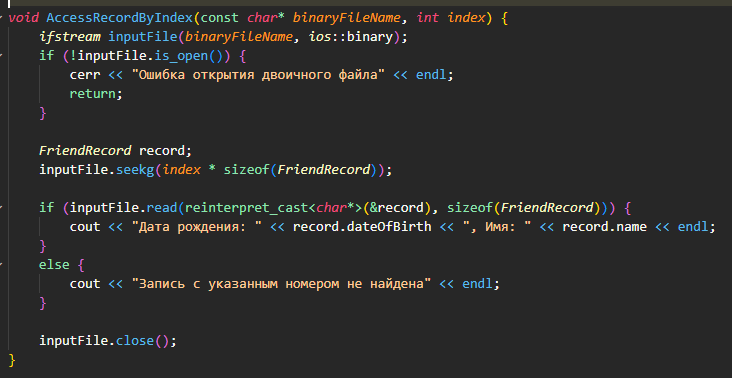
*Рисунок 10 — структура FriendRecord и функция ConvertToBinary.*

Функция SaveToText переносит данные из текстового варианта в бинарный (рис. 11). Функция PrintAllRecords выводит данные на экран (рис. 11).



*Рисунок 11 — функции SaveToText и PrintAllRecords.*

Функция AccessRecordByIndex позволяет получить доступ к записи по индексу (рис. 12).



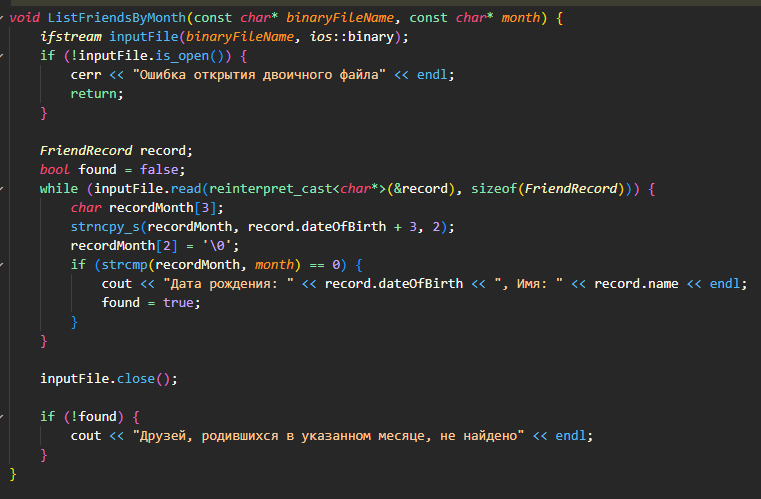
*Рисунок 12 — функция AccessRecordByIndex.*

Функция DeleteRecordByName позволяет удалить запись по ключу (рис. 13).



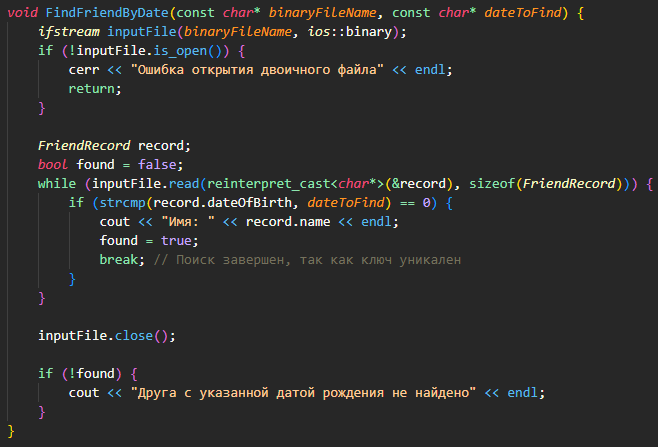
*Рисунок 13 — функция DeleteRecordByName.*

Функция ListFriendsByMonth формирует список друзей, родившихся в указанном месяце (рис. 14).



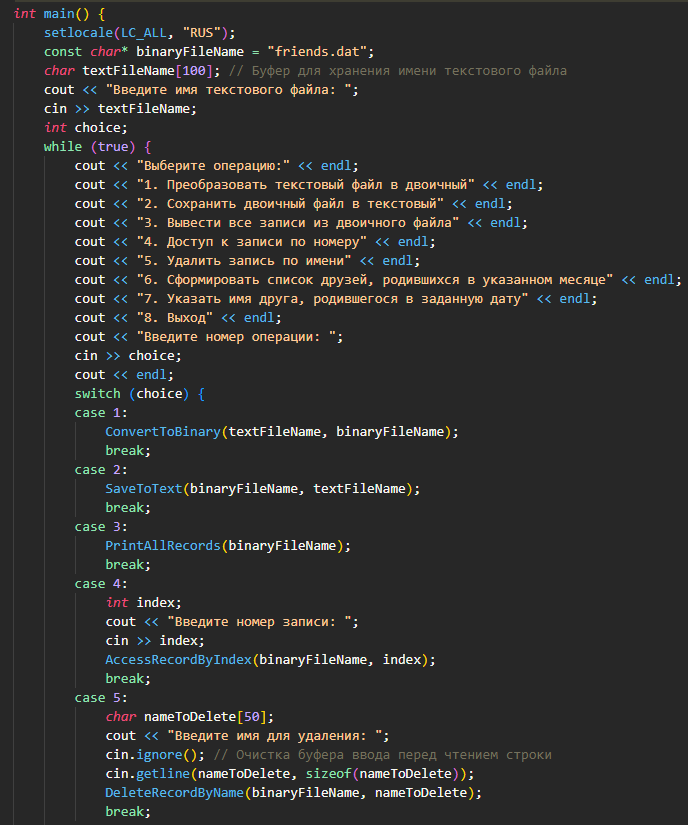
*Рисунок 14 — функция ListFriendsByMonth.*

Функция FindFriendByDate позволяет получить имя друга по дате рождения (рис. 15).

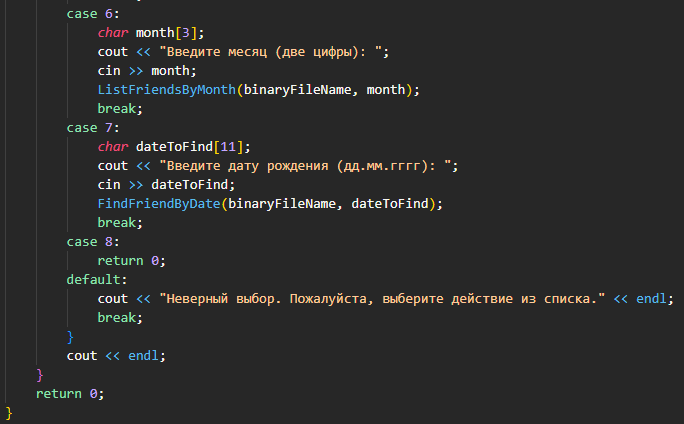


*Рисунок 15 — функция FindFriendByDate.*

Функция main — главная функция программы (рис. 16 и 17).

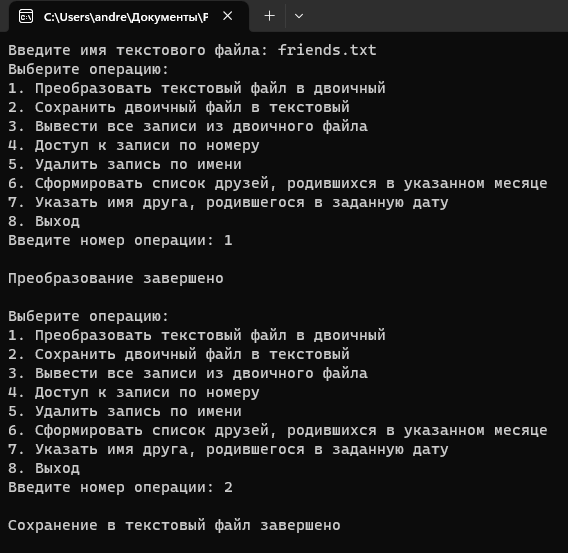


*Рисунок 16 — функция main (1/2).*

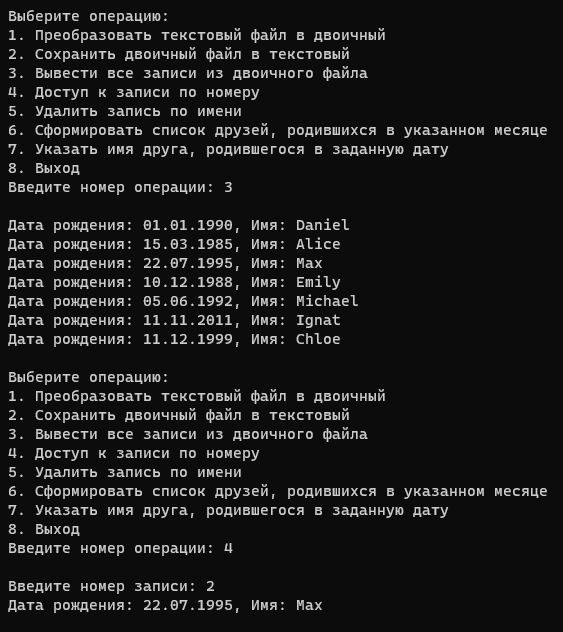


*Рисунок 17 — функция main (2/2).*

## **3.2. Тестирование программы.**

**

*Рисунок 18 — вывод программы.*

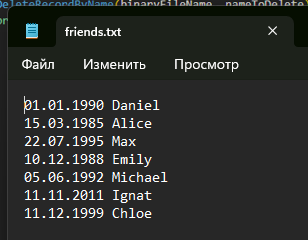
**

*Рисунок 19 — вывод программы.*

**

*Рисунок 20 — вывод программы.*

## **3.3. Входные данные программы.**



*Рисунок 21 — входные данные программы.*