

Министерство образования РБ  
Полоцкий государственный университет

Факультет информационных технологий  
Кафедра технологий программирования

КУРСОВАЯ РАБОТА  
по дисциплине  
«Основы алгоритмизации и программирования»

на тему: «Однотабличная база данных «Библиотека книг»»

Исполнитель: \_\_\_\_\_ Орлов Д.С.

студент 1 курса группы 17-ИТд

Руководитель: \_\_\_\_\_ Тропникова В.А.

Оценка защиты курсовой работы: \_\_\_\_\_

Дата защиты: « \_\_\_\_ » \_\_\_\_\_ 2018 г.

Члены комиссии: \_\_\_\_\_

Полоцк, 2018

## Содержание

Введение .....	4
1 Анализ исходных данных и постановка задач .....	5
2 Проектирование программы .....	6
3 Реализация программы .....	10
4 Методика и результаты тестирования.....	17
Заключение .....	20
Список используемых источников .....	21
Приложения .....	22

					<b>ОДС.1710106.ПЗ</b>			
<b>Изм.</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Подпись</b>	<b>Дата</b>				
Разраб.		Орлов Д.С.			Однотабличная база данных "Библиотека книг" (пояснительная записка)	для	Лист	Листов
Провер.		Тропникова В.А.					3	22
Реценз.						Учреждение образования «Полоцкий государственный университет» гр.17-ИТд		
Н. Контр.								
Утверд.								

## Введение

Язык C++ представляет собой набор команд, которые говорят компьютеру, что необходимо сделать. Этот набор команд, обычно называется исходный код или просто код. Командами являются или «функции» или «ключевые слова». Ключевые слова (зарезервированные слова C/C++) являются основными строительными блоками языка. Функции являются сложными строительными блоками, так как записаны они в терминах более простых функций – это будет представлено в разрабатываемой программе, которая будет описана в пояснительной записке. Структура функций напоминает содержание книги. Содержание может показывать главы книги, каждая глава в книге может иметь своё собственное содержание, состоящее из пунктов, каждый пункт может иметь свои подпункты. Хотя C++ предоставляет много общих функций и зарезервированных слов, которые можно использовать.

Темой курсовой работы является разработка простейшей однотабличной базы данных. База данных содержит информацию о библиотечных книгах, а именно: уникальный шифр книги, автор книги, название книги, дата выдачи книги, дата возврата книги. Разрабатываемая база данных должна включать в себя следующий функционал:

- запись и загрузка файла;
- добавление новых записей, удаление и редактирование старых;
- сортировать записи по любому из полей базы данных в любом направлении;
- фильтровать записи по значению любого поля;
- осуществлять поиск записей по значению любого поля;
- выполнять дополнительную обработку (с сохранением результата в текстовый файл).

В ходе выполнения работы будет проведен анализ задания, описана постановка задач. А также будут рассмотрены основные теоретические сведения по теме курсовой работы. Далее будет описан процесс проектирования программы, реализации программы. В конце будет описана методика и результаты тестирования разработанной простейшей однотабличной базы данных.

## 1 Анализ задания и постановка задач

Курсовая работа представляет собой создание простейшей однотабличной базы данных в предметной области «Библиотечные книги». Программа должна осуществлять следующий функционал:

- запись и загрузка файла;
- добавление новых записей, удаление и редактирование старых;
- сортировать записи по любому из полей базы данных в любом направлении;
- фильтровать записи по значению любого поля;
- осуществлять поиск записей по значению любого поля;
- выполнять дополнительную обработку (с сохранением результата в текстовый файл).

Обработку данных производить в динамическом списке связанного хранения. Выделение и освобождение динамической памяти осуществляется поэлементно. Чтение и запись данных в файл базы данных производится поэлементно. Программа должна обладать дружественным и интуитивно понятным интерфейсом и проводить проверку на корректность вводимых данных. При сдаче курсовой работы на проверку должен быть предоставлен файл базы данных, содержащий не менее 30 записей. Данные должны быть введены корректно, не допускается ввод «мусора». Содержание предоставляемой для проверки базы данных должно позволить осуществить проверку всех функций программы.

База данных содержит информацию о библиотечных книгах: уникальный шифр книги (шестизначное целое число), автор (строка 20 символов), название (строка 30 символов), дата выдачи (дд.мм.гггг), дата возврата (дд.мм.гггг). Дополнительно программа должна формировать отчет по книге (пользователь вводит шифр) за определенный период.

Даже при наличии большого числа специализированных программных продуктов, используемых для создания и ведения баз данных, разработка новых не теряет своей актуальности. Это обусловлено тем, что не все из них отражают ту специфику, все те необходимые аспекты, которые зависят от предметной области и требований. К тому же постоянно возникают новые задачи, изменяются требования к существующим проектам, изменяется сам предмет. Поэтому уже существующие программы зачастую не могут применяться в полной мере. В связи с этим разработку базы данных можно считать актуальной и распространённой задачей.

## 2 Проектирование программы

Для реализации курсовой работы необходимо разработать простейшую однотабличную базу данных. База данных – это представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ) [10].

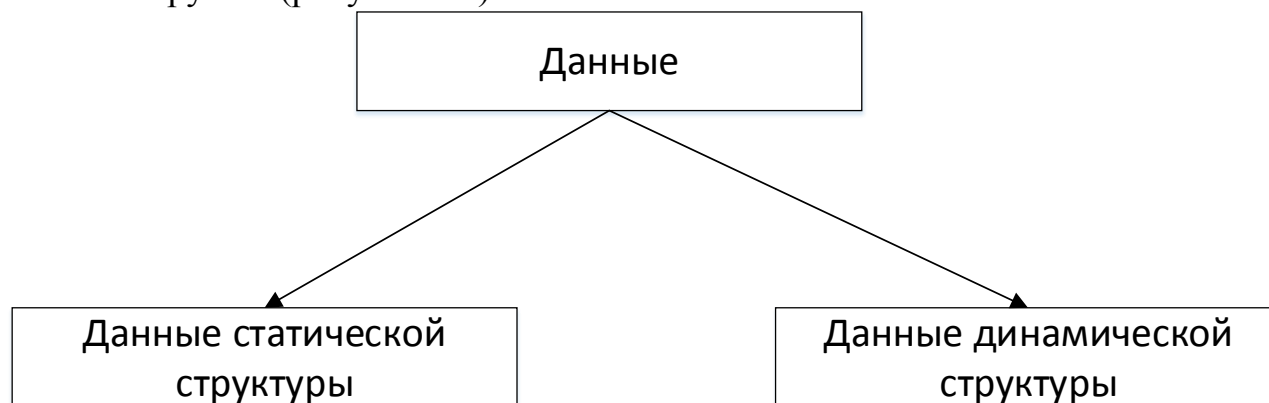
В данной работе база данных представляет собой совокупность записей, хранящихся в типизированном файле.

Запись – это структура, состоящая из фиксированного числа компонент, называемых полями. Данные одного поля разных записей имеют один и тот же тип, а разных полей могут иметь разные типы [10].

Файл – это упорядоченная последовательность однотипных компонентов, расположенных на внешнем носителе. Файлы предназначены только для хранения информации, а обработка этой информации осуществляется программами [10].

Файл, определенный стандартным или пользовательским типом данных, называется типизированным.

Используемые в программировании данные можно разделить на две большие группы (рисунок 2.1):



**Рисунок 2.1** – Группы способов хранения данных в программировании

Данные статической структуры – это данные, взаиморасположение и взаимосвязи элементов которых всегда остаются постоянными.

Данные динамической структуры – это данные, внутреннее строение которых формируется по какому-либо закону, но количество элементов, их взаиморасположение и взаимосвязи могут динамически изменяться во время выполнения программы, согласно закону формирования [10].

Во время работы программы, данные, из типизированного файла перемещаются в двунаправленный список. Двухсвязный список имеет значительное преимущество над массивами, потому как не ограничен числом элементов, позволяет быстрее организовать работу с памятью компьютера.

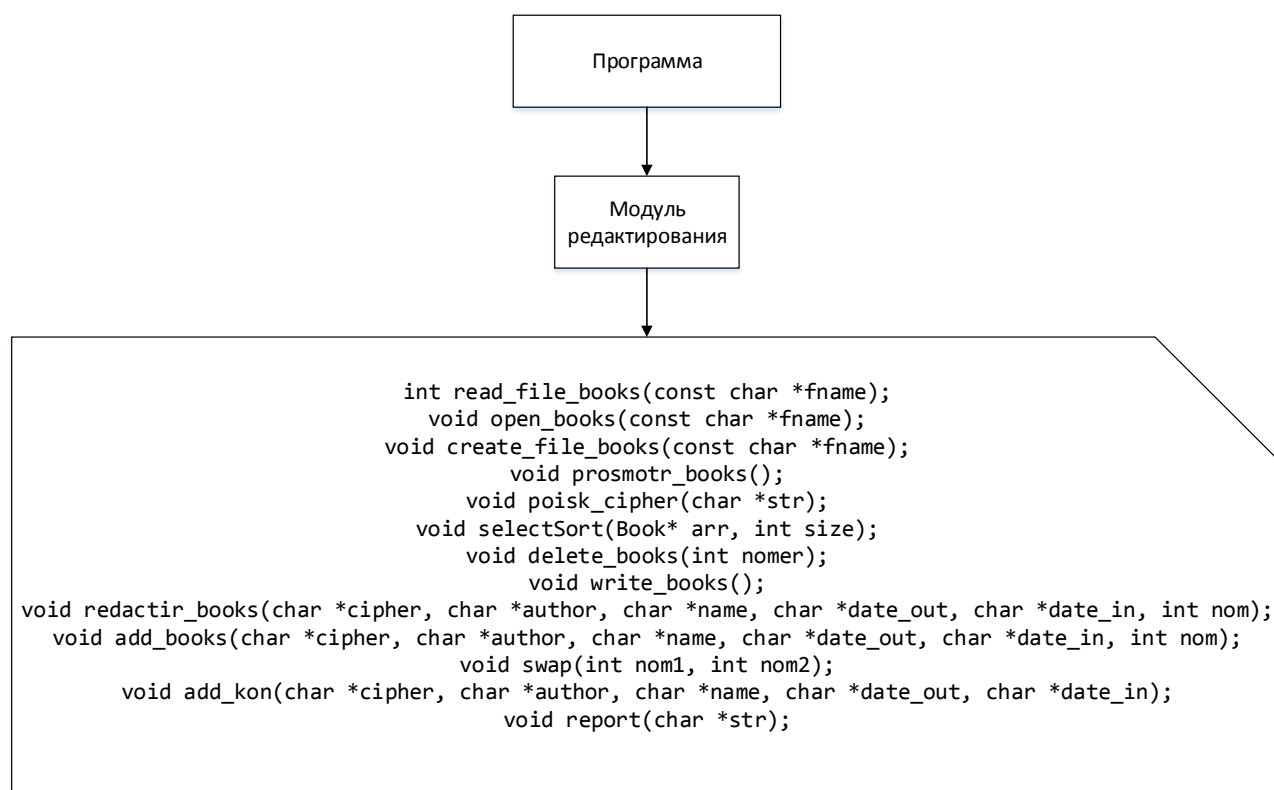
Структура данных, представляющая собой множество элементов (узлов), связанных друг с другом посредством указателей, называется связным списком. Каждый элемент двунаправленного списка содержит поле с данными, а также указатель (ссылку) на следующий и предыдущий элемент. Эта структура позволяет эффективно выполнять операции добавления и удаления элементов для любой позиции в последовательности. Причем это не потребует реорганизации структуры, которая бы потребовалась в массиве. Минусом двунаправленного списка, как и других структур типа «список», в сравнении его с массивом, является отсутствие возможности работать с данными в режиме произвольного доступа, т. е. список – структура последовательно доступа, в то время как массив – произвольного. Последний недостаток снижает эффективность ряда операций [10].

Данные сортируются при помощи алгоритма сортировки вставками.

Алгоритм сортировки – это алгоритм для упорядочивания элементов в списке. В случае, когда элемент списка имеет несколько полей, поле, служащее критерием порядка, называется ключом сортировки

Алгоритм сортировки вставками – на каждом шаге алгоритма мы выбираем один из элементов входных данных и вставляем его на нужную позицию в уже отсортированном списке, до тех пор, пока набор входных данных не будет исчерпан. Метод выбора очередного элемента из исходного массива произволен; может использоваться практически любой алгоритм выбора. Обычно (и с целью получения устойчивого алгоритма сортировки), элементы вставляются по порядку их появления во входном массиве

Структура программы имеет следующий вид (рисунок 3.1).



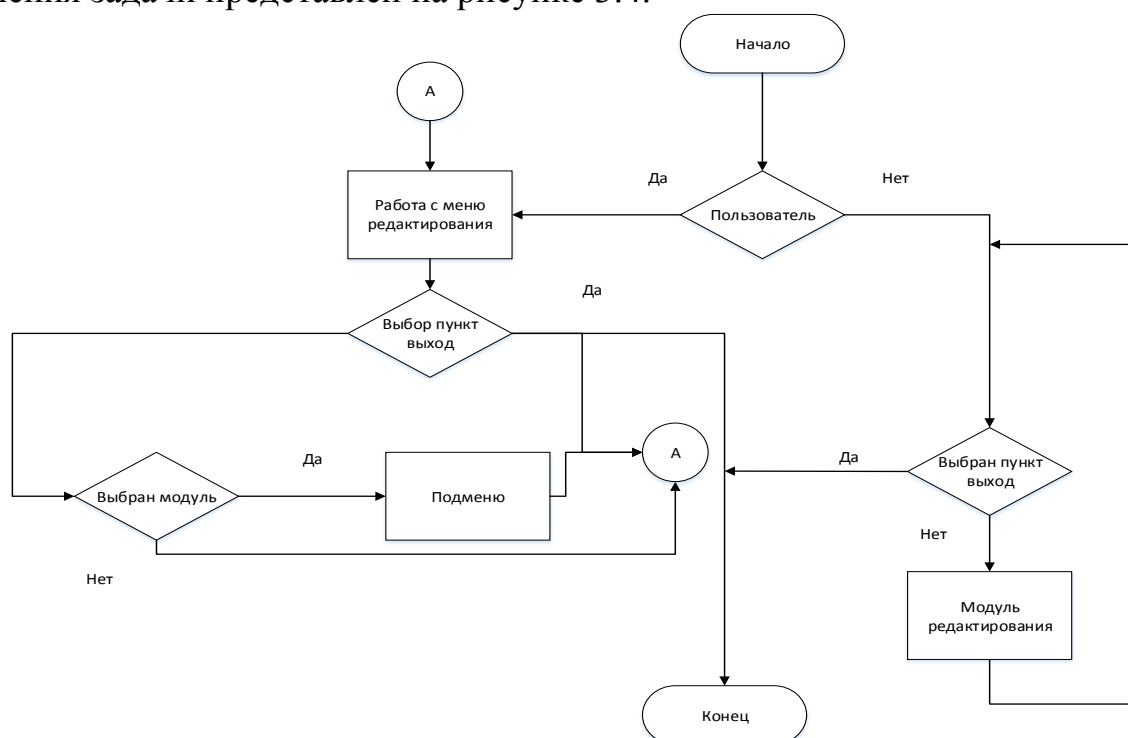
**Рисунок 3.1 – Структура программы**

Данные в программе реализованы в виде структуры: Book. Эта структура хранит в себе следующие поля:

- cipher – уникальный шифр книги;
- author – автор книги;
- name – название книги;
- date\_out – дата выдачи;
- date\_in – дата возврата.

Данные в программе хранятся в виде массивов, которые объявлены глобально. А также в программе хранятся константы, которые оповещают пользователя об одном из действий.

Записи должны записываться в типизированный файл для хранения и переноса на другой компьютер. Корректность файла базы данных проверяется при помощи первой проверочной записи. При записи в начало файла записывается запись с заранее описанной информацией, и при считывании первый делом проверяется наличие этой записи. Отсутствие записи означает, что файл базы данных был создан не в разрабатываемой в рамках данной курсовой работы программе, и не может быть открыт. Все записи из файла, кроме проверочной, копируются в список, после этого они могут быть отредактированы, удалены, отсортированы. После завершения работы со списком требуется сохранить файл, иначе данные будут утеряны. Алгоритм решения задачи представлен на рисунке 3.4.



**Рисунок 3.4** – Обобщённая схема алгоритма решения задачи

Алгоритм (рисунок 3.4) представляет собой вход в программу, после которого пользователю доступно меню редактирования с соответствующими пунктами. После выбора одно из них пользователь попадает в подменю программы для выполнения одной из предложенных операций. При выборе пункта выхода из программы, процесс взаимодействия программы с пользователем прекращается.

### 3 Реализация программы

В программе предусмотрены следующие функции:

- `int read_file_books(const char *fname)` – добавление содержимого файла `books` в массив и возвращение количество записей;
- `void redactor_menu()` – меню редактирования;
- `void open_books(const char *fname)` – открыть файл `books`;
- `void create_file_books(const char *fname)` – создать файл `books`;
- `void prosmotr_books()` – просмотр таблицы `books`;
- `void poisk_cipher(char *str)` – поиск по шифру;
- `void selectSort(Book* arr, int size)` – сортировка выбором;
- `void delete_books(int nomer)` – удалить запись о книге;
- `void write_books()` – запись данных в файл `books`;
- `void redactir_books(char *cipher, char *author, char *name, char *date_out, char *date_in, int nom)` – редактировать запись о книге;
- `void add_books(char *cipher, char *author, char *name, char *date_out, char *date_in, int nom)` – добавить запись о книге;
- `void swap(int nom1, int nom2)` – обмен местами двух записей;
- `void add_kon(char *cipher, char *author, char *name, char *date_out, char *date_in)` – добавление в конец файла;
- `void report(char *str)` – отчет по книге.

Функция `void redactor_menu()` – будет предоставлять меню пользователю для дальнейших действий.

Функция `void open_books(const char *fname)` открывает файл для дальнейшей выгрузки данных в массив.

Функция `void create_file_books(const char *fname)` создаёт файл для дальнейшей записи в него всех данных содержащихся в массиве.

Функция `void swap(int nom1, int nom2)` производит обмен местами данных по двум заданным числам.

Функция `void sort(Exam* arr, int size)` будет производить сортировку данных. В данной сортировке будет применен метод выборки. Этот метод входит в состав медленных сортировок, специфика данной сортировки заключается в том, чтобы пересмотреть каждый элемент массива структуры.

Функция `void poisk_cipher(char *str)` выполняет поиск книг по шифрам. Поиск будет происходить обычным перебором всех данных. Специфика данного поиска заключается в возможности пройти по всем данным и вывести информацию на экран.

Функция `void add_books(char *cipher, char *author, char *name, char *date_out, char *date_in, int nom)` будет происходить добавление элемента в массив. Новый элемент будет добавляться в начало массива.

Функция `void redactir_books(char *cipher, char *author, char *name, char *date_out, char *date_in, int nom)` будет предлагать выбрать книгу, которую нужно будет отредактировать. После чего, на экран будут выводиться данные

					ОДС.1710106.ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		



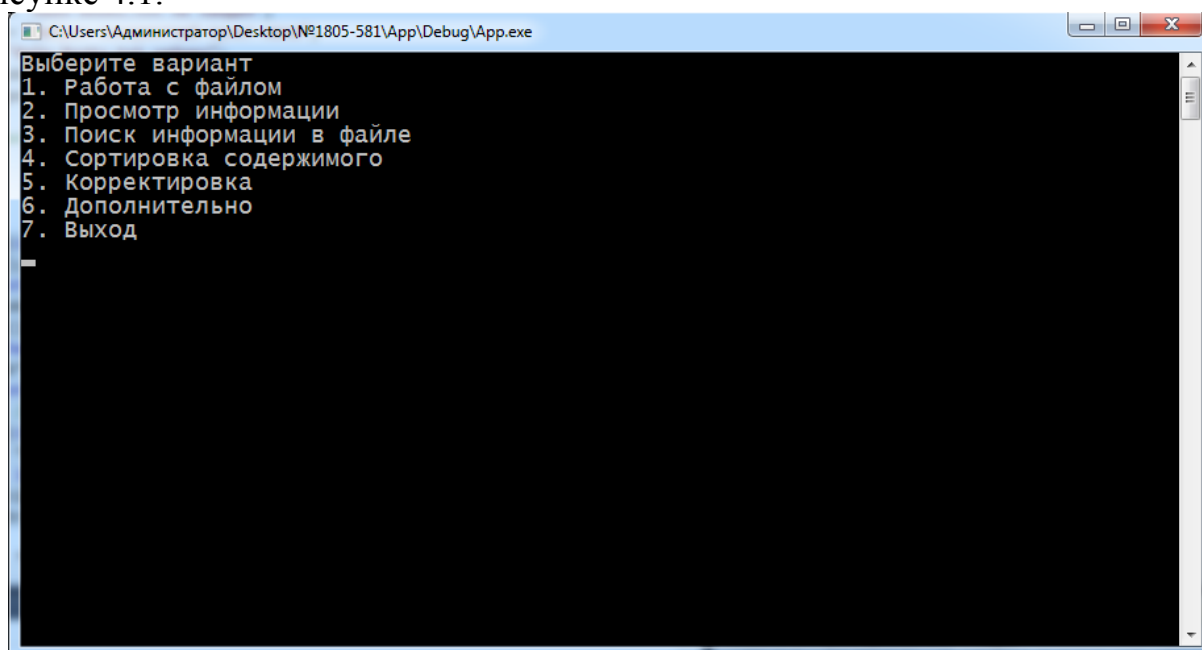
книги. После ввода новых данных, информация обновится и при просмотре будет выводиться на экране.

Функция `void delete_books(int nomer)` отвечает за удаление пользователей по номеру, который требуется удалить.

Функция `void prosmotr_books()` выводит все данные, которые существуют в массиве на экран.

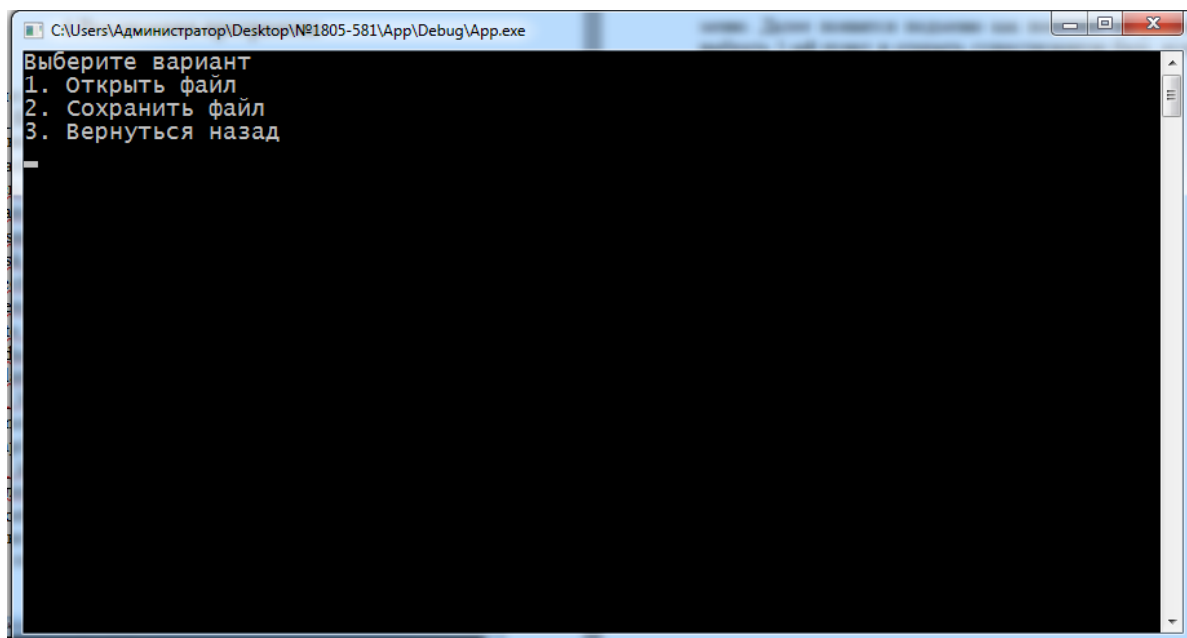
Функция `void sort(Exam* arr, int size)` будет производить сортировку данных. В данной сортировке будет применен метод Пузырька. Этот метод входит в состав медленных сортировок, специфика данной сортировки заключается в том, чтобы пересмотреть каждый элемент массива структуры.

При запуске программы перед пользователем появляется меню редактирования с соответствующими пунктами. Данное меню представлено на рисунке 4.1.



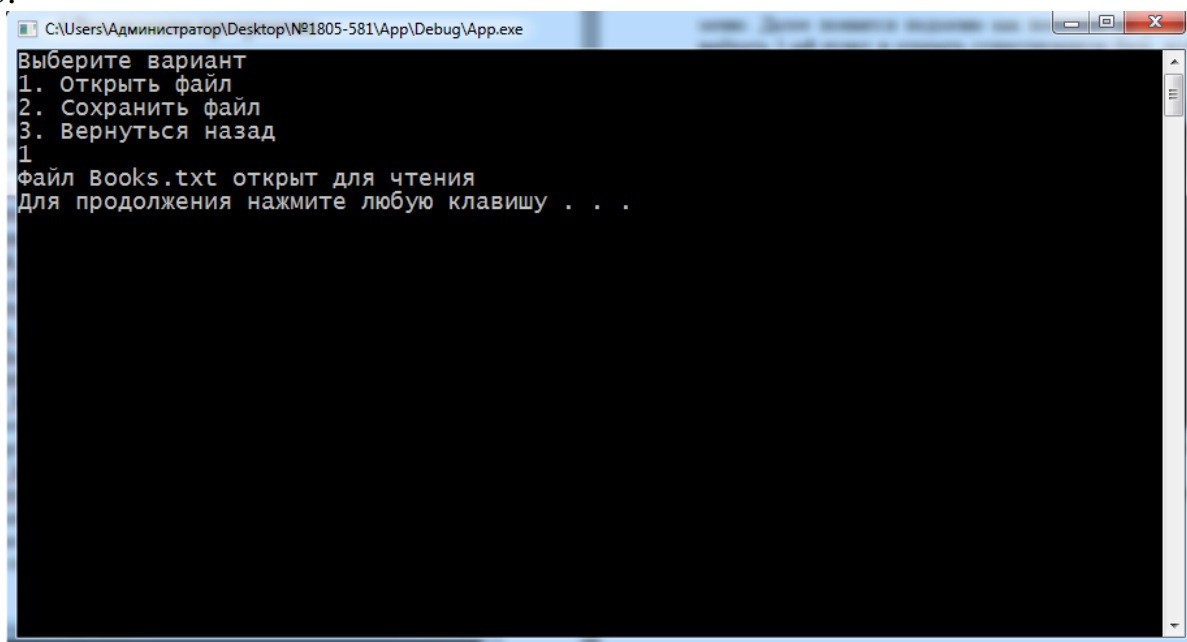
**Рисунок 4.1** – Меню редактирования

После того как пользователь начал процесс взаимодействия с программой, ему необходимо подгрузить базу данных выбрав 1-ый пункт меню. Далее появится подменю как показано на рисунке 4.2. Необходимо выбрать 1-ый пункт и открыть существующую базу, если такая есть, или же создать новую, выбрав пункт под номером 2.



**Рисунок 4.2** – Результат открытия 1-го пункта меню

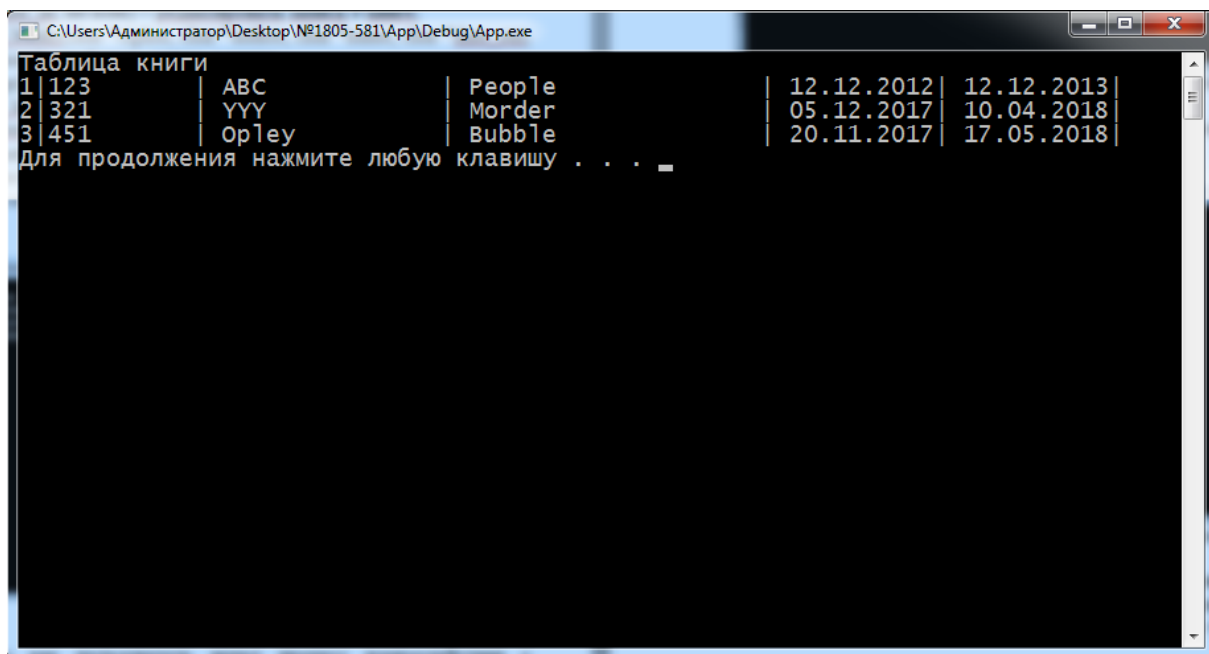
Результат открытия существующей базы данных представлен на рисунке 4.3.



**Рисунок 4.3** – Открытие базы данных

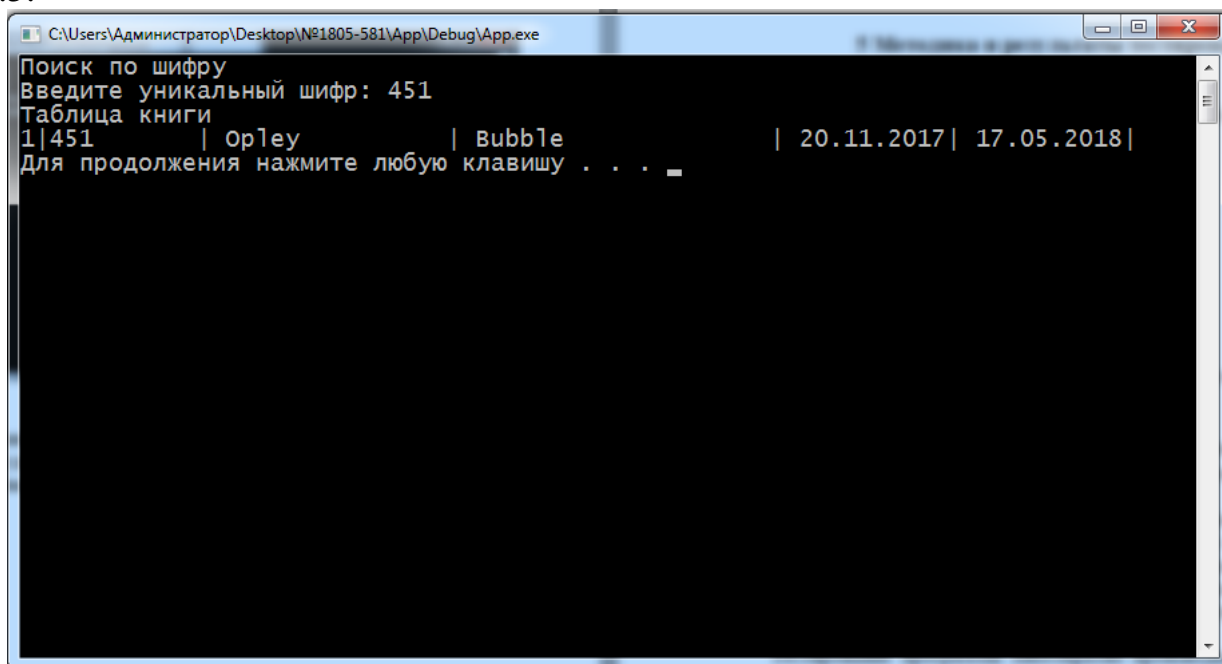
Далее с главного меню выбираем 2-ой пункт меню и просматриваем справочную информацию ранее открывшееся базы данных.

Результат выбора 2-го пункта меню редактирования представлен на рисунке 4.4.



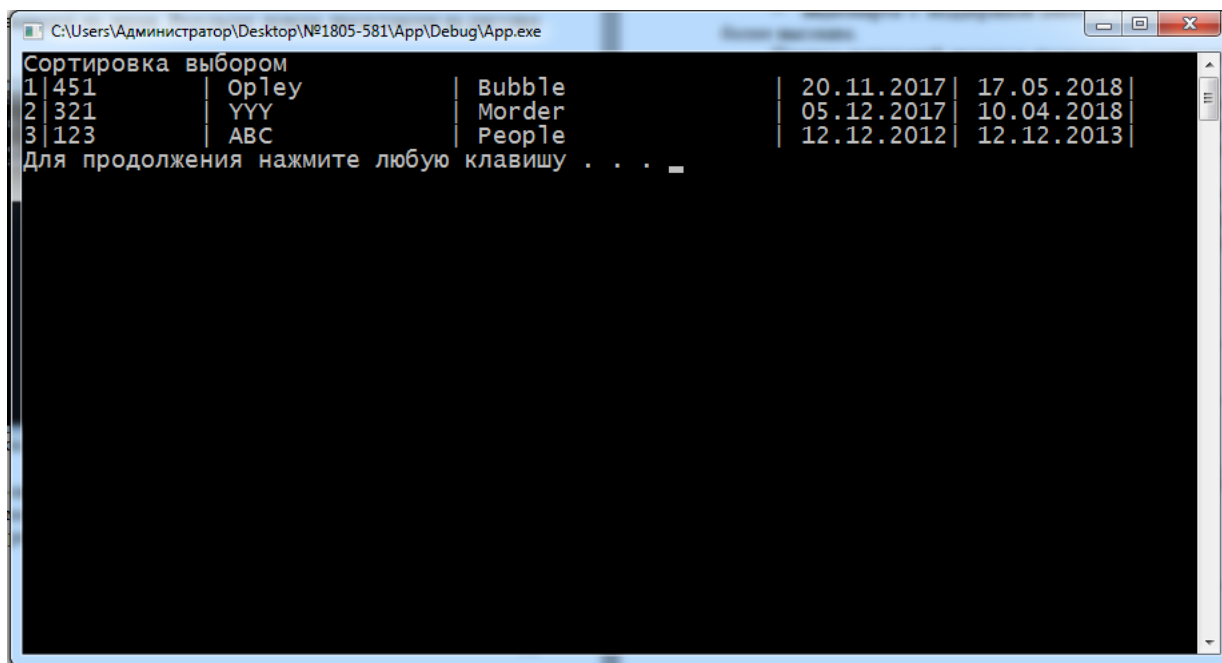
**Рисунок 4.4 – Просмотр информации**

Для поиска информации в базе необходимо выбрать третий пункт меню. Пользователю будет предложено ввести искомое значение, после чего результат будет выведен на экран. Результат поиска представлен на рисунке 4.5.



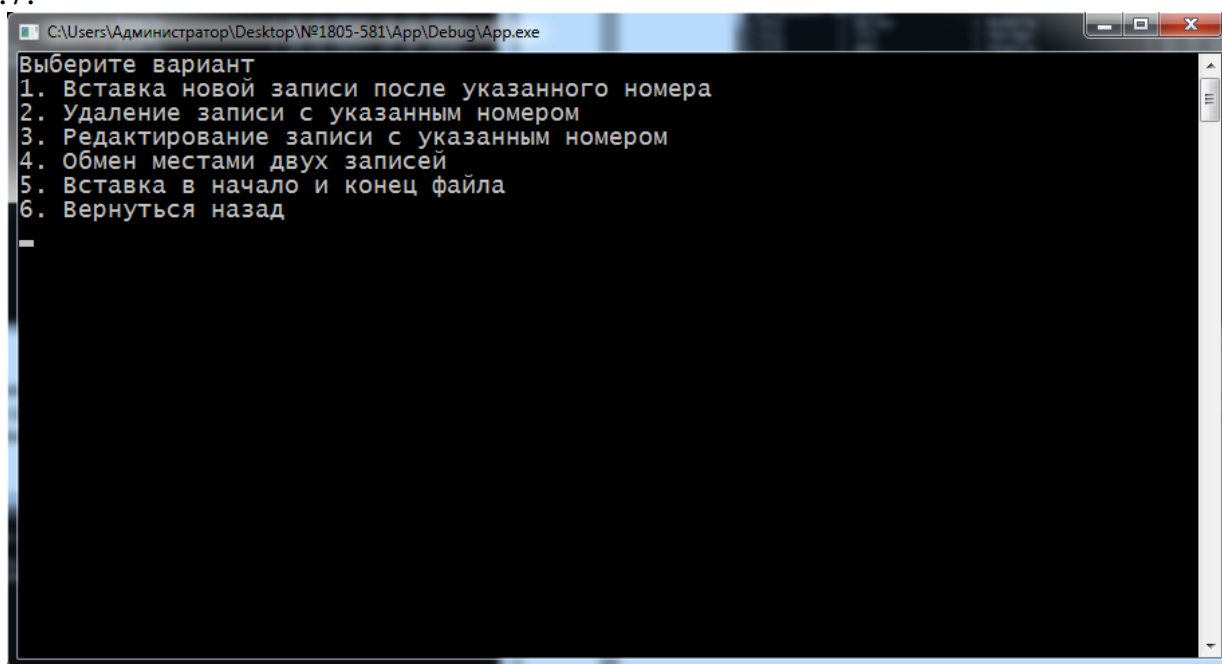
**Рисунок 4.5 – Результат поиска информации по шифру**

Для сортировки справочной информации в ранее открывшееся базы данных необходимо выбрать 4-ый пункт меню. После чего перед пользователем будет отображен результат, который представлен на рисунке 4.6.



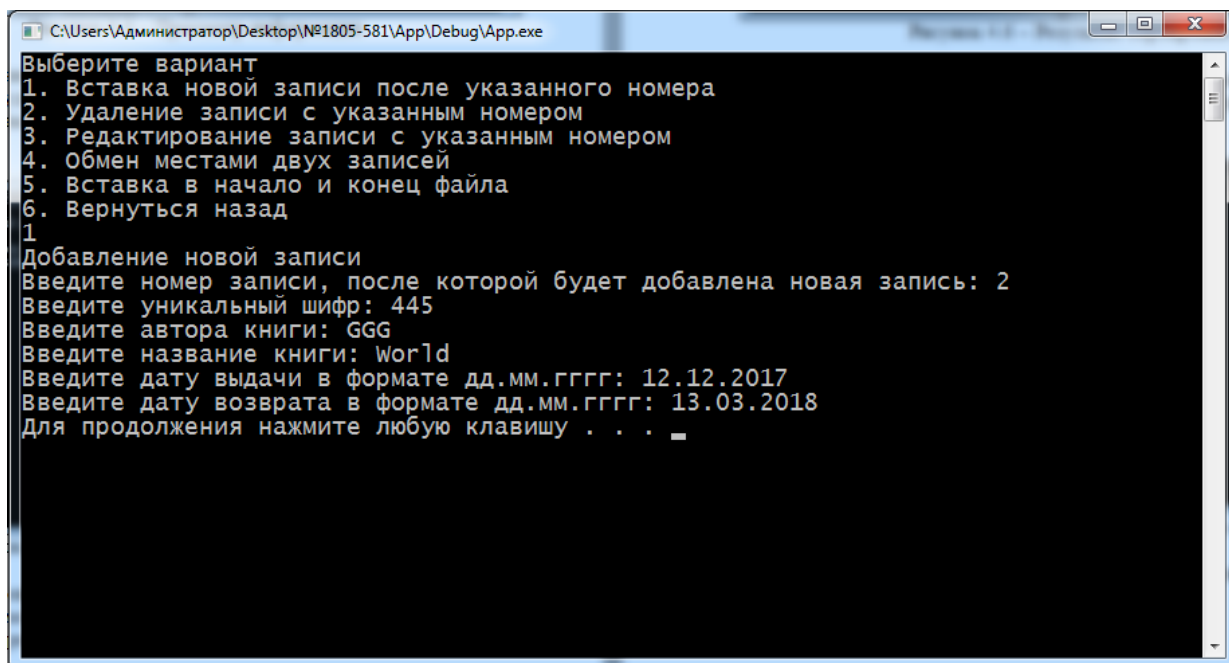
**Рисунок 4.6 – Результат сортировки**

Для корректировки информации необходимо выбрать 5-ый пункт меню. Откроется подменю с вариантами работы, которые представлены на рисунке 4.7.

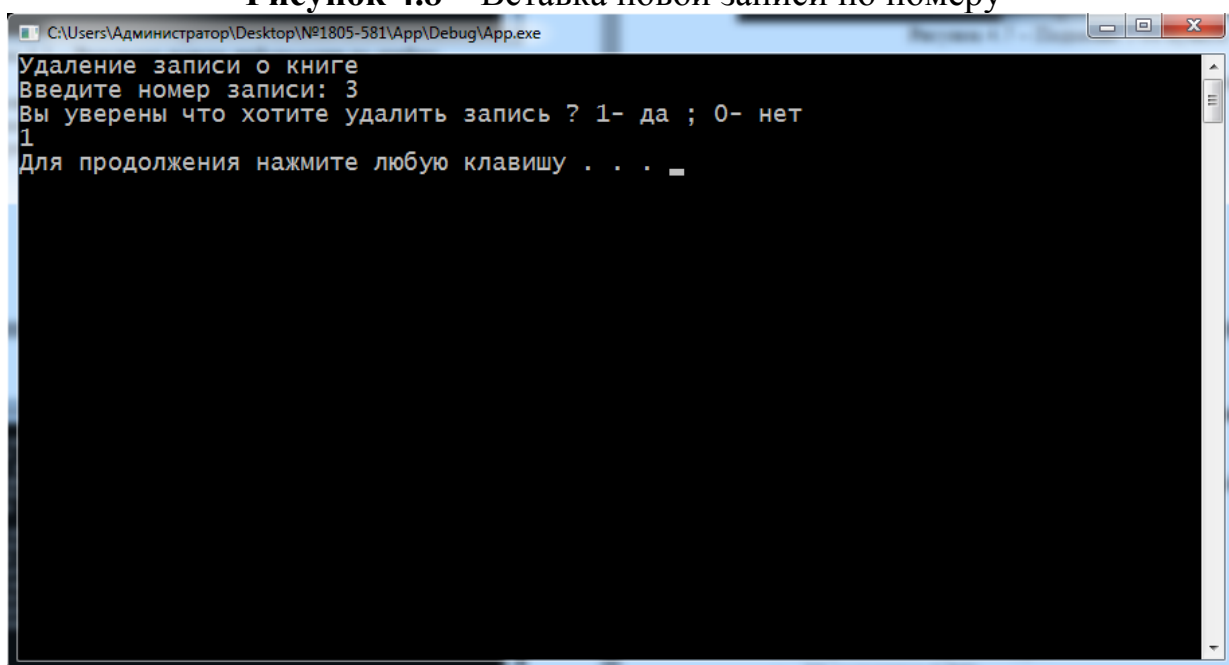


**Рисунок 4.7 – Подменю 5-го пункта**

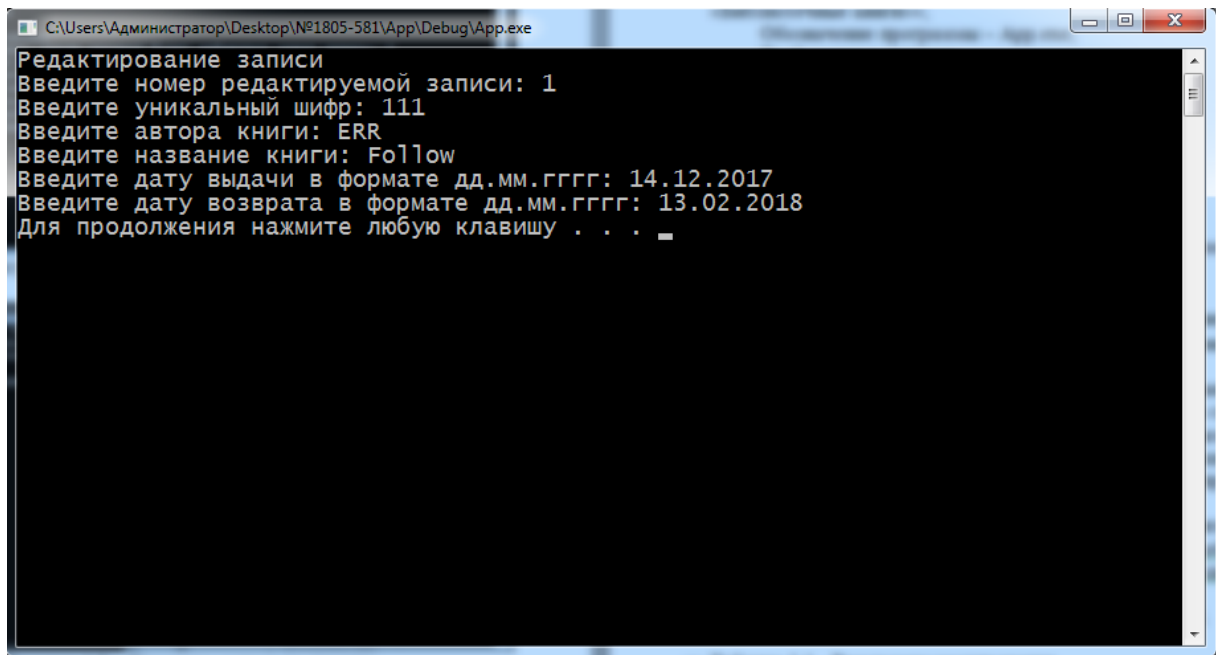
Результаты работы с данным подменю представлены на рисунках 4.8 – 4.12.



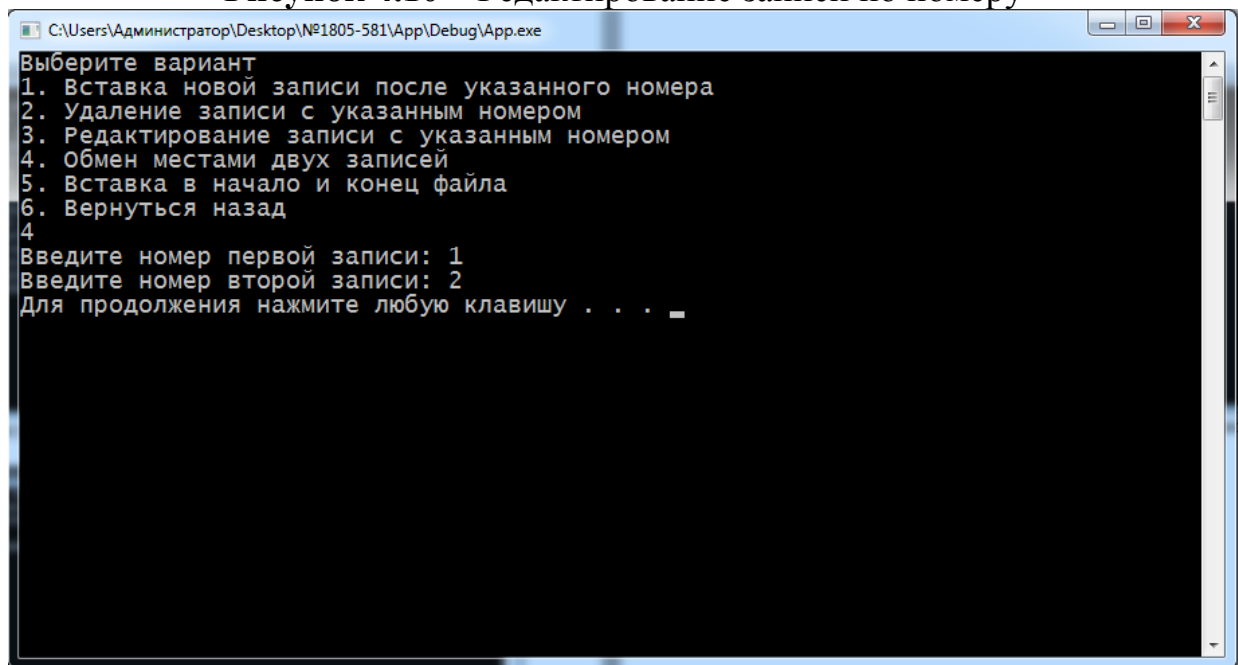
**Рисунок 4.8 – Вставка новой записи по номеру**



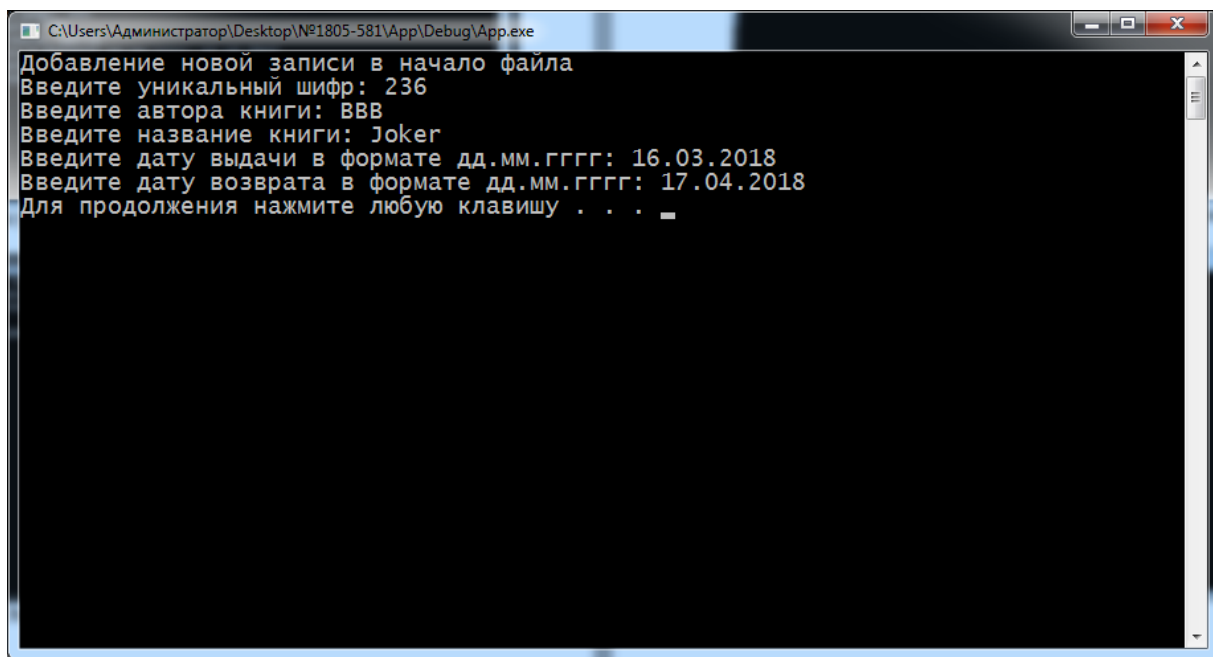
**Рисунок 4.9 – Удаление записи по номеру**



**Рисунок 4.10 – Редактирование записи по номеру**

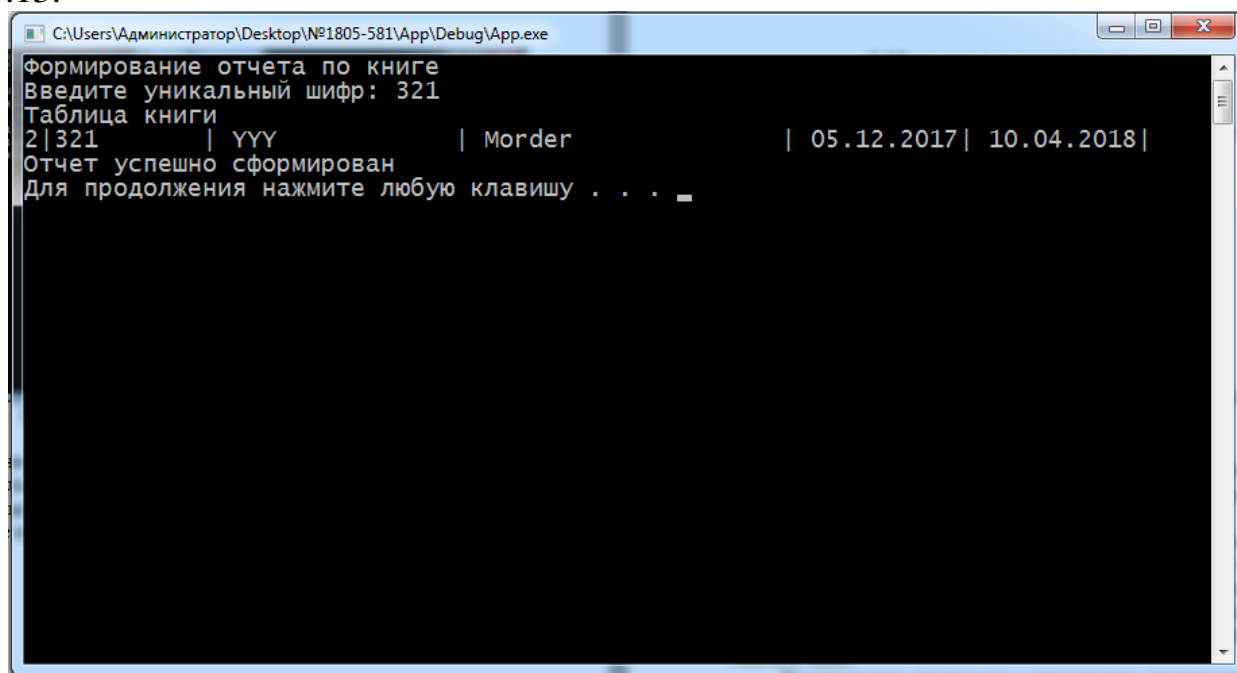


**Рисунок 4.11 – Обмен записей**



**Рисунок 4.12 – Вставка записи в начало**

Для выполнения дополнительного задания необходимо выбрать 6-ой пункт меню. Пользователю будет предложено ввести шифр, после чего отобразятся данные и будут записаны в текстовый файл. Данный файл будет храниться в корне проекта. Результат данного действия представлен на рисунке 4.13.



**Рисунок 4.13 – Формирование отчета**

## 4 Методика и результаты тестирования

Наименование программы – «Простейшая однотабличная база данных «Библиотечные книги»»;

Обозначение программы – App.exe;

Требования к системе:

Поддерживаемые операционные системы:

- Windows XP (архитектуры x86 и x64);
- Windows 7 (архитектуры x86 и x64);
- Windows 8 (архитектуры x86 и x64);

Поддерживаемые архитектуры:

- 32-разрядные (x86);
- 64-разрядные (x64).

Требования к оборудованию:

- процессор с тактовой частотой 1,6 ГГц или более быстрый;
- 1 ГБ ОЗУ (1,5 ГБ в виртуальной машине);
- 10 ГБ свободного места на жестком диске;
- жесткий диск со скоростью 5400 оборотов в минуту;
- видеокарта с поддержкой DirectX 9 и разрешением 1024 x 768 или более высоким.

Планом испытаний является проведение тестирования программы. Для проверки работы программы будут созданы разного рода ситуации, например: указание несуществующего файла базы данных или добавление новых записей с некорректно введенными данными, проверка функциональных клавиш перехода между пунктами меню и т.д. Тестирования программы подробно описано ниже.

Тестирование программы многократно производилось методом ввода случайных данных. При вводе значений пунктов меню, добавлении, удалении, редактировании записей, поиске, фильтрации, данные вводились как корректно, так, как требует этого программа, так и некорректно.

Результаты тестирования приведены в таблице 5.1.

**Таблица 5.1 - Результаты тестирования**

Описание теста	Выполняемые действия	Реакция программы	Вывод
Запуск программы	Активирован исполняемый файл программы	Появился список возможных действий	Выполнено
Открытие базы данных	Указали имя файла для открытия	Отобразился список возможных действий над открытой базой данных	Выполнено



Попытка загрузить базу из некорректного файла	Указали имя несуществующего файла	База не загрузилась. Система выдала сообщение о том, что файл с таким именем не существует. Было предложено создать новый файл. Программа продолжает функционировать	Выполнено
Добавление новой записи	Корректно ввели данные	В списке появилась новая запись	Выполнено
Добавление новой записи с некорректно введенными данными	Данные введены некорректно	Запись не добавлена. Программа сообщает об ошибке. Предложено повторить ввод.	Выполнено
Удаление записи	Указали место удаления	Запись исчезла из выбранной позиции в списке	Выполнено
Поиск в базе	Ввели значение нужного поля	Соответствующие результаты выводятся на экран удовлетворяющие запросу	Выполнено
Поиск в базе при некорректно введенных данных	Там, где требуется ввод числа, ввели строку	Программа сообщает об ошибке ввода. Предложено повторить ввод	Выполнено
Вывод базы на экран	Ввели нужную команду	Содержание базы появилось на экране	Выполнено
	Ввели значение в несколько полей	Вывела список с совпадающими требованиями	Выполнено

## Окончание таблицы 5.1

Работа с запросами при некорректно введенных данных	Данные введены некорректно	Программа сообщает об ошибке. Предложено повторить ввод	Выполнено
Сортировка	Ввели нужную команду. Выбрали нужную категорию и направление	Программа выполнила сортировку по необходимому полю и в указанном направлении	Выполнено
Редактирование	Ввели нужную команду. Выбрали необходимую запись, и ввели данные в каждое поле	Программа заменила каждое поле в записи на введенное пользователем.	Выполнено
Закрытие файла	Указали нужную команду	Файл закрыт.	Выполнено

В ходе тестирования программы были найдены и устранены ошибки, которые приводили к неправильной работе всей программы или неправильным действиям. Были проверены следующие функции:

- Добавление новой записи.
- Удаление записи.
- Поиск в базе.
- Вывод базы на экран.
- Сортировка.
- Редактирование.

После выполнения тестирования программа готова к использованию.

## Заключение

В результате выполнения курсовой работы, была разработана простейшая однотабличная база данных в предметной области «Библиотечные книги». Результат работы имеет консольный вариант, который выполняет весь необходимый функционал в соответствии с поставленной задачей. Пользователь с лёгкостью сможет ознакомиться со всей необходимой ему информацией из приложения, процесс работы законченного приложения был продемонстрирован в пояснительной записке.

При выполнении курсовой работы были закреплены навыки работы на языке программирования C++ и приобретены навыки написания консольных приложений. Сделаны соответствующие выводы о необходимости внедрения информационных технологий в различные сферы жизнедеятельности, о том, как это действительно упрощает работу с информацией.

Программа имеет специфическую направленность – вся работа ведётся над информацией о библиотечных книгах. Поэтому использовать её сможет узкий круг людей, но для них она окажется очень полезной.

					Орлов Дмитрий Сергеевич 1710106	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

## Список используемых источников

1. В. В. Подбельской, С. С. Фомин. Программирование на языке Си++. М.: ФиС, 2012, 600 с. (рекомендуемый учебник по направлению «Информатика и ВТ»).
2. Х. М. Дейтел., П. Дж. Дейтел. Как программировать на Си++. М.: ЗАО БИНОМ, 2007, 1000 с. (первые 600 страниц - изложение обычного Си, наиболее полно соответствует структуре курса для 1-2семестра).
3. У. Топп, У. Форд. Структуры данных в Си++. М.: ЗАО БИНОМ, 2011, 800 с. (наиболее полно соответствует структуре курса для 3-4 семестра).
4. Климова Л. М. Основы практического программирования на языке Си++. М.: Издательство ПРИОР, 2014, 464 с.
5. Романов Е.Л. Язык Си. Структуры данных. Конспект лекций. Тестовые вопросы и задания к лабораторным работам. – Новосибирск, Изд-во НГТУ, 2012, -80 с. («методичка» для 2,3-го семестра, библиотека НГТУ – 73Р693)
6. Т. Фейсон. Объектно-ориентированное программирование на Borland C++ 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
7. Т. Сван. Освоение Borland C++ 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
8. Г. Шилдт. Самоучитель C++: Пер. с англ. - Санкт-Петербург: BHV-Санкт-Петербург, 1998. 620с.
9. К. Джамса. Учимся программировать на языке C++: Пер. с англ. - Москва: Мир, 1997. 320с.
10. Свободная энциклопедия [Электронный ресурс] – 2014. - Режим доступа: <http://ru.wikipedia.org> – Дата доступа: 15.05.2018.

ПРИЛОЖЕНИЯ

					ОДС.1710106.ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Однотабличная база данных "Библиотека книг" (пояснительная записка)	для	Лист	Листов
Разраб.		Орлов Д.С.					22	9
Провер.		Тропникова В.А.						
Реценз.						Учреждение образования «Полоцкий государственный университет» гр.17-ИТд		
Н. Контр.								
Утверд.								

# Приложение А

## (обязательное)

### Блок-схемы алгоритмов

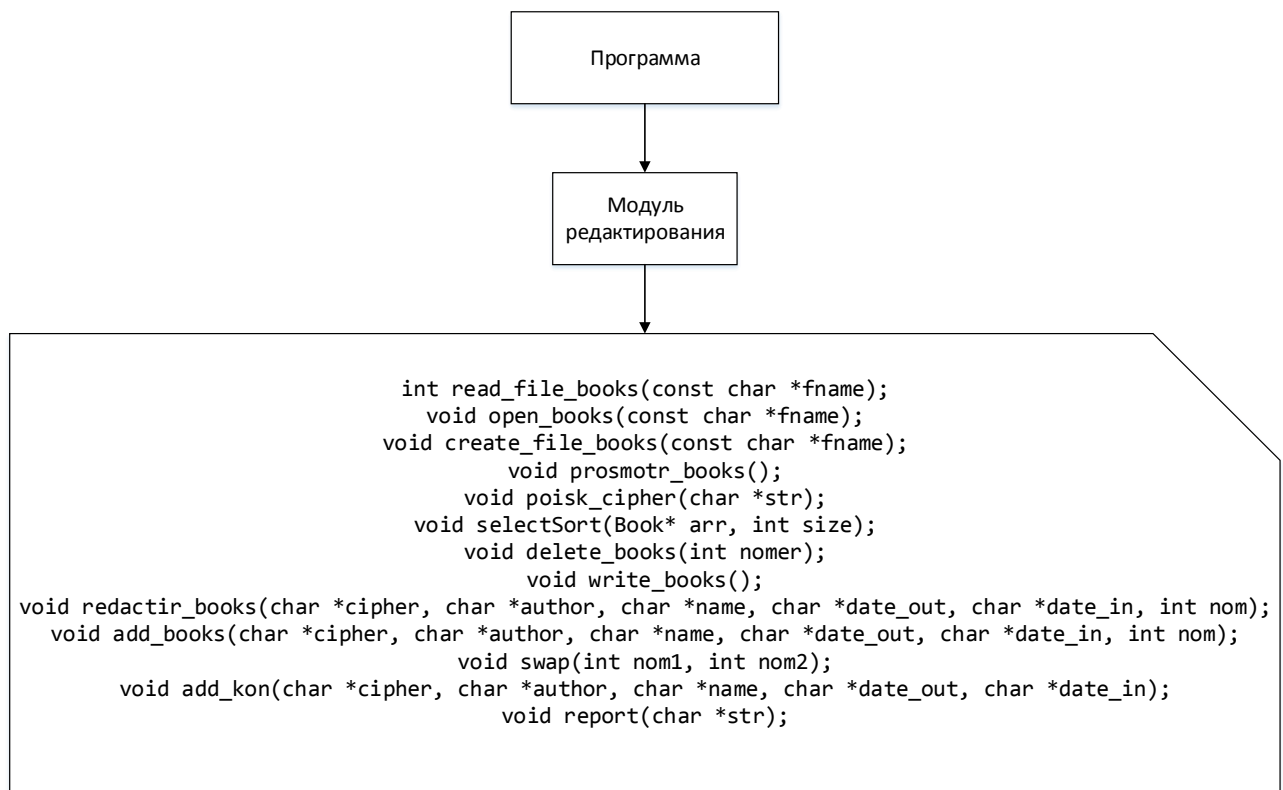


Рисунок А.1 – Структура программы

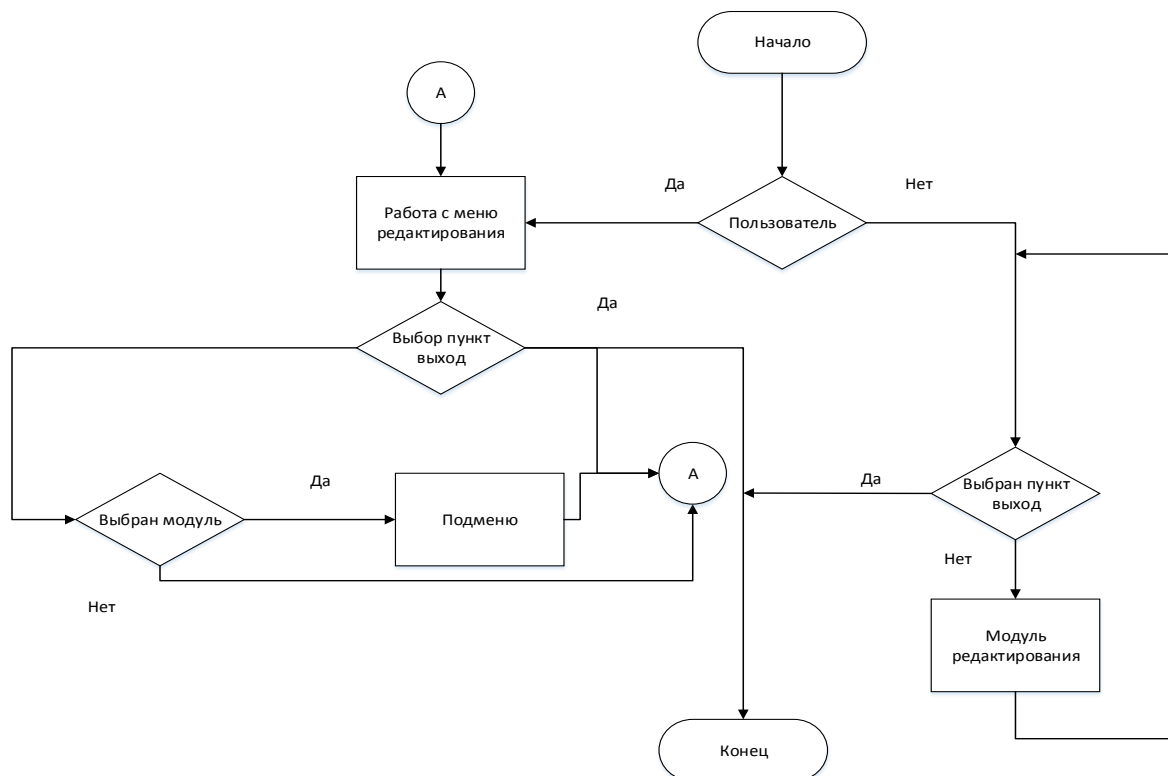


Рисунок А.2 – Обобщённая схема алгоритма решения задачи

## Приложение Б (обязательное) Интерфейс программы

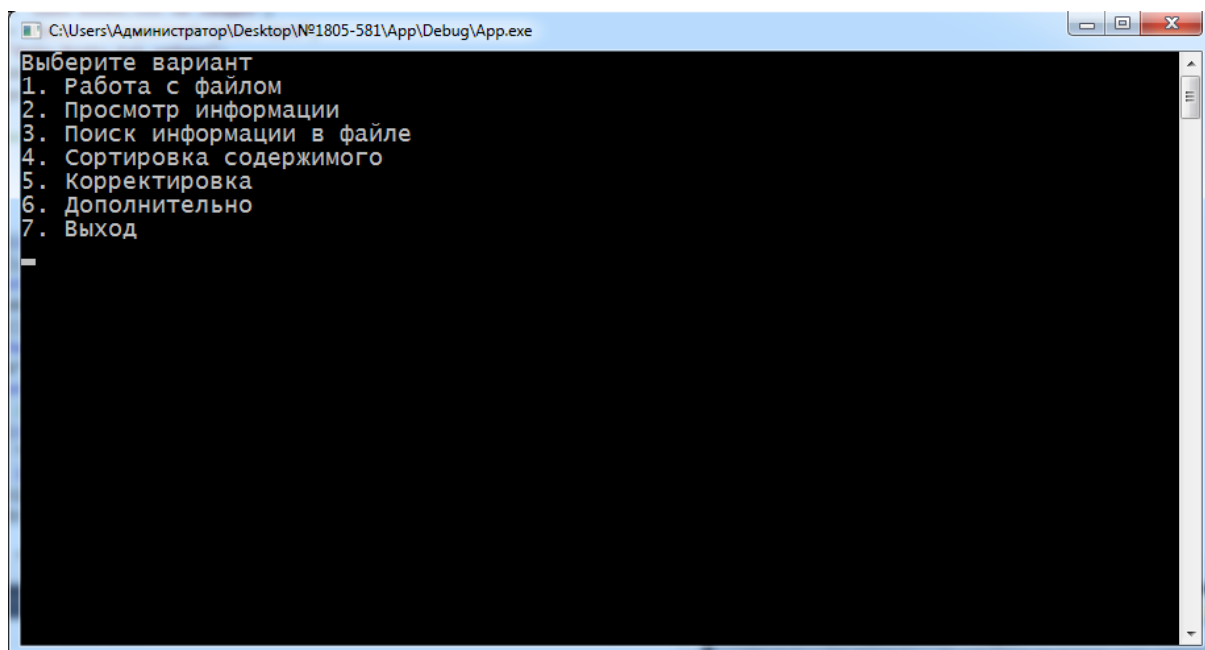


Рисунок Б.1 – Меню редактирования

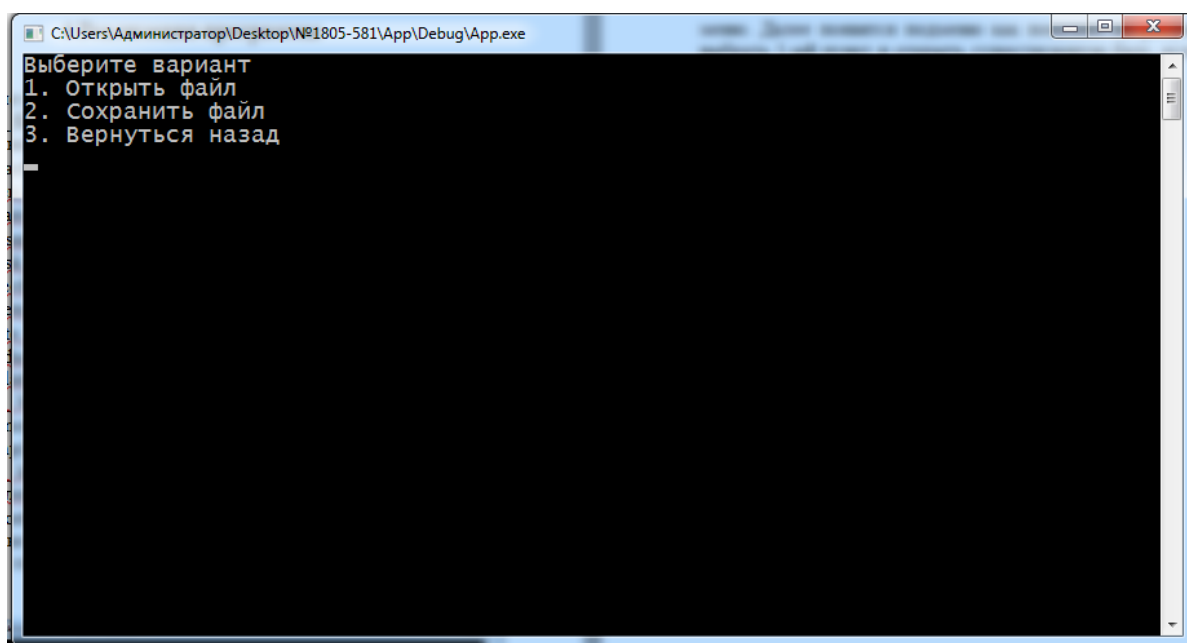


Рисунок Б.2 – Результат открытия 1-го пункта меню

Изм.	Лист	№ докум.	Подпись	Дата

ОДС.1710106.ПЗ

Лист

2

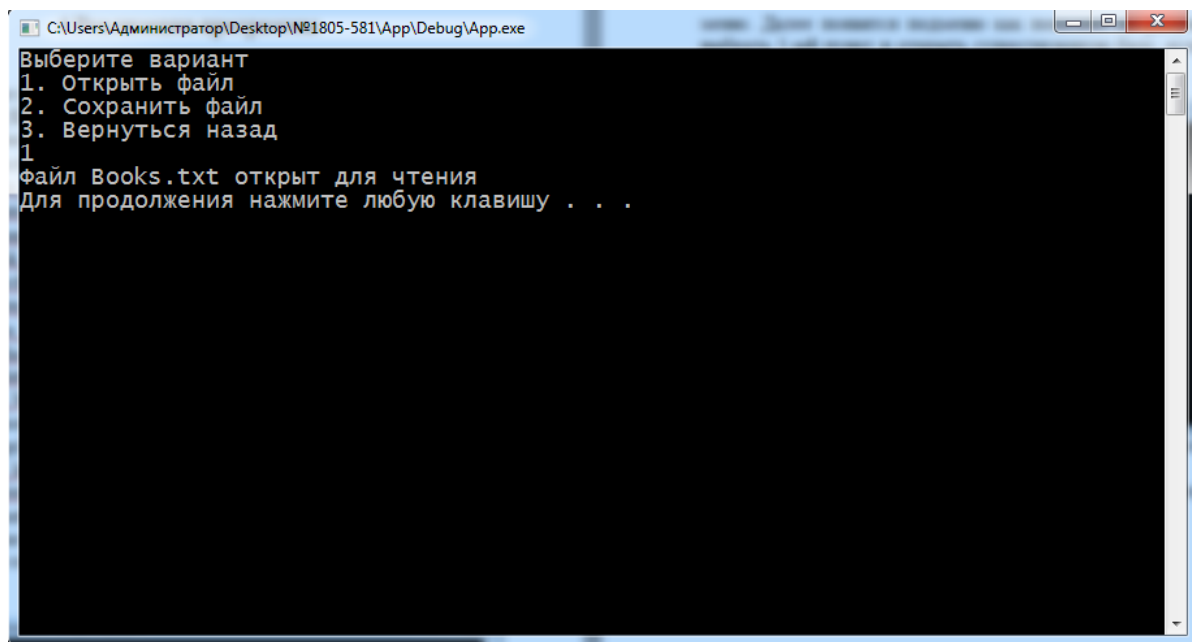


Рисунок Б.3 – Открытие базы данных

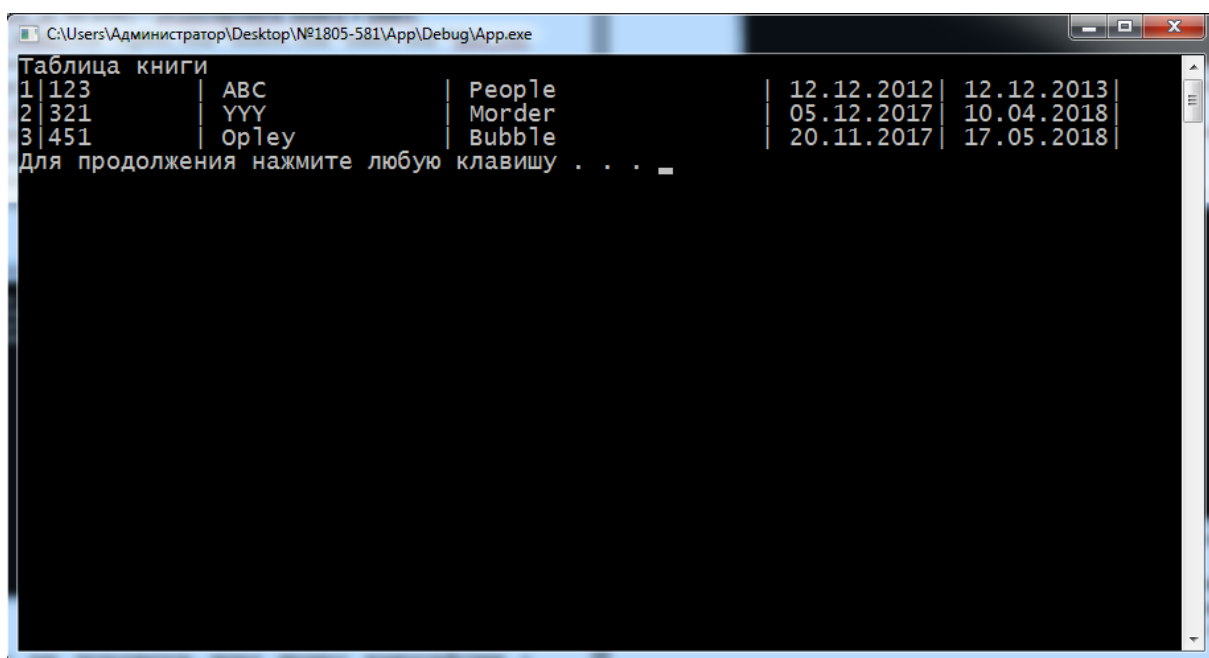
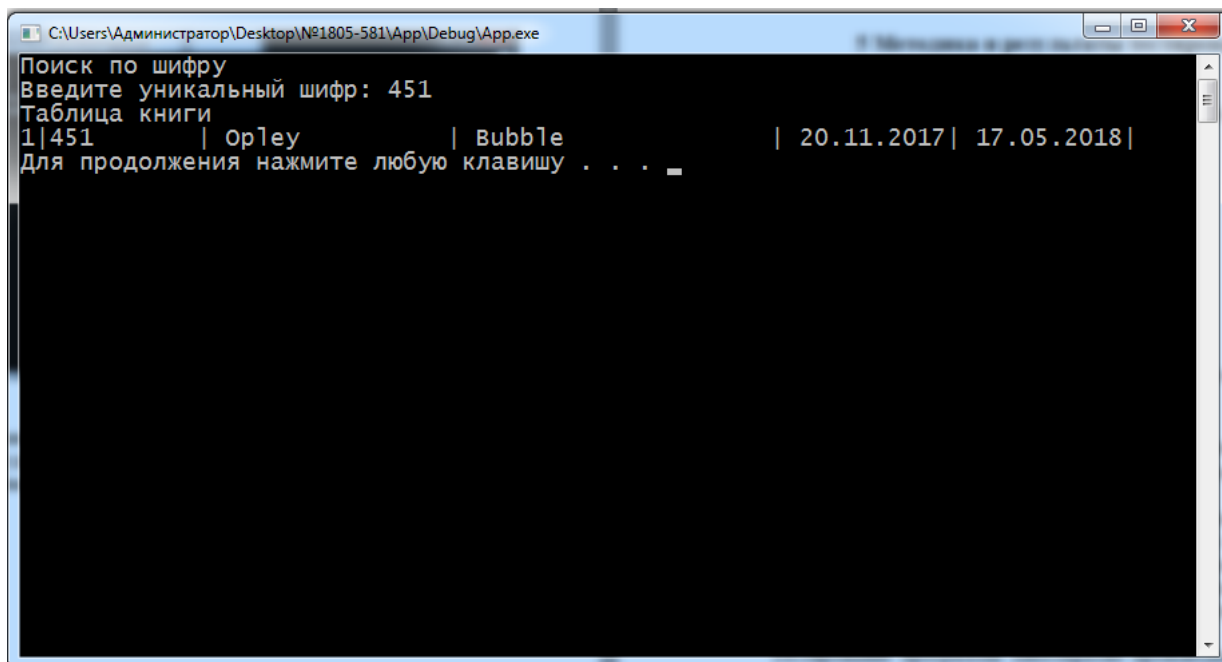
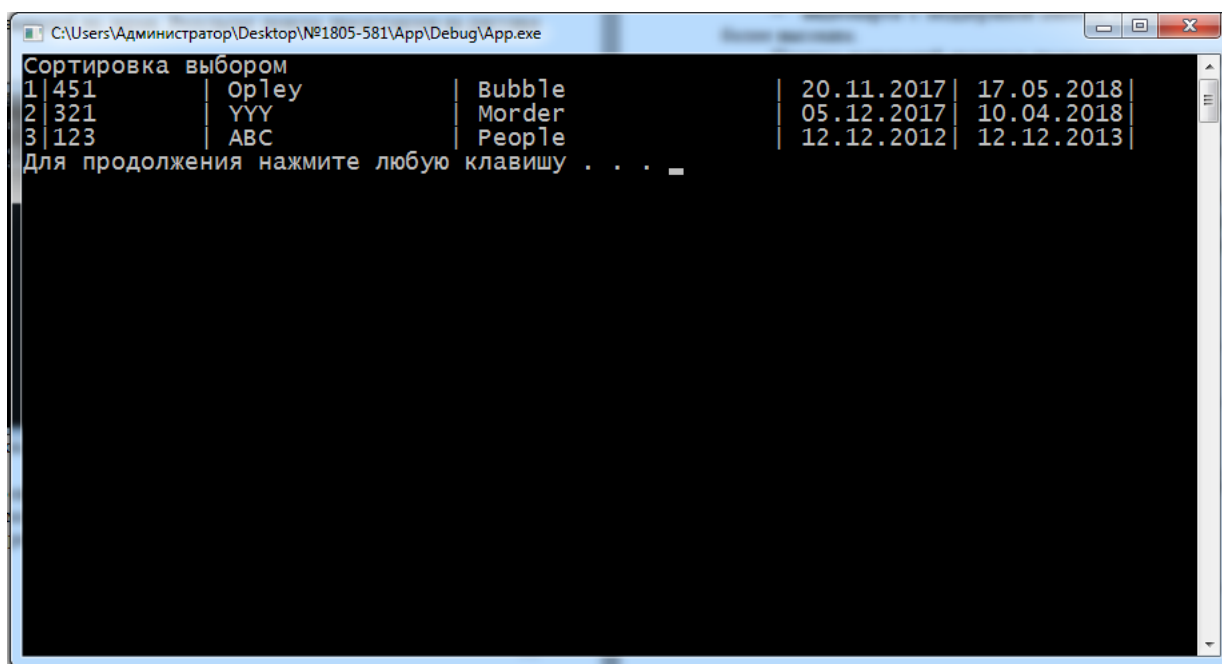


Рисунок Б.4 – Просмотр информации





**Рисунок Б.5** – Результат поиска информации по шифру



**Рисунок Б.6** – Результат сортировки

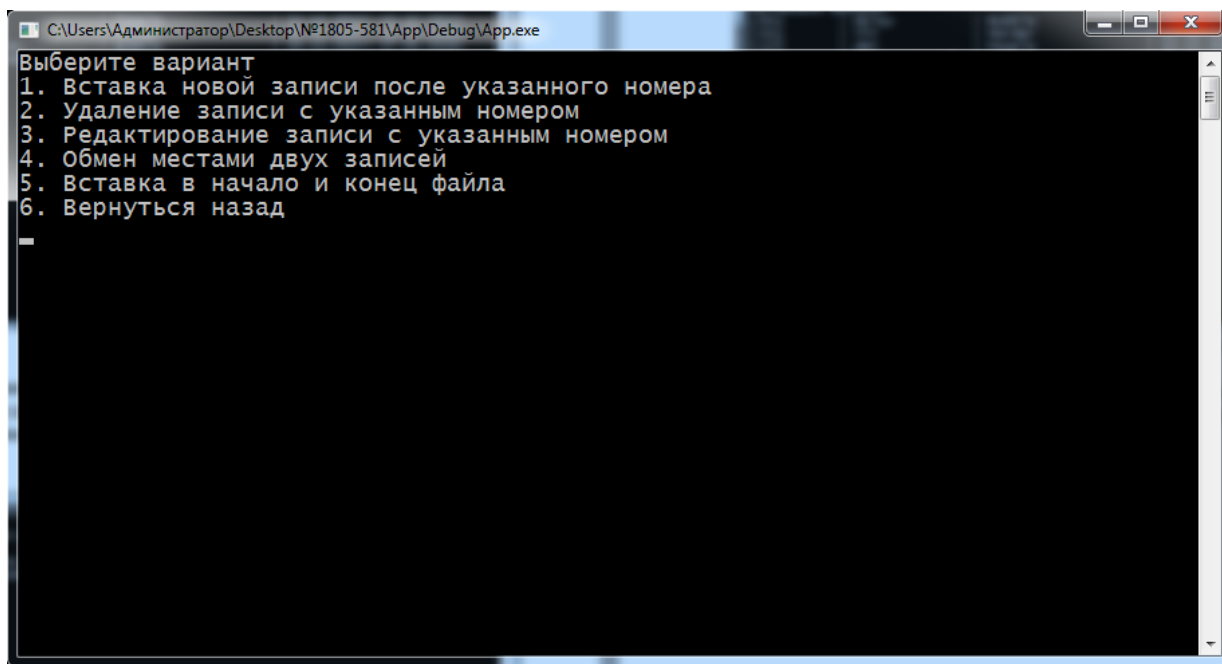


Рисунок Б.7 – Подменю 5-го пункта

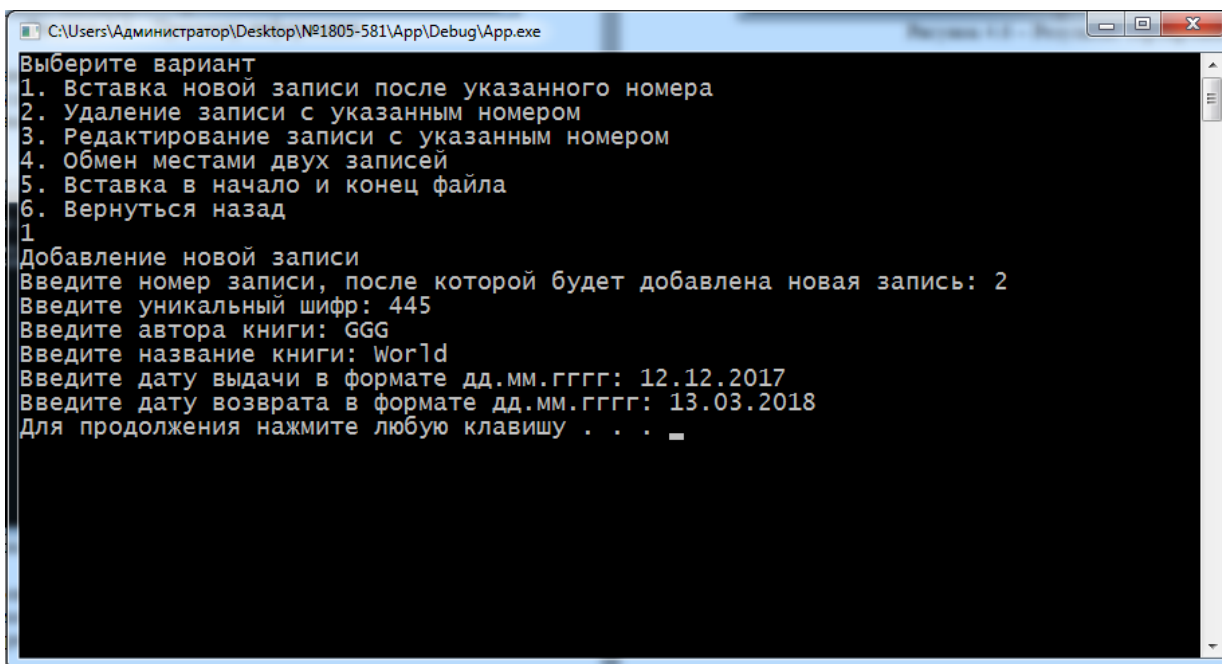
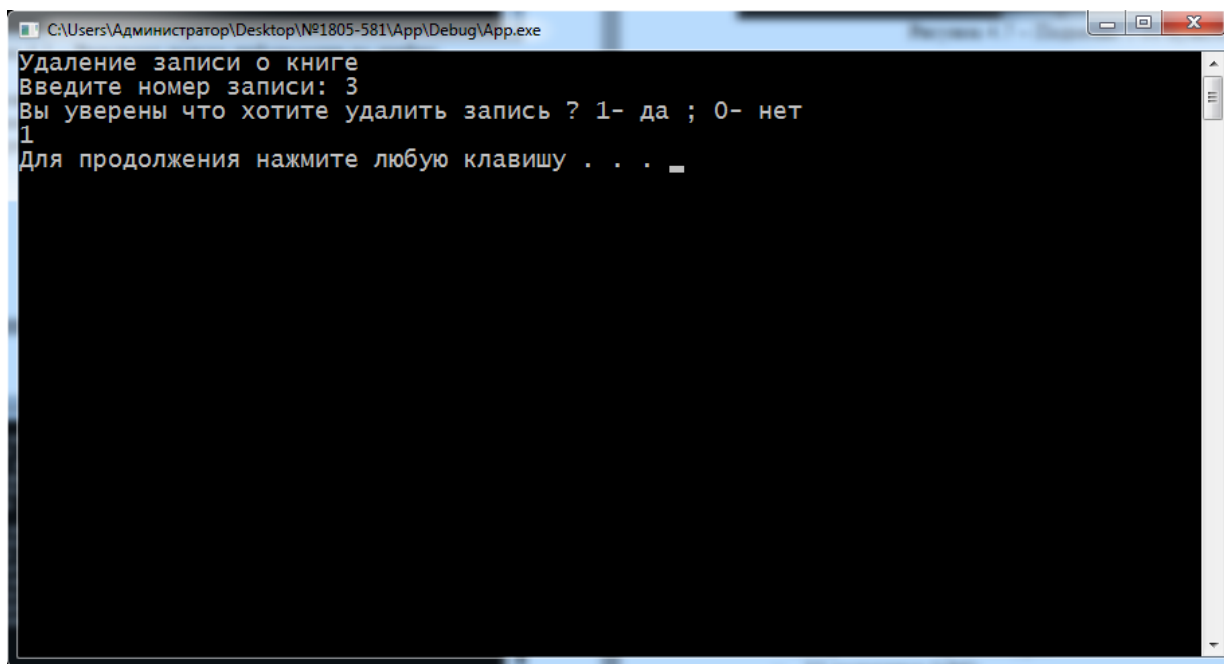
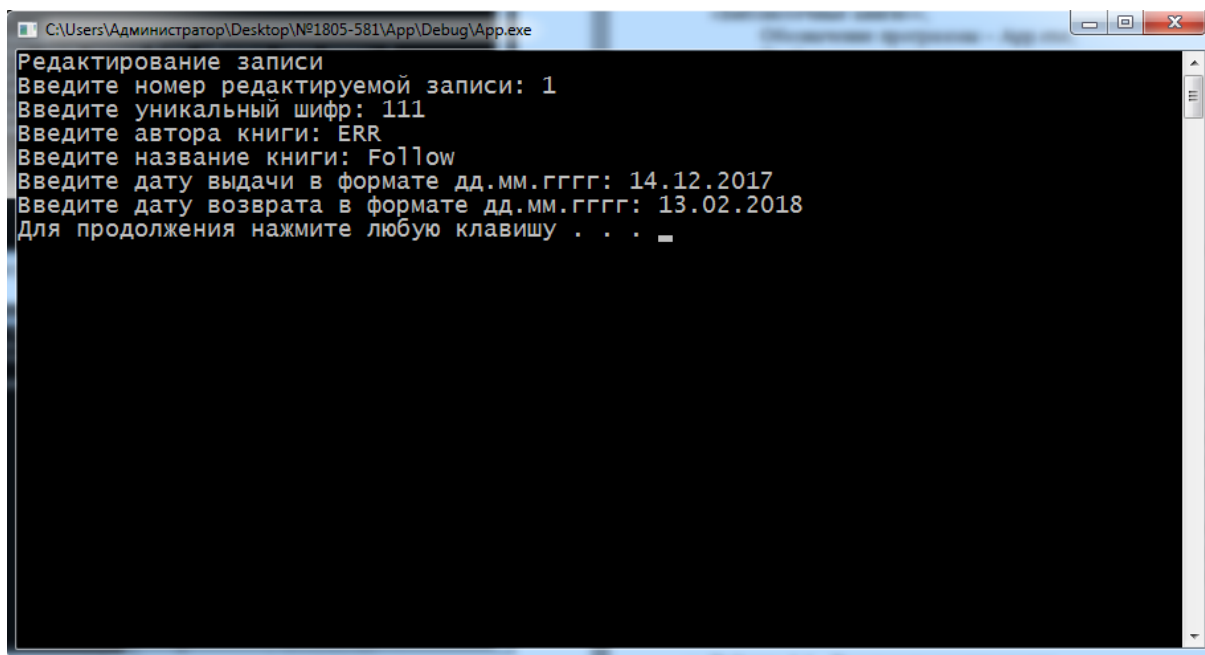


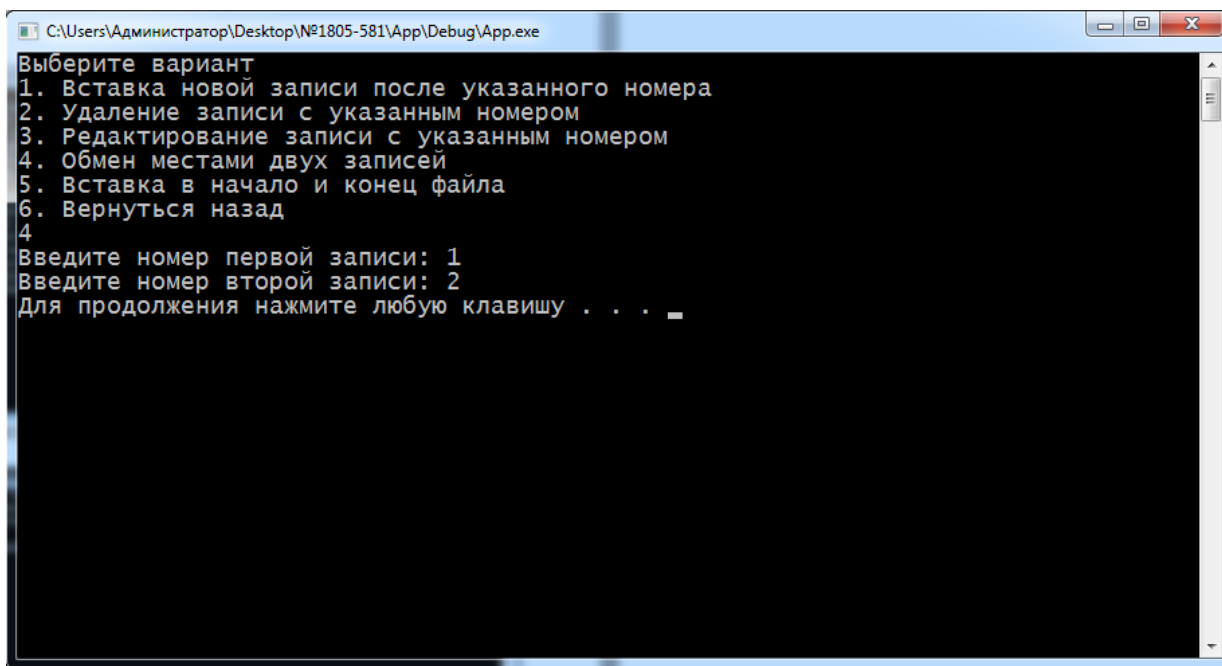
Рисунок Б.8 – Вставка новой записи по номеру



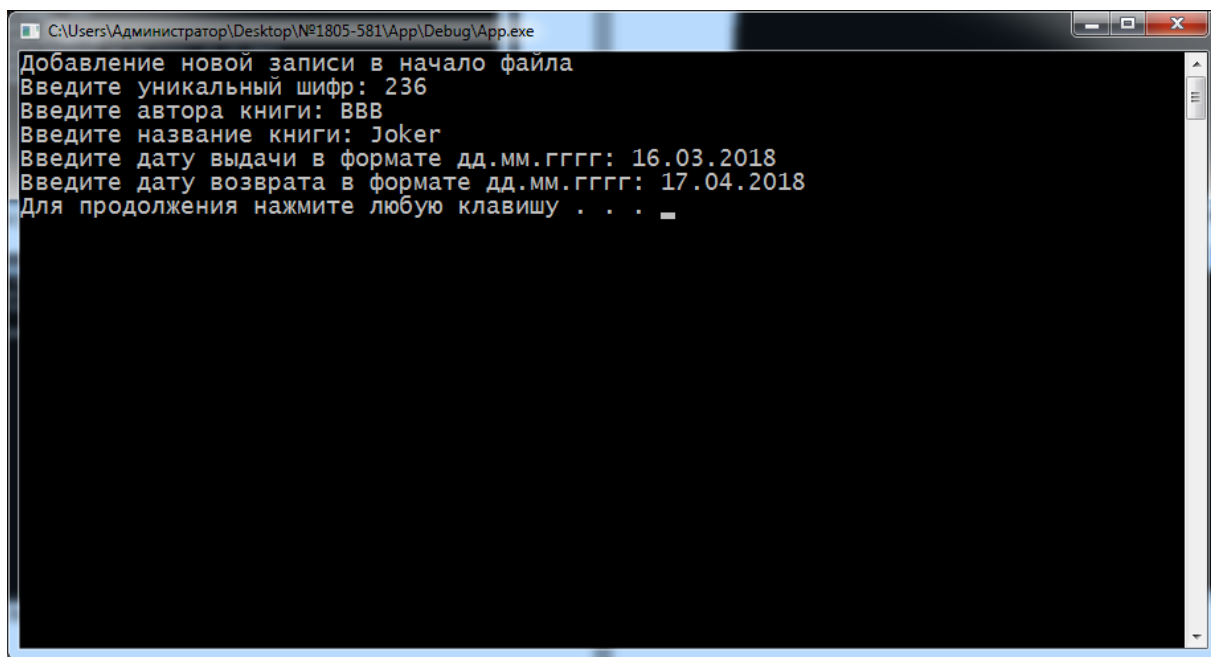
**Рисунок Б.9 – Удаление записи по номеру**



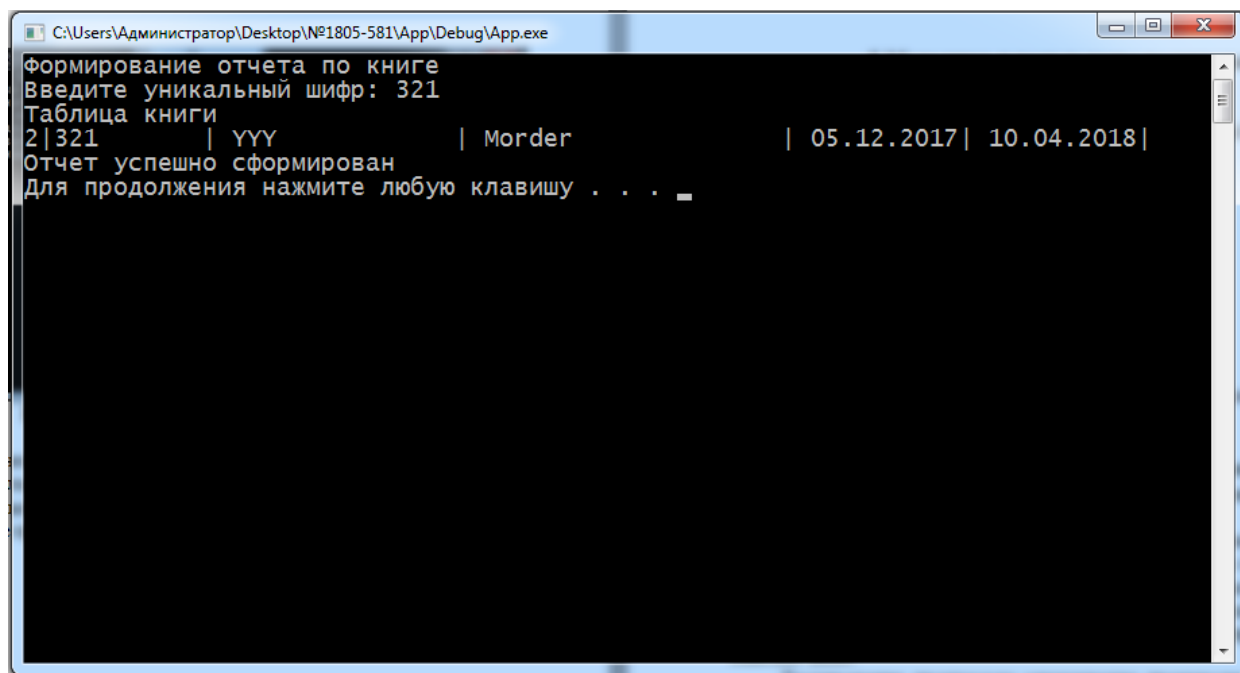
**Рисунок Б.10 – Редактирование записи по номеру**



**Рисунок Б.11 – Обмен записей**



**Рисунок Б.12 – Вставка записи в начало**



**Рисунок Б.13 – Формирование отчета**

**Приложение В**  
(обязательное)  
**Листинг программы**

Листинг программы находится на диске в файле file.docx

					ОДС.1710106.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9