

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Пензенский государственный университет»
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по курсу «Программирование»
на тему «Разработка программы сложной структуры методом
нисходящего программирования. Географическая база данных.»

Отмечено
21.05.2026
Г. Генералова А.А.

Выполнил:

студент группы 25BBB2

Никоноров С.Д.

Принял:

к.т.н., доцент Генералова А.А.

Пенза 2026

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

М.А. Митрохин

«11» февраля 2026 г.

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Хименкову Семёну Дмитриевичу Группа 258802

Тема проекта: **Разработка программы сложной структуры методом нисходящего программирования.**

Исходные данные (технические требования) на проектирование

Программа „Геометрическая база данных“
Добавление, удаление, редактирование, поиск, сортировка записей;
Фильтрация записей по определённым критериям;
Создание новой базы данных; Сохранение текущей базы данных;
Загрузка сохранённых баз данных.

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio.

Объем работы по курсу

1. Расчетная часть

Разработка программы.

2. Графическая часть

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания " 11 " февраль 2026г.

Дата защиты проекта " 21 " март 2026г.

Руководитель И. И. Плещинский

Задание получил " 11 " февраль 2026г.

Студент И. И. Плещинский

Содержание

Введение	5
Постановка задачи.....	6
Выбор решения.....	7
Описание разработки программы	8
Отладка и тестирование	9
Описание программы	9
Руководство пользователя	23
Заключение	31
Список используемых источников.....	31
Приложение А. Листинг Программы.....	32
Приложение В. Результаты работы программы	54
Приложение С. Результат работы программы.....	54

Введение

В современном мире информация играет ключевую роль в жизни человека. С каждым днём объёмы данных растут, и для их хранения и обработки требуются специальные программные решения. Одним из таких решений являются базы данных.

База данных – это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации. Существует множество систем управления базами данных (СУБД), но для понимания принципов работы с данными полезно реализовать собственную простую базу данных в виде программы на языке С.

Данная курсовая работа посвящена разработке программы для ведения базы данных стран мира. Программа позволяет пользователю хранить подробную информацию о различных государствах: название, столицу, население, площадь, язык, валюту, ВВП (внутренний валовый продукт), ИЧР (индекс человеческого развития) и многие другие параметры.

Актуальность работы заключается в том, что созданная программа демонстрирует практическое применение навыков программирования на языке С, работы с динамическими структурами данных, файлового ввода-вывода и даже использования ассемблерных вставок. Программа может использоваться в учебных целях или как основа для более серьёзного проекта.

Особенностью данной работы является комплексный подход: программа не просто хранит данные, но и предоставляет удобный интерфейс для добавления, редактирования, удаления, поиска, фильтрации и сортировки записей, а также сохраняет данные в файл и загружает их оттуда.

Постановка задачи

Необходимо разработать программу для работы с базой данных стран мира. Программа должна иметь интуитивно понятный интерфейс для пользователя. Необходимо определить названия пунктов меню, построить алгоритмы работы с массивом структур для фильтрации и сортировки записей по определённым критериям, изучить функции для работы с файлами, чтобы сохранять базу данных в файл и загружать из файла.

Многомодульность программы. Программа должна быть поделена на логические модули. Это упростит поиск ошибок при отладке и тестировании программы, а также позволит легко расширять функционал программы.

Использование сложных типов данных – массивов, структур, файлов. Это необходимо для более простой и интуитивной обработки данных в коде программы.

Режим работы видеосистемы – текстовый. Необходимо определиться с типом интерфейса и с элементами управления, затем изучить способы их реализации.

Устройство ввода-вывода – клавиатура. Необходимо различать и идентифицировать действия, произведённые с её помощью, это облегчит использование программы.

Интерфейс должен быть построен на основе меню. Это необходимо для создания интуитивно понятного интерфейса для пользователя.

Заставка необходима для того, чтобы пользователь, запустивший программу, смог получить достаточную информацию о ней.

Выбор решения

Программа реализована на языке программирования C, используя при этом интегрируемую среду разработки Microsoft Visual Studio 2022 и вставку на языке Assembler.

1. Добавление записи. Пользователь вводит данные, необходимые для нового элемента базы данных в порядке, указанном в программе. Программа добавляет новую запись в базу данных.

2. Вывод записей. Пользователь выбирает один пункт из нового меню:

- 2.1 Показать все записи (краткая таблица)
- 2.2 Показать расширенную таблицу
- 2.3 Полный просмотр

3. Фильтрация записей. Пользователь выбирает критерий фильтрации:

- По населению
- По ВВП
- По площади
- По валюте
- По ИЧР

4. Сортировка записей. Пользователь выбирает параметр сортировки:

- По населению (по возрастанию)
- По ВВП (по убыванию)
- По площади (по убыванию)
- По ИЧР (по возрастанию)

5. Поиск страны. Пользователь вводит название страны для поиска.

6. Удаление записи. Пользователь вводит индекс записи, которую необходимо удалить.

7. Хранение информации о странах – программа должна хранить следующие сведения о каждой стране:

8. Редактирование записи. Пользователь вводит индекс записи, которую необходимо изменить, и вводит новые данные.

9. Полнофункциональное меню – пользователь должен иметь возможность выбирать действия с помощью клавиатуры, вводя соответствующие цифры.

10. Работа с файлами – сохранение текущей базы данных в бинарный файл и загрузка из файла.

11. Ассемблерная вставка – кэш шифрование данных.

12. Демонстрационные данные – возможность загрузки предварительно заполненной базы данных из всех стран мира.

13. Валидация ввода – проверка корректности вводимых данных с защитой от ошибок.

14. Выход. Выход из программы.

Навигация в меню осуществляется с помощью клавиатуры, вводом цифр от 0 до 11.

Описание разработки программы

Разработанная программа состоит из нескольких модулей, которые реализовывались в следующем порядке:

1. main.c – основной файл программы. Меню программы.
2. database.c / database.h – работа с базой данных (CRUD операции).
3. utils.c / utils.h – вспомогательные функции (ввод, цвет, очистка экрана).
4. file_manager.c / file_manager.h – сохранение и загрузка базы данных.
5. filtres.c / filtres.h – фильтрация записей.
6. sort.c / sort.h – сортировка записей.
7. search.c / search.h – поиск стран.
8. demo_data.c – предварительная база данных.

По порядку реализации модулей можно понять, что использовался нисходящий подход реализации программы. Этот метод основан на том, что

сначала проектируются основные компоненты программы, а затем уже дорабатываются ее мелкие детали. Это позволяет упростить ее тестирование, так как каждый модуль тестируется сразу же после его реализации, что помогает в создании более структурированных программ.

Отладка и тестирование

В качестве среды разработки была выбрана программа Microsoft Visual Studio 2022. Программа обладает всеми средствами, необходимыми при разработке и отладке программы.

Для отладки использовались несколько возможностей Visual Studio: точка останова, трассировка, анализ содержимого переменных.

Тестирование проводилось во время разработки и также после завершения разработки. В ходе нее было выявлено несколько проблем, связанных с работой с файлами, работой с памятью, работой с полями структуры. Также была исправлена ошибка в функции editCountry, которая приводила к некорректному редактированию записей.

Описание программы

Приложение countries.exe является основным модулем программы. При запуске программы происходит вывод заставки-приветствия, после чего пользователю необходимо выбрать пункт меню для дальнейшего использования базы данных. Выход из меню осуществляется после выбора пользователем нужного пункта. Описание состояний программы выполнено в таблицах 1.1-1.4.

Таблица 1.1 – Главное меню программы. В нём можно выбирать нужные действия для работы с базой данных

Клавиши, вызывающее событие	Действие пользователя	Действие программы
1,Enter	Выбран пункт «Создать новую БД»	Очищается текущая база данных
2,Enter	Выбран пункт «Добавить страну»	Запускается диалог добавления страны
3,Enter	Выбран пункт «Удалить страну»	Запускается диалог удаления страны
4,Enter	Выбран пункт «Редактировать страну»	Запускается диалог редактирования страны
5,Enter	Выбран пункт «Показать таблицу»	Открывается подменю просмотра
6,Enter	Выбран пункт «Фильтрация»	Открывается меню фильтрации
7,Enter	Выбран пункт «Сортировка»	Открывается меню сортировки
8,Enter	Выбран пункт «Поиск страны»	Запускается поиск по названию
9,Enter	Выбран пункт «Сохранить БД»	Сохранение базы в файл
10,Enter	Выбран пункт «Загрузить БД»	Загрузка базы из файла
11,Enter	Выбран пункт «Загрузить демо-данные»	Загружает файл с готовой базой данных всех стран мира
0,Enter	Выбран пункт «Выход»	Выполняется закрытие программы

Таблица 1.2 – Подменю просмотра базы данных. В нём нужно выбрать тип просмотра базы данных

Клавиши, вызывающее событие	Действие пользователя	Действие программы
1,Enter	Выбран пункт «Краткая таблица»	Вывод ID, страны, столицы, населения, площади, ИЧР
2,Enter	Выбран пункт «Расширенная таблица»	Вывод ВВП, ВВП на душу, валюты, часовых поясов
3,Enter	Выбран пункт «Полный просмотр»	Вывод всей информации о каждой стране
0,Enter	Выбран пункт «Назад»	Возврат в главное меню

Таблица 1.3 – Меню фильтрации. В нём нужно выбрать критерий фильтрации

Клавиши, вызывающее событие	Действие пользователя	Действие программы
1,Enter	По населению	Ввод порога, вывод стран с населением больше
2,Enter	По ВВП	Ввод порога, вывод стран с ВВП больше
3,Enter	По площади	Ввод порога, вывод стран с площадью больше
4,Enter	По валюте	Ввод названия валюты, вывод стран

Продолжение таблицы 1.3

5,Enter	По ИЧР	Ввод порога, вывод стран с ИЧР меньше или равно
0,Enter	Назад	Возврат в главное меню

Таблица 1.4 – Меню сортировки. В нём нужно выбрать критерий сортировки

Клавиши, вызывающее событие	Действие пользователя	Действие программы
1,Enter	По населению (возрастание)	Сортировка по полю population
2,Enter	По ВВП (убывание)	Сортировка по полю gdp
3,Enter	По площади (убывание)	Сортировка по полю area
4,Enter	По ИЧР (возрастание)	Сортировка по полю hdiRank
0,Enter	Назад	Возврат в главное меню

По завершению программы появляется титульный лист с фамилией и именем учащегося и номером группы.

Схемы

На рисунке 2.1 представлена схема взаимодействия программы. Main.c обращается через функции ко всем файлам системы. File_manager.c работает только с базой данных. В database.c используются функции сортировки, фильтрации и поиска, а также использует демо-данные, которые находятся в файлах demo_data.c, filtros.c, search.c, sort.c. В utils.c находятся вспомогательные функции и функции проверки корректного ввода данных через консоль, которые используются и в других функциях программы.

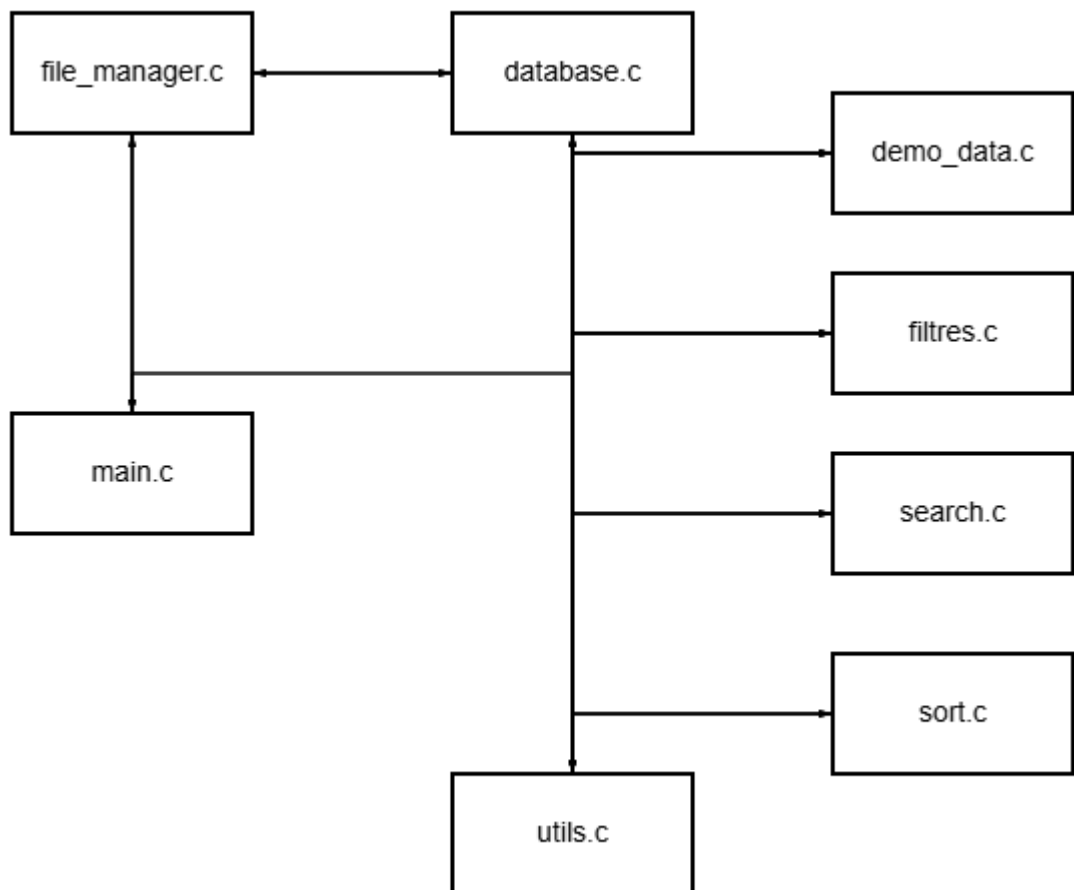


Рисунок 2.1 – Схема взаимодействия программы

На рисунке 2.2 изображена схема данных. Пользователь обращается к программе через операционную систему путём ввода данных с клавиатуры. Результат поступает пользователю на монитор.



Рисунок 2.2 – Схема данных

В файле main.c представлено главное меню программы (рисунок 2.3). В нём пользователь, путём выбора значения (0-11) переходит на функцию конкретной команды (поиска, сортировки и т.д.).

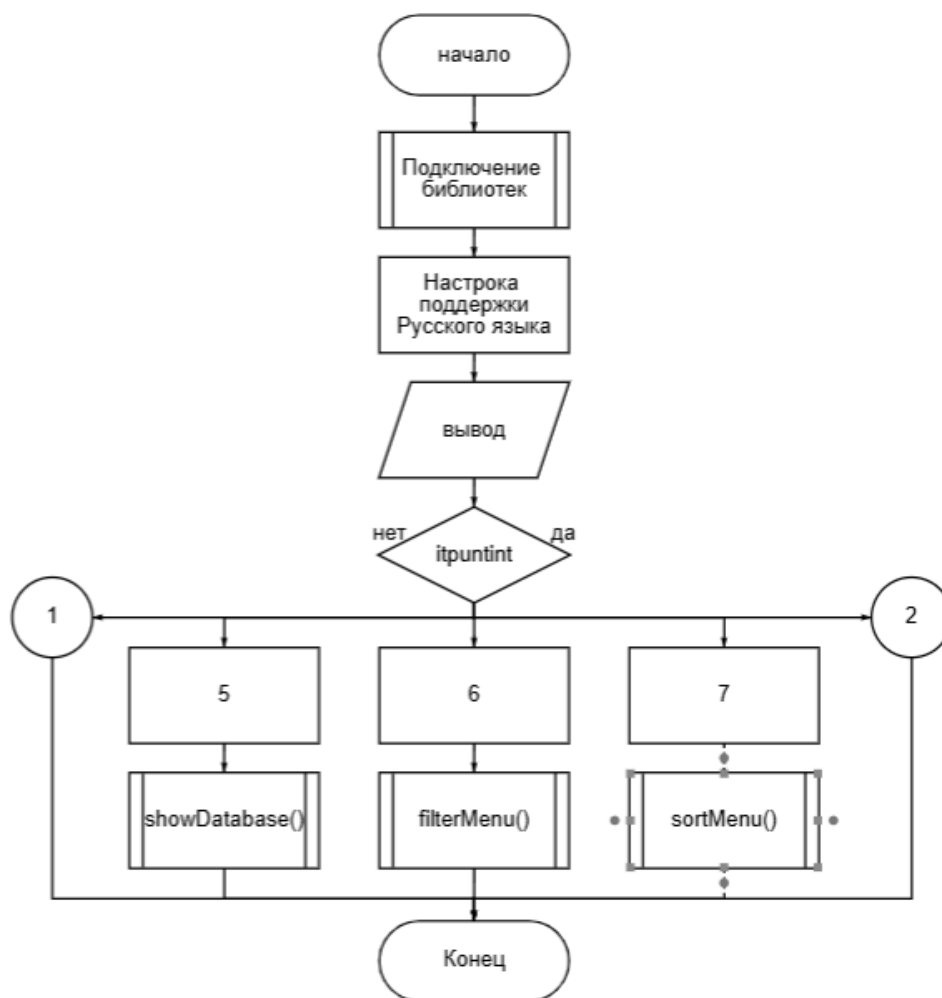


Рисунок 2.3 – Схема программы. Функция main. Начало и выбор с 5-7

На рисунке 2.4 представлено выбор меню с 1-4 и соответствующие к ним функции.

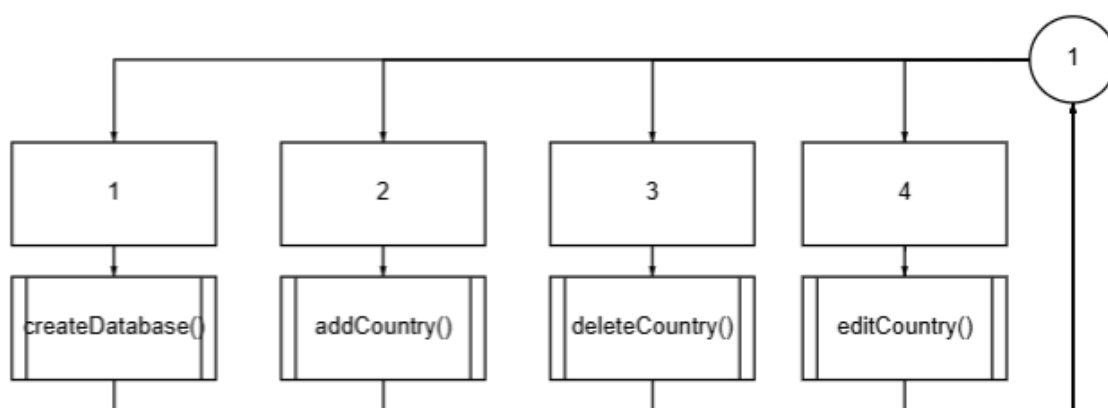


Рисунок 2.4 – Схема программы. Функция main. Выбор с 1-4

На рисунке 2.5 представлено выбор меню с 8-11 и соответствующие к ним функции.

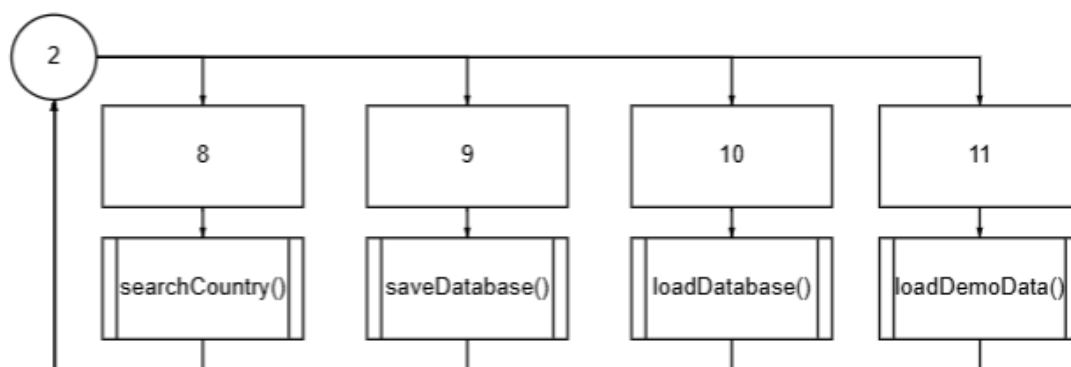


Рисунок 2.5 – Схема программы. Функция main. Выбор с 8-11

На рисунке 2.6 изображена схема программы функции database.c. Она включает в себя создание новой базы данных, добавление страны, удаление страны, редактирование страны и просмотр базы данных.

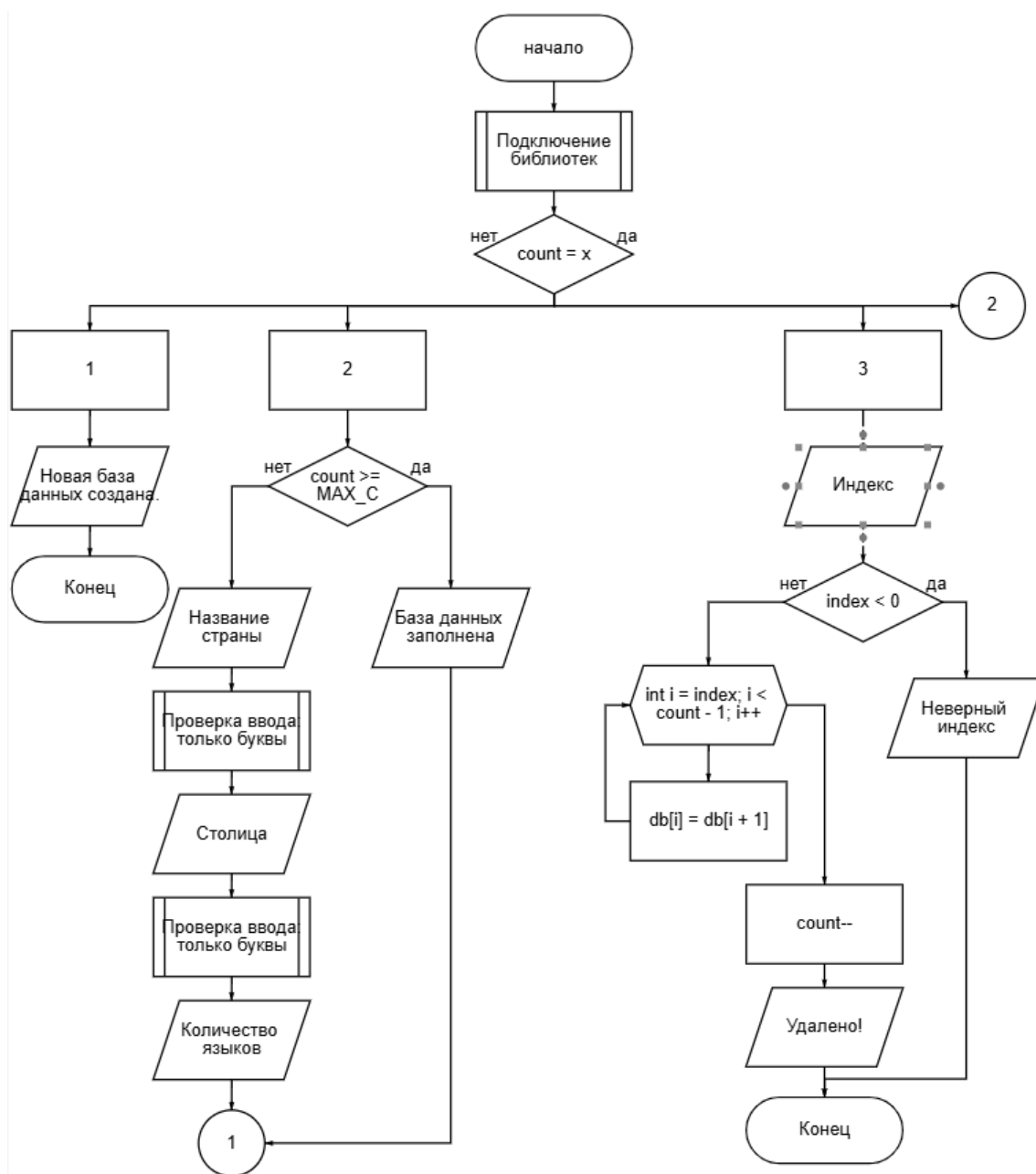


Рисунок 2.6 – Схема программы. Функция `database.c`

На рисунке 2.7 представлено продолжение функции по добавлению страны.

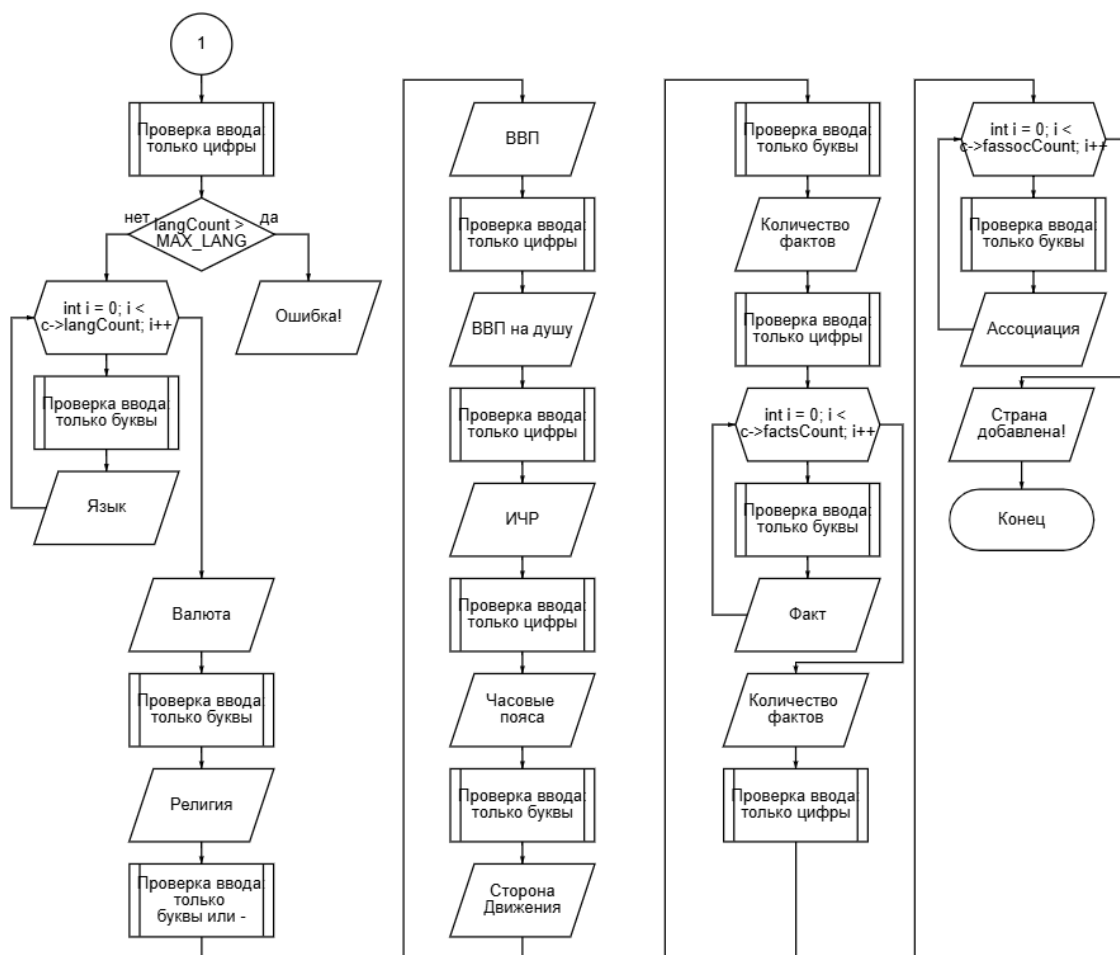


Рисунок 2.7 – Схема программы. Функция database.c. Добавление страны

На рисунке 2.8 представлено удаление и показ таблицы в различных формах.

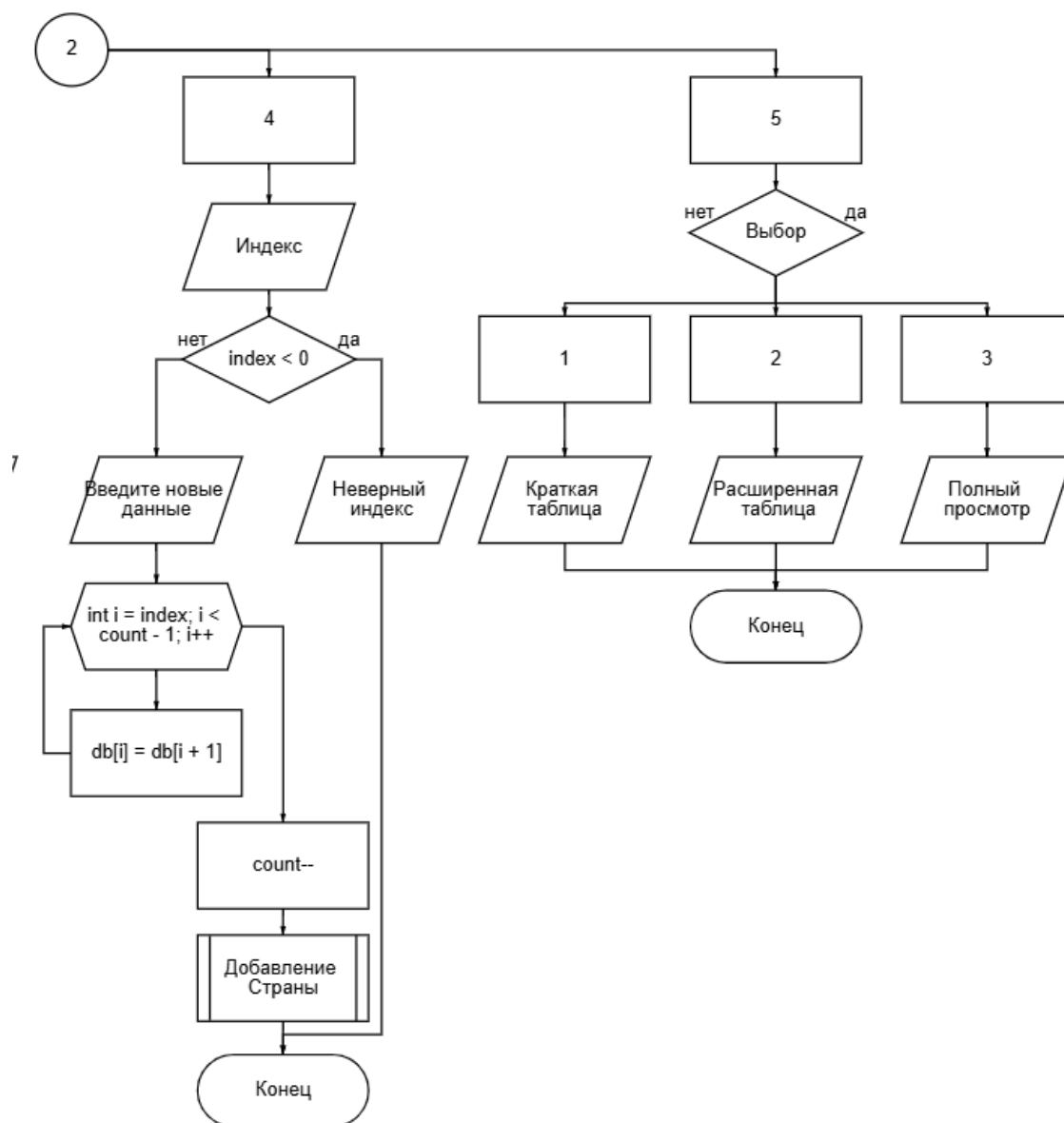


Рисунок 2.8 – Схема программы. Функция database.c. Выбор 4 и 5

На рисунке 2.9 изображена схема программы функции file_manager.c. В ней пользователь сохраняет и загружает базу данных. Так же после сохранения и перед загрузкой все данные шифруются с помощью ассемблерной вставки методом XOR-шифрованием.

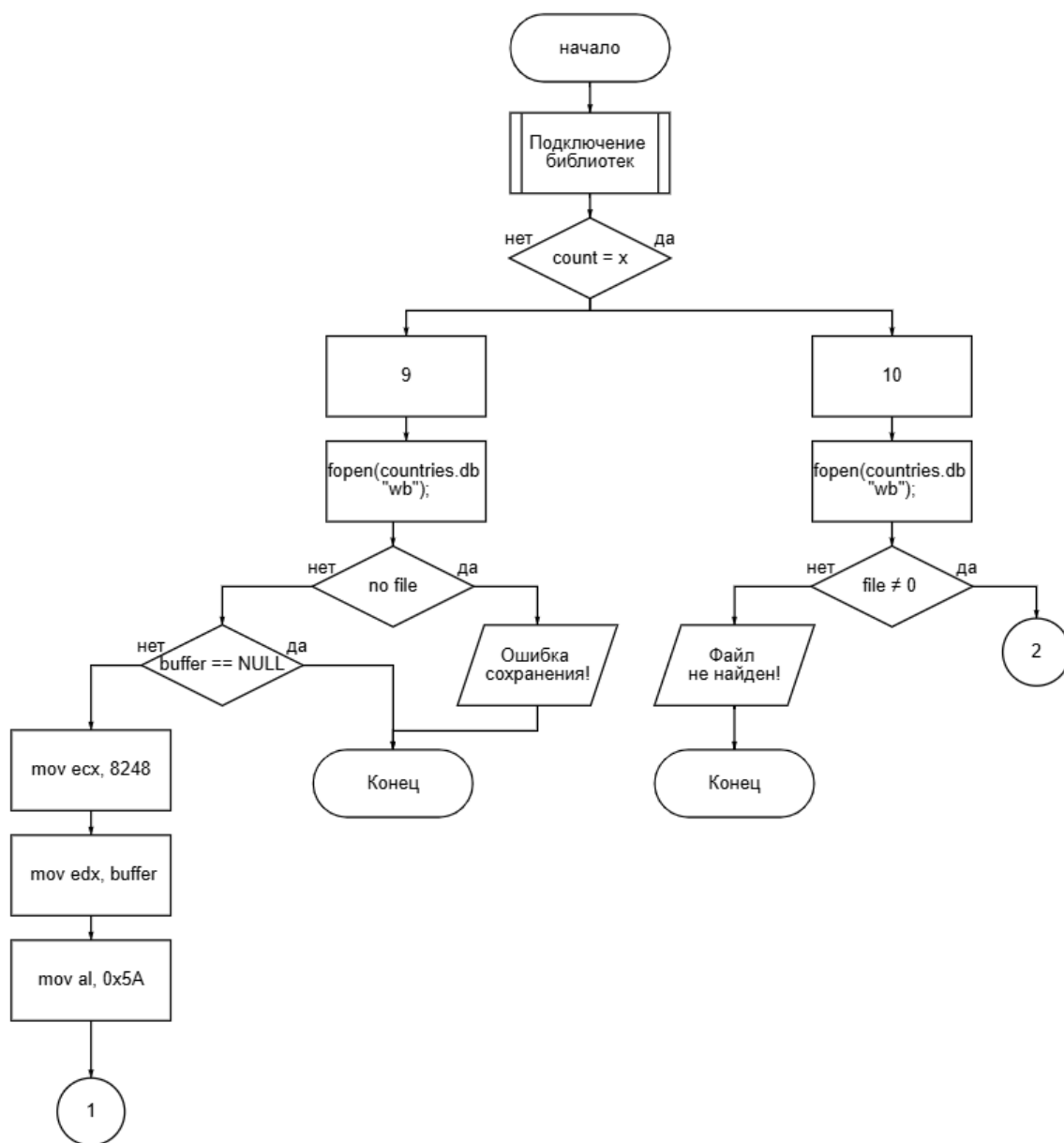


Рисунок 2.9 – Схема программы. Функция file_manager.c

На рисунке 2.10 представлено продолжение загрузки программы.



Рисунок 2.10 – Схема программы. Функция file_manager.c. Выбор 9

На рисунке 2.11 представлено продолжение сохранения программы.

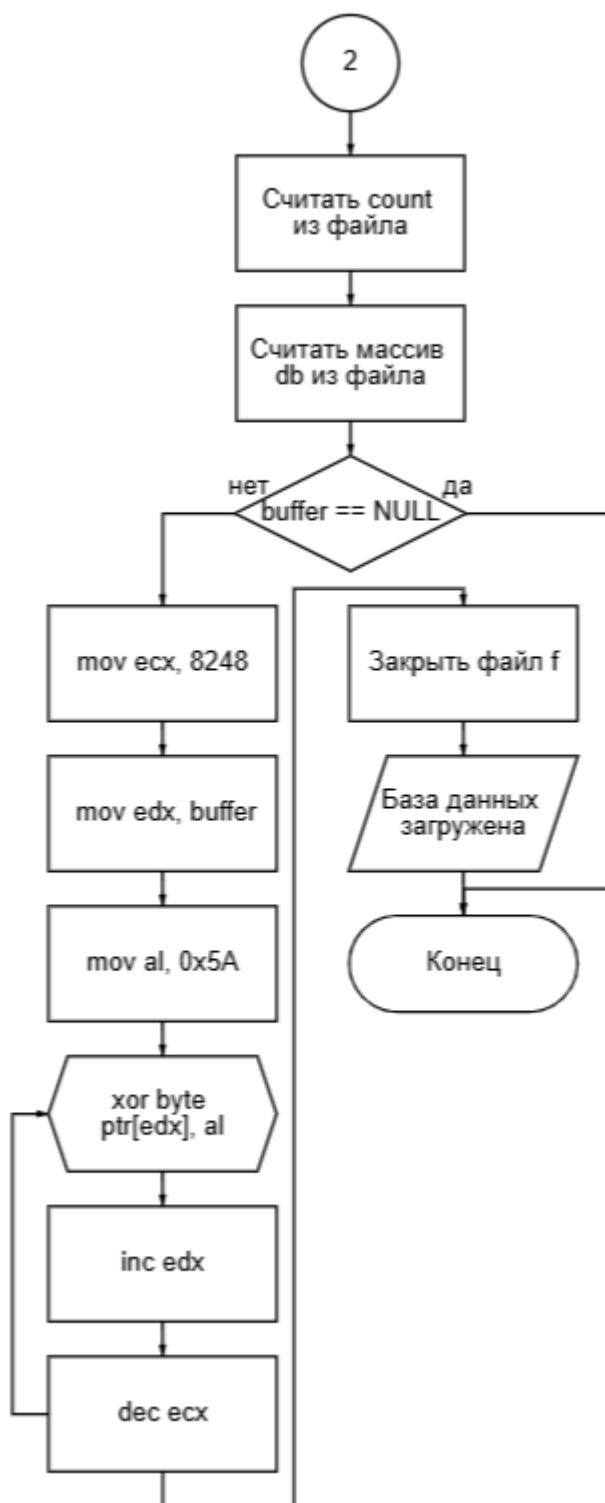


Рисунок 2.11 – Схема программы. Функция file_manager.c. Выбор 10

На рисунке 2.12 представлена иерархическая схема программы.

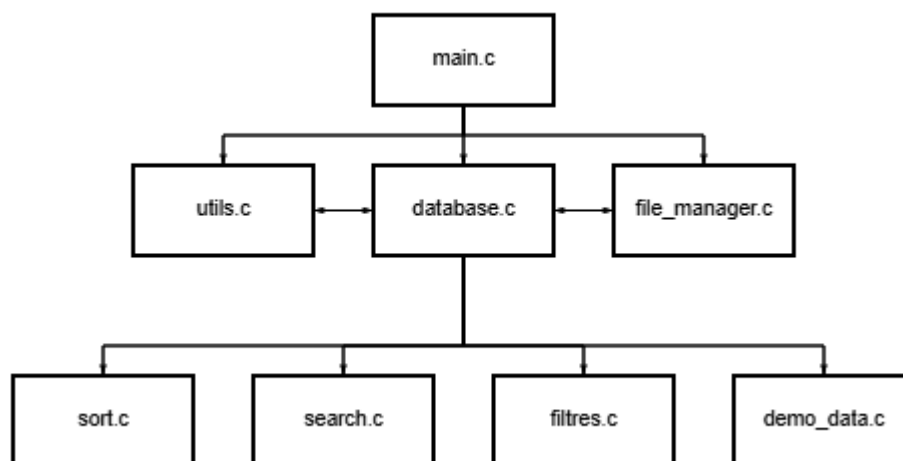


Рисунок 2.12 – Иерархическая схема программы

Данные схемы показывают основные алгоритмы работы программы.

Руководство пользователя

Программа countries.exe предназначена для хранения информации о странах мира. Программа имеет интуитивно понятный интерфейс и поддерживает такие операции как добавление новой записи в базу данных, удаление записи из базы данных, изменение записи в базе данных, поиск записи в базе данных, сохранение существующей базы данных в файл, импорт базы данных из файла, сортированный вывод записей базы данных и фильтрация записей в базе данных.

При запуске программы отображается главное меню (рисунок 3.1):

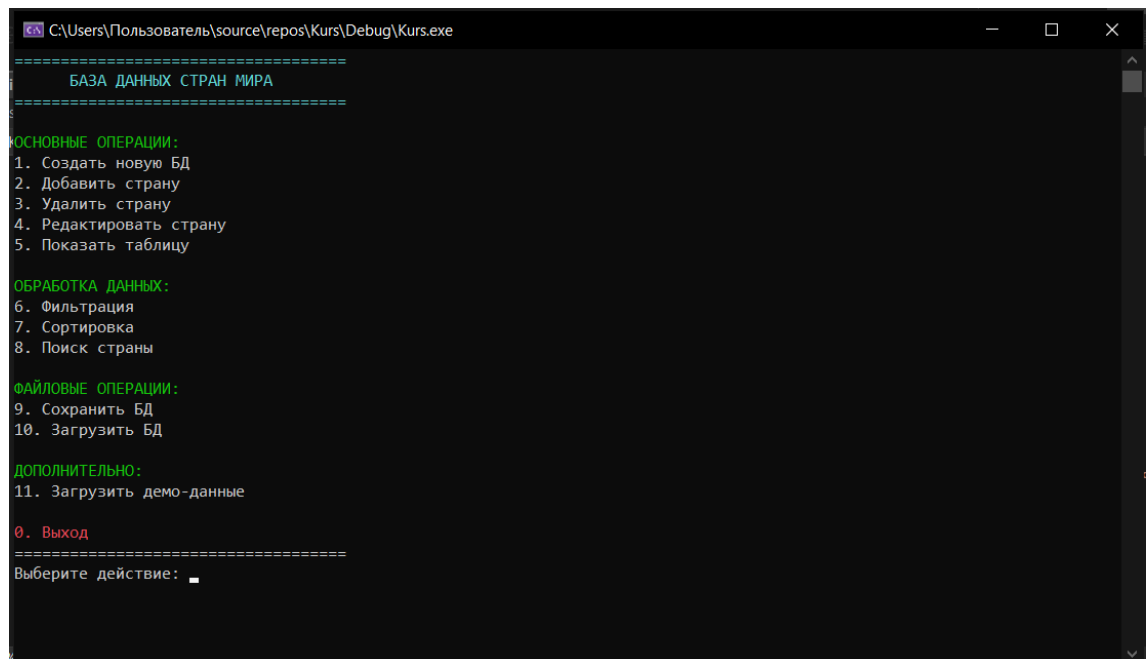


Рисунок 3.1 – Главное меню

Для добавления страны в базу данных необходимо выбрать пункт меню «Добавить страну». Далее необходимо заполнить форму (рисунок 3.2):

```

0. Выход
=====
Выберите действие: 2

=== ДОБАВЛЕНИЕ СТРАНЫ ===
Название страны: Австралия
Столица: Канберра
Население: 27012300
Площадь (км?): 7692024
Количество языков (0-5): 2
Язык 1: Австралийский
Язык 2: Английский
Валюта: австралийский доллар
Религия (или '-'): -
ВВП (долл. США): 178000000000
ВВП на душу: 66098
ИЧР (1=лучший): 5
Часовые пояса (+3, +8...+11 и т.д.): +8...+11
Сторона движения (лево/право): лево
Количество фактов (0-10): 3
Факт 1: Австралийское шоссе №1 является самой длинной национальной сетью автодорог в мире. Ее общая длина составляет примерно 14,5 тысячи километров.
Факт 2: Почти половина жителей Австралии (12 млн) живет в трех городах - Мельбурне, Сиднее и Брисбене.
Факт 3: Австралии самая толерантная страна в мире
Количество ассоциаций (0-10): 3
Ассоциация 1: Редкие животные
Ассоциация 2: Пустыни
Ассоциация 3: Сиднейский оперный театр_

```

Рисунок 3.2 – Форма добавления страны

Для удаления записи из каталога необходимо выбрать пункт меню «Удалить запись». Далее необходимо ввести ID удаляемой записи и нажать клавишу Enter (рисунок 3.3):

```

C:\Users\Пользователь\source\repos\Kurs\Debug\Kurs.exe
=====
      БАЗА ДАННЫХ СТРАН МИРА
=====

ОСНОВНЫЕ ОПЕРАЦИИ:
1. Создать новую БД
2. Добавить страну
3. Удалить страну
4. Редактировать страну
5. Показать таблицу

ОБРАБОТКА ДАННЫХ:
6. Фильтрация
7. Сортировка
8. Поиск страны

ФАЙЛОВЫЕ ОПЕРАЦИИ:
9. Сохранить БД
10. Загрузить БД

ДОПОЛНИТЕЛЬНО:
11. Загрузить демо-данные

0. Выход
=====
Выберите действие: 3
Индекс: 0

```

Рисунок 3.3 – Удаление страны

Для просмотра базы данных выберите пункт «Показать таблицу».
Откроется подменю (рисунок 3.4):

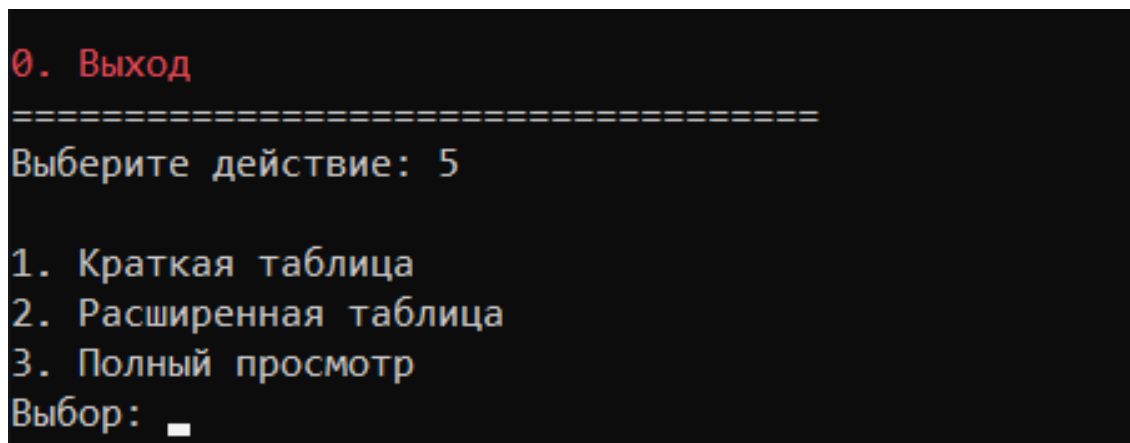


Рисунок 3.4 – Подменю выбора показа таблицы

В нём нужно выбрать режим показа данных (рисунок 3.5):

C:\Users\Пользователь\source\repos\Kurs\Debug\Kurs.exe

1. Краткая таблица
2. Расширенная таблица
3. Полный просмотр
Выбор: 1

ID	Страна	Столица	Население	Площадь	ИЧР
0	Австралия	Канберра	27012300	7692024	5
1	Австрия	Вена	9027999	83879	25
2	Азербайджан	Баку	10187539	86600	87
3	Албания	Тирана	2793592	28748	69
4	Алжир	Алжир	45400000	2381740	82
5	Ангола	Луанда	37290193	1246700	149
6	Андорра	Андорра-ла-Велья	81588	468	36
7	Антигуа и Барбуда	Сент-Джонс	100772	440	74
8	Аргентина	Буэнос-Айрес	7327407	2780400	48
9	Армения	Ереван	3165000	29800	85
10	Афганистан	Кабул	39232003	652864	169
11	Багамские Острова	Нассау	412628	13878	60
12	Бангладеш	Дакка	169828911	148460	135
13	Барбадос	Бриджтаун	290604	430	56
14	Бахрейн	Манама	1705003	766	45
15	Белиз	Бельмопан	430131	22966	103
16	Беларусь	Минск	9155978	207628	60
17	Бельгия	Брюссель	11584008	30528	17
18	Бенин	Порто-Ново	12864634	112622	163
19	Болгария	София	6847402	110994	68
20	Боливия	Сукре(юр), Ла-Пас(факт)	12186079	1098581	114

Рисунок 3.5 – Результат выбора показа таблицы

Для фильтрации выберите пункт «Фильтрация». Появится подменю (рисунок 3.6):



Рисунок 3.6 – Меню фильтрации

В нём нужно выбрать критерий фильтрации (рисунок 3.7):

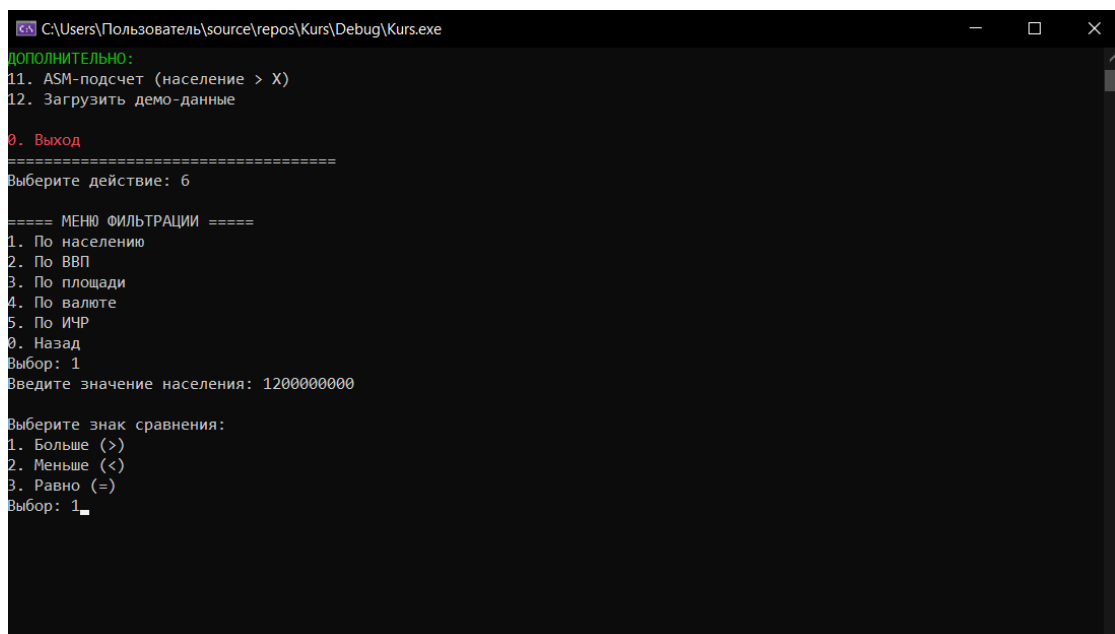


Рисунок 3.7 – Критерии фильтрации

Затем нужно выбрать знак сравнения (рисунок 3.8):

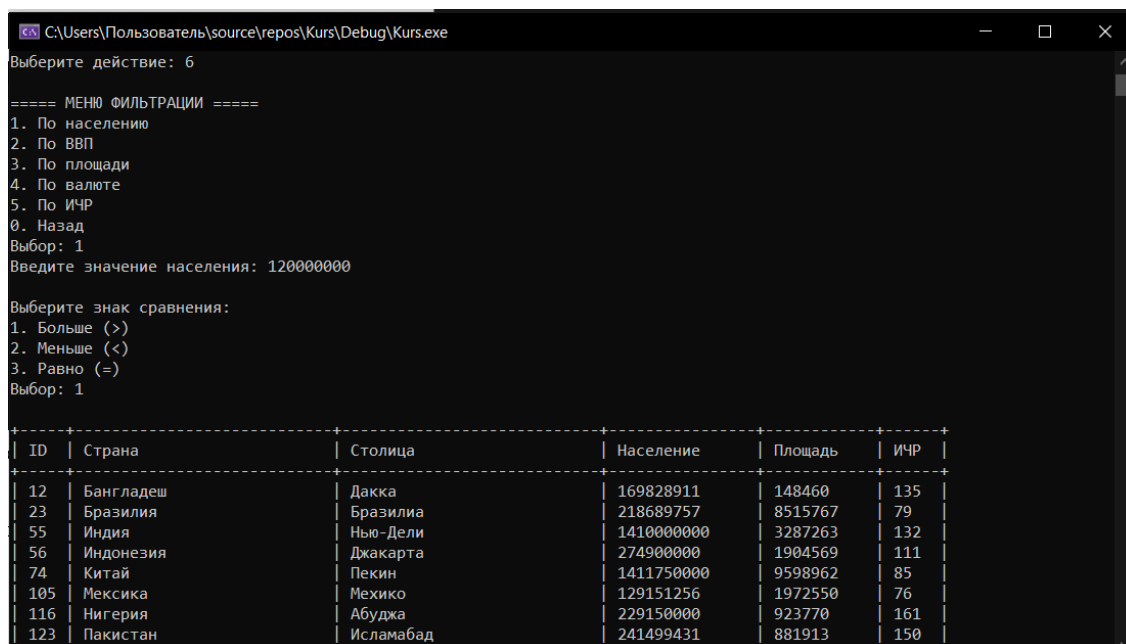


Рисунок 3.8 – Выбор знака сравнения

Для сортировки выберите пункт «Сортировка» (рисунок 3.9):

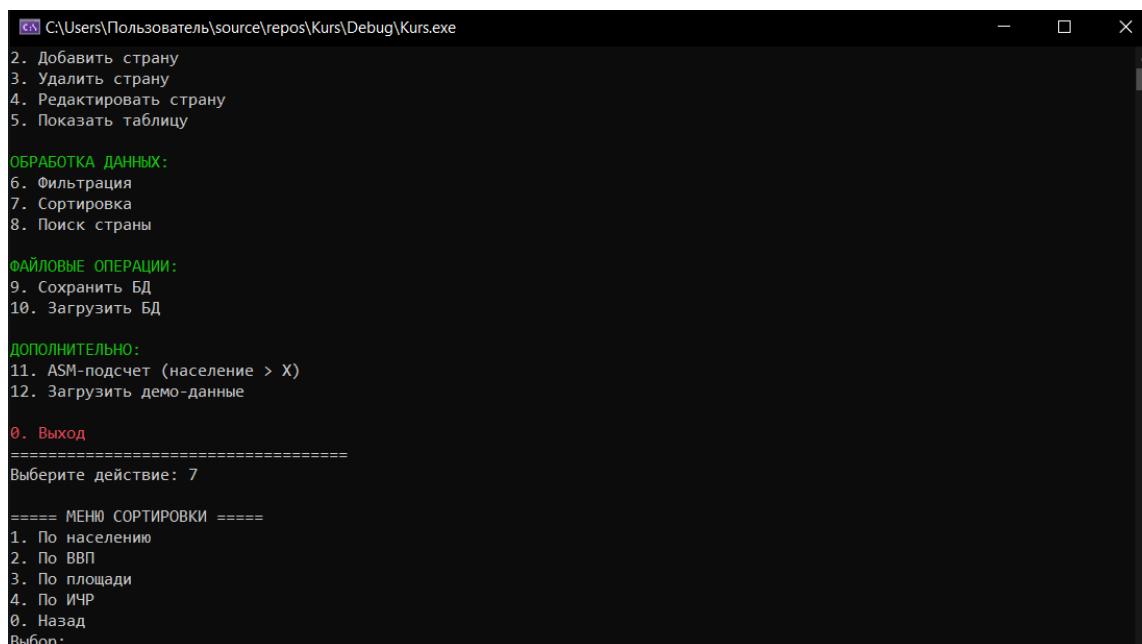


Рисунок 3.9 – Меню сортировки

В нём нужно выбрать критерий сортировки (рисунок 3.10):

```
C:\Users\Пользователь\source\repos\Kurs\Debug\Kurs.exe
ОБРАБОТКА ДАННЫХ:
6. Фильтрация
7. Сортировка
8. Поиск страны

ФАЙЛОВЫЕ ОПЕРАЦИИ:
9. Сохранить БД
10. Загрузить БД

ДОПОЛНИТЕЛЬНО:
11. АСМ-подсчет (население > X)
12. Загрузить демо-данные

0. Выход
=====
Выберите действие: 7

===== МЕНЮ СОРТИРОВКИ =====
1. По населению
2. По ВВП
3. По площади
4. По ИЧР
0. Назад
Выбор: 1

Выберите направление сортировки:
1. По возрастанию (от меньшего к большему)
2. По убыванию (от большего к меньшему)
Выбор:
```

Рисунок 3.11 – Выбор критерия сортировки

Затем нужно выбрать направление сортировки (рисунок 3.11):

```
C:\Users\Пользователь\source\repos\Kurs\Debug\Kurs.exe
12. Загрузить демо-данные

0. Выход
=====
Выберите действие: 7

===== МЕНЮ СОРТИРОВКИ =====
1. По населению
2. По ВВП
3. По площади
4. По ИЧР
0. Назад
Выбор: 1

Выберите направление сортировки:
1. По возрастанию (от меньшего к большему)
2. По убыванию (от большего к меньшему)
Выбор: 2

Сортировка по населению выполнена (по убыванию)!

Результат сортировки:
+-----+-----+-----+-----+-----+
| ID | Страна | Столица | Население | Площадь | ИЧР |
+-----+-----+-----+-----+-----+
| 0 | Китай | Пекин | 1411750000 | 9598962 | 85 |
| 1 | Индия | Нью-Дели | 1410000000 | 3287263 | 132 |
| 2 | США | Вашингтон | 333287557 | 9833517 | 15 |
| 3 | Индонезия | Джакарта | 274900000 | 1904569 | 111 |
+-----+-----+-----+-----+-----+
```

Рисунок 3.11 – Выбор направления сортировки

Для поиска выберите пункт «Поиск страны» (рисунок 3.12):

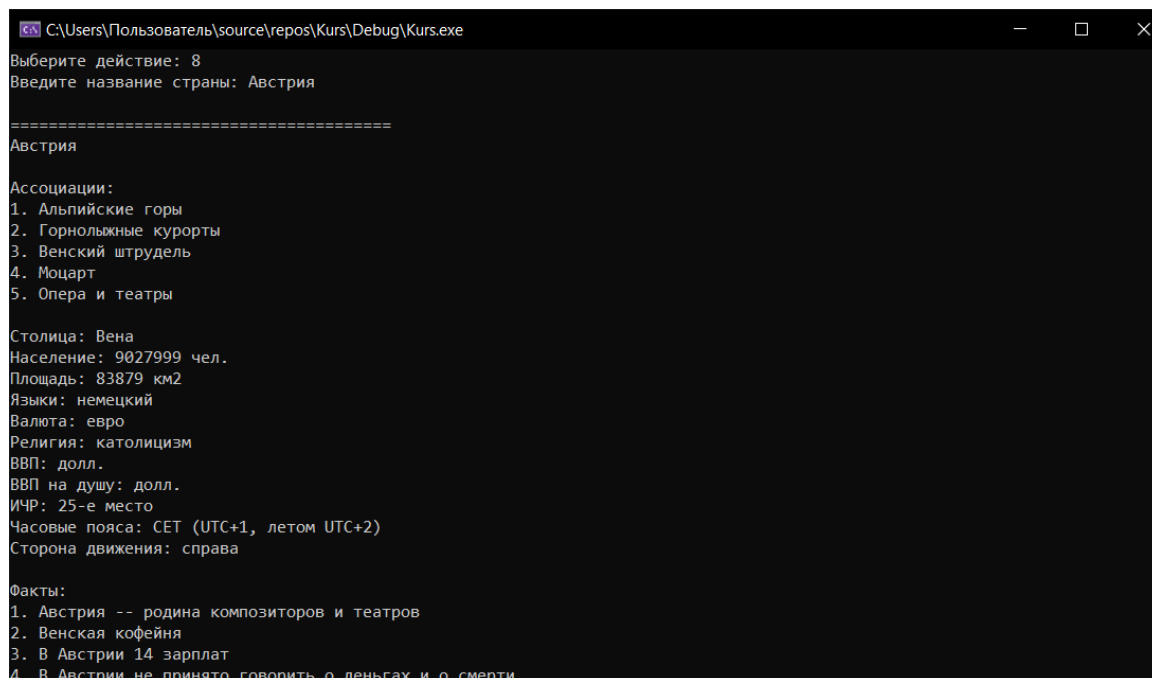


Рисунок 3.12 – Ввод названия страны для сортировки

Сохранение: выберите пункт «Сохранить БД». Все данные будут сохранены в файл countries.db (рисунок 3.13):

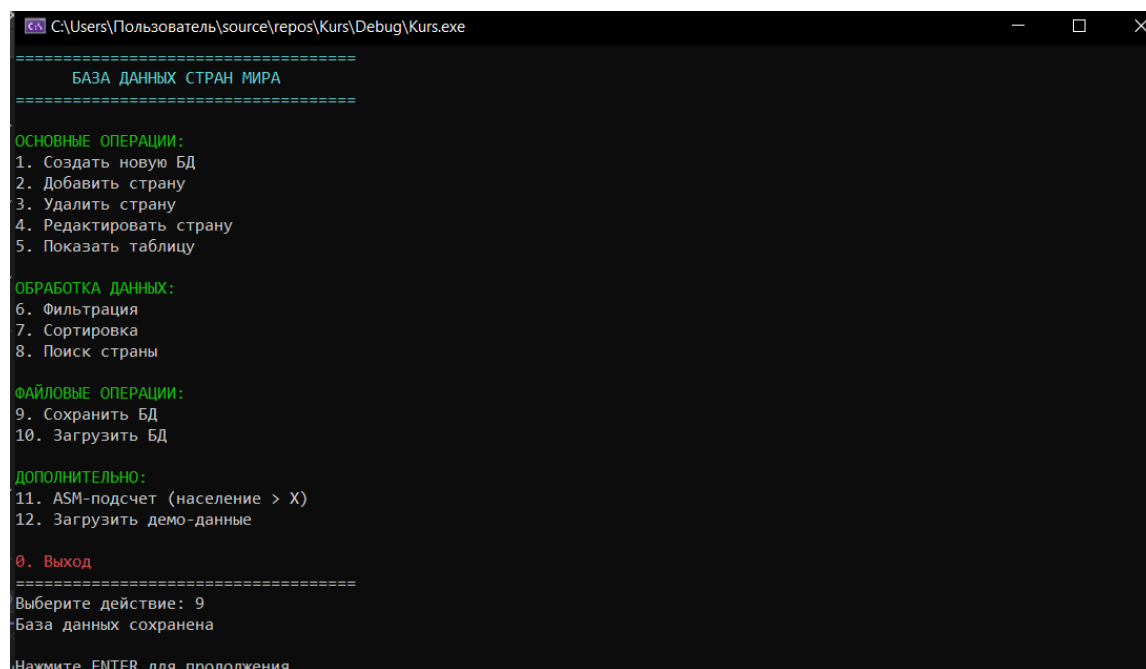


Рисунок 3.13 – Сохранение базы данных

Загрузка: выберите пункт «Загрузить БД». Данные будут загружены из файла countries.db (рисунок 3.14):

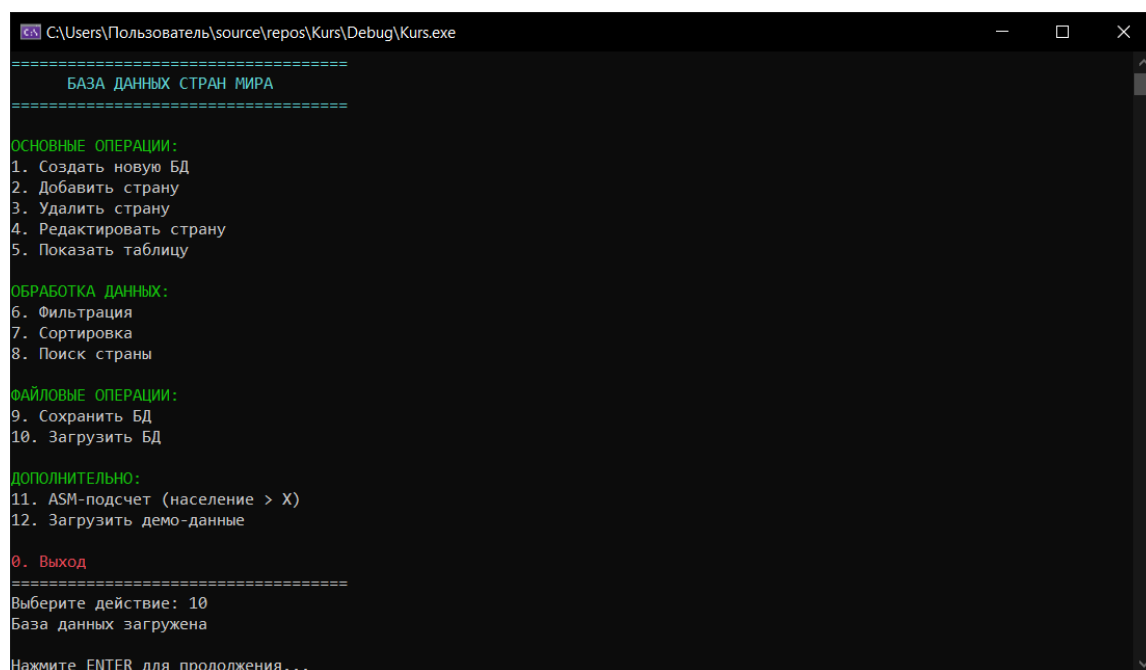


Рисунок 3.14 – Загрузка базы данных

Для выхода выберите пункт «Выход» или нажмите 0.

Заключение

При выполнении данной курсовой работы были получены навыки программирования на языке С, работы со структурами и массивами, работы с файлами (чтение/запись), использования ассемблерных вставок, разработки многофункционального меню, фильтрации и сортировки данных, отладки и тестирования программного обеспечения.

В рамках курсовой работы была написана программа для работы с базой данных стран. Она имеет все необходимые функции для работы с базами данных.

В дальнейшем программу можно улучшить путём добавления графического интерфейса, возможности использования мыши, экспорта данных в форматы CSV/JSON, добавления возможности редактирования отдельных полей, расширения количества стран в базе данных.

Список используемых источников

1. Керниган, Б. У. Язык программирования С: [перевод с английского] / Б. У. Керниган, Д. М. Ритчи. – 2-е изд. – Москва: Вильямс, 2015. – 304 с. – ISBN 978-5-8459-0891-9.
2. Ирвин, К. Р. Язык ассемблера для процессоров Intel: [перевод с английского] / Кип Р. Ирвин. – 4-е изд. – Москва: Вильямс, 2005. – 912 с. – ISBN 5-8459-0779-9.
3. Иванова, Г. С. Программирование на ассемблере ПЭВМ: учебное пособие / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев; под редакцией Г. С. Ивановой. – Москва: Издательство МГТУ им. Н. Э. Баумана, 2000. – 154 с. – ISBN 5-7038-1594-5. [1, 2, 3, 4]

Приложение А. Листинг Программы

Файл database.c

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include "database.h"
#include "utils.h"

Country db[MAX_COUNTRIES];
int count = 0;

// Проверка, что строка состоит только из букв и пробелов
int isValidText(const char* str)
{
    if (strlen(str) == 0) return 0;
    for (int i = 0; str[i]; i++)
        if (!isalpha(str[i]) && str[i] != ' ' && str[i] != '-')
            return 0;
    return 1;
}

// Проверка, что в строке только цифры
int isOnlyDigits(const char* str)
{
    for (int i = 0; str[i]; i++)
        if (!isdigit(str[i]))
            return 0;
    return 1;
}

// Ввод числа с проверкой (только цифры)
int inputPositiveInt(const char* msg)
{
    char buf[100];
    int x;
    while (1)
    {
        printf("%s", msg);
        fgets(buf, sizeof(buf), stdin);
        buf[strcspn(buf, "\n")] = 0;

        // Проверяем, что ввели только цифры
        int onlyDigits = 1;
        for (int i = 0; buf[i]; i++)
            if (!isdigit(buf[i]))
                onlyDigits = 0;

        if (onlyDigits && sscanf(buf, "%d", &x) == 1 && x > 0)
            return x;
    }
}
```

```

        printf("Ошибка! Введите положительное число (только
цифры).\n");
    }
}

// Ввод числа с плавающей точкой
double inputPositiveDouble(const char* msg)
{
    char buf[100];
    double x;
    while (1)
    {
        printf("%s", msg);
        fgets(buf, sizeof(buf), stdin);
        buf[strcspn(buf, "\n")] = 0;

        if (sscanf(buf, "%lf", &x) == 1 && x > 0)
            return x;

        printf("Ошибка! Введите положительное число.\n");
    }
}

void createDatabase()
{
    count = 0;
    printf("Новая база данных создана.\n");
}

void addCountry()
{
    if (count >= MAX_COUNTRIES)
    {
        printf("База данных заполнена.\n");
        return;
    }

    Country* c = &db[count];

    printf("\n=== ДОБАВЛЕНИЕ СТРАНЫ ===\n");

    inputValidName("Название страны: ", c->name, MAX_STR);
    inputValidName("Столица: ", c->capital, MAX_STR);

    c->population = inputPositiveInt("Население: ");
    c->area = (float)inputPositiveDouble("Площадь (км²): ");

    c->langCount = inputPositiveInt("Количество языков (0-5):
");
    if (c->langCount > MAX_LANG) c->langCount = MAX_LANG;

    for (int i = 0; i < c->langCount; i++)
    {

```

```

        char temp[50];
        sprintf(temp, "Язык %d: ", i + 1);
        inputValidName(temp, c->languages[i], MAX_STR);
    }

    inputValidName("Валюта: ", c->currency, MAX_STR);
    inputString("Религия (или '-'): ", c->religion, MAX_STR);
    if (strlen(c->religion) == 0) strcpy(c->religion, "-");

    c->gdp = inputPositiveDouble("ВВП (долл. США): ");
    c->gdpPerCapita = inputPositiveDouble("ВВП на душу: ");
    c->hdiRank = inputPositiveInt("ИЧР (1=лучший): ");

    inputString("Часовые пояса (+3, +8...+11 и т.д.): ", c->timezones, MAX_STR);
    inputValidName("Сторона движения (лево/право): ", c->drivingSide, MAX_STR);

    c->factsCount = inputPositiveInt("Количество фактов (0-10): ");
    if (c->factsCount > MAX_FACTS) c->factsCount = MAX_FACTS;

    for (int i = 0; i < c->factsCount; i++)
    {
        char temp[50];
        sprintf(temp, "Факт %d: ", i + 1);
        inputString(temp, c->facts[i], 200);
    }

    c->assocCount = inputPositiveInt("Количество ассоциаций (0-10): ");
    if (c->assocCount > MAX_ASSOC) c->assocCount = MAX_ASSOC;

    for (int i = 0; i < c->assocCount; i++)
    {
        char temp[50];
        sprintf(temp, "Ассоциация %d: ", i + 1);
        inputString(temp, c->associations[i], MAX_STR);
    }

    count++;
    printf("✓ Страна добавлена!\n");
}

void deleteCountry()
{
    int index = inputPositiveInt("Индекс: ");

    if (index < 0 || index >= count)
    {
        printf("Неверный индекс\n");
        return;
    }
}

```

```

    }

    printf("Удаление: %s\n", db[index].name);

    for (int i = index; i < count - 1; i++)
        db[i] = db[i + 1];

    count--;
    printf("✓ Удалено\n");
}

void editCountry()
{
    int index = inputPositiveInt("Индекс: ");

    if (index < 0 || index >= count)
    {
        printf("Неверный индекс\n");
        return;
    }

    printf("Редактирование: %s\n", db[index].name);
    printf("Введите новые данные:\n");

    Country temp = db[index];

    for (int i = index; i < count - 1; i++)
        db[i] = db[i + 1];

    count--;

    addCountry();
}

void showDatabase()
{
    if (count == 0)
    {
        printf("База данных пуста.\n");
        return;
    }

    printf("\n1. Краткая таблица\n");
    printf("2. Расширенная таблица\n");
    printf("3. Полный просмотр\n");

    int mode = inputPositiveInt("Выбор: ");

    if (mode == 1)
    {
        printTableHeader();
        for (int i = 0; i < count; i++)

```



```

#define MAX_FACTS 20
#define MAX_LANG 5
#define MAX_ASSOC 10

typedef struct
{
    char name[MAX_STR];
    char capital[MAX_STR];
    int population;
    float area;
    char languages[MAX_LANG][MAX_STR];
    int langCount;
    char currency[MAX_STR];
    char religion[MAX_STR];
    double gdp;
    double gdpPerCapita;
    int hdiRank;
    char timezones[MAX_STR];
    char drivingSide[MAX_STR];
    char facts[MAX_FACTS][200];
    int factsCount;
    char associations[MAX_ASSOC][MAX_STR];
    int assocCount;
} Country;

extern Country db[MAX_COUNTRIES];
extern int count;

void createDatabase();
void addCountry();
void deleteCountry();
void editCountry();
void showDatabase();
void printCountry(Country c);
void printTableHeader();
void printTableRow(int i);
void printExtendedTable();
void printTableLow(int i);
void printTableFooter();
void loadDemoData();

void formatPopulation(int population, char* output);

#endif

```

Файл demo_data.c

```

#include "database.h"
#include "utils.h"

void loadDemoData()

```

```

{
    count = 0;

    // ===== 1. АВСТРАЛИЯ =====
    strcpy(db[count].name, "Австралия");
    strcpy(db[count].capital, "Канберра");
    db[count].population = 27012300;
    db[count].area = 7692024;
    db[count].langCount = 2;
    strcpy(db[count].languages[0], "английский");
    strcpy(db[count].languages[1], "австралийский");
    strcpy(db[count].currency, "австралийский доллар");
    strcpy(db[count].religion, "-");
    db[count].gdp = 1780000000000.0;
    db[count].gdpPerCapita = 66098.0;
    db[count].hdiRank = 5;
    strcpy(db[count].timezones, "+8...+11");
    strcpy(db[count].drivingSide, "слева");
    db[count].factsCount = 11;
    strcpy(db[count].facts[0], "Австралийское шоссе №1
является самой длинной национальной сетью автодорог в мире");
    strcpy(db[count].facts[1], "Почти половина жителей
Австралии (12 млн) живет в трех городах - Мельбурне, Сиднее и
Брисбене");
    strcpy(db[count].facts[2], "63% австралийцев страдают от
избыточного веса");
    strcpy(db[count].facts[3], "Первые жители Австралии - это
политические заключённые из Европы");
    strcpy(db[count].facts[4], "В Австралии очень много
полезных ресурсов, которые они отправляют в Китай");
    strcpy(db[count].facts[5], "В Австралии много студентов,
так как обучение стоит дешевле");
    strcpy(db[count].facts[6], "В Австралии очень много
иммигрантов");

```

```

        strcpy(db[count].facts[7], "Австралия самая толерантная
страна в мире");
        strcpy(db[count].facts[8], "В Австралии много людей
помешаны на спорте");
        strcpy(db[count].facts[9], "Здесь самые дорогие сигареты
в мире");
        strcpy(db[count].facts[10], "ВВП - услуги 72%");
        db[count].assocCount = 5;
        strcpy(db[count].associations[0], "Кенгуру");
        strcpy(db[count].associations[1], "Сиднейский оперный
театр");
        strcpy(db[count].associations[2], "Опасные животные");
        strcpy(db[count].associations[3], "Коала");
        strcpy(db[count].associations[4], "Пустыни");
        count++;

```

Файл file_manager.c

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include "database.h"

void xorBufferAsm(void* buffer)
{
    if (buffer == NULL) return;

    __asm
    {
        mov ecx, 8248
        mov edx, buffer
        mov al, 0x5A

        XOR_LOOP :
        xor byte ptr[edx], al
        inc edx
        dec ecx
        jnz XOR_LOOP
    }
}

void saveDatabase()
{
    FILE* f = fopen("countries.db", "wb");

    if (!f)
    {

```

```

        printf("Ошибка сохранения!\n");
        return;
    }

    xorBufferAsm(db);

    fwrite(&count, sizeof(int), 1, f);
    fwrite(db, sizeof(Country), count, f);

    fclose(f);

    printf("База данных сохранена\n");
}

void loadDatabase()
{
    FILE* f = fopen("countries.db", "rb");

    if (!f)
    {
        printf("Файл не найден\n");
        return;
    }

    fread(&count, sizeof(int), 1, f);

    fread(db, sizeof(Country), count, f);

    xorBufferAsm(db);

    fclose(f);

    printf("База данных загружена\n");
}

```

Файл file_manager.h

```

#ifndef FILE_MANAGER_H
#define FILE_MANAGER_H

void saveDatabase();
void loadDatabase();

#endif

```

Файл filtros.c

```

#include <string.h>
#include <stdio.h>
#include "database.h"
#include "utils.h"

void printFilterResult(int indices[], int foundCount)
{

```

```

    if (foundCount == 0)
    {
        printf("\nСтраны не найдены.\n");
        return;
    }

    printf("\n");
    printTableHeader();
    for (int i = 0; i < foundCount; i++)
    {
        printTableRow(indices[i]);
    }
    printTableFooter();
    printf("\nНайдено стран: %d\n", foundCount);
}

void filterMenu()
{
    while (1)
    {
        printf("\n===== МЕНЮ ФИЛЬТРАЦИИ =====\n");
        printf("1. По населению\n");
        printf("2. По ВВП\n");
        printf("3. По площади\n");
        printf("4. По валюте\n");
        printf("5. По ИЧР\n");
        printf("0. Назад\n");

        int c = inputInt("Выбор: ");

        if (c == 0) return;

        // ===== 1. ФИЛЬТР ПО НАСЕЛЕНИЮ =====
        if (c == 1)
        {
            int value = inputInt("Введите значение населения:
");

            printf("\nВыберите знак сравнения:\n");
            printf("1. Больше (>)\n");
            printf("2. Меньше (<)\n");
            printf("3. Равно (=)\n");
            int sign = inputInt("Выбор: ");

            int foundIndices[MAX_COUNTRIES];
            int foundCount = 0;

            for (int i = 0; i < count; i++)
            {
                if (sign == 1 && db[i].population > value)
                    foundIndices[foundCount++] = i;
                else if (sign == 2 && db[i].population < value)
                    foundIndices[foundCount++] = i;
                else if (sign == 3 && db[i].population == value)

```

```

        foundIndices[foundCount++] = i;
    }

    printFilterResult(foundIndices, foundCount);
}

// 2. ФИЛЬТР ПО ВВП
if (c == 2)
{
    double value = inputDouble("Введите значение ВВП:
");

    printf("\nВыберите знак сравнения:\n");
    printf("1. Больше (>)\n");
    printf("2. Меньше (<)\n");
    printf("3. Равно (=\n");
    int sign = inputInt("Выбор: ");

    int foundIndices[MAX_COUNTRIES];
    int foundCount = 0;

    for (int i = 0; i < count; i++)
    {
        if (sign == 1 && db[i].gdp > value)
            foundIndices[foundCount++] = i;
        else if (sign == 2 && db[i].gdp < value)
            foundIndices[foundCount++] = i;
        else if (sign == 3 && db[i].gdp == value)
            foundIndices[foundCount++] = i;
    }

    printFilterResult(foundIndices, foundCount);
}

// 3. ФИЛЬТР ПО ПЛОЩАДИ
if (c == 3)
{
    double value = inputDouble("Введите значение
площади: ");

    printf("\nВыберите знак сравнения:\n");
    printf("1. Больше (>)\n");
    printf("2. Меньше (<)\n");
    printf("3. Равно (=\n");
    int sign = inputInt("Выбор: ");

    int foundIndices[MAX_COUNTRIES];
    int foundCount = 0;

    for (int i = 0; i < count; i++)
    {
        if (sign == 1 && db[i].area > value)
            foundIndices[foundCount++] = i;
        else if (sign == 2 && db[i].area < value)
            foundIndices[foundCount++] = i;
    }
}

```

```

        else if (sign == 3 && db[i].area == value)
            foundIndices[foundCount++] = i;
    }

    printFilterResult(foundIndices, foundCount);
}

// 4. ФИЛЬТР ПО ВАЛЮТЕ (точное совпадение)
if (c == 4)
{
    char cur[MAX_STR];
    inputString("Введите название валюты: ", cur,
MAX_STR);

    int foundIndices[MAX_COUNTRIES];
    int foundCount = 0;

    for (int i = 0; i < count; i++)
    {
        if (strcmp(db[i].currency, cur) == 0)
            foundIndices[foundCount++] = i;
    }

    printFilterResult(foundIndices, foundCount);
}

// 5. ФИЛЬТР ПО ИЧР
if (c == 5)
{
    int value = inputInt("Введите значение ИЧР
(1=лучший): ");
    printf("\nВыберите знак сравнения:\n");
    printf("1. Больше (>)\n");
    printf("2. Меньше (<)\n");
    printf("3. Равно (=)\n");
    printf("4. Меньше или равно (<=)\n");
    printf("5. Больше или равно (>=)\n");
    int sign = inputInt("Выбор: ");

    int foundIndices[MAX_COUNTRIES];
    int foundCount = 0;

    for (int i = 0; i < count; i++)
    {
        if (sign == 1 && db[i].hdiRank > value)
            foundIndices[foundCount++] = i;
        else if (sign == 2 && db[i].hdiRank < value)
            foundIndices[foundCount++] = i;
        else if (sign == 3 && db[i].hdiRank == value)
            foundIndices[foundCount++] = i;
        else if (sign == 4 && db[i].hdiRank <= value)
            foundIndices[foundCount++] = i;
        else if (sign == 5 && db[i].hdiRank >= value)

```

```

        foundIndices[foundCount++] = i;
    }

    printFilterResult(foundIndices, foundCount);
}
}

```

Файл **filtres.h**

```

#ifndef FILTRES_H
#define FILTRES_H
void filterMenu();
#endif

```

Файл **main.c**

```

#include "database.h"
#include "utils.h"
#include "file_manager.h"
#include "filtres.h"
#include "sort.h"
#include "search.h"

void menu();

int main()
{
    setRussianLocale();

    while (1)
    {
        menu();
        int c = inputInt("Выберите действие: ");

        if (c == 0) {
            printf("До свидания!\n");
            return 0;
        }
        if (c == 1) createDatabase();
        if (c == 2) addCountry();
        if (c == 3) deleteCountry();
        if (c == 4) editCountry();
        if (c == 5) showDatabase();
        if (c == 6) filterMenu();
        if (c == 7) sortMenu();
        if (c == 8) searchCountry();
        if (c == 9) saveDatabase();
        if (c == 10) loadDatabase();
        if (c == 11) loadDemoData();

        pause();
    }
}

```

```

void menu()
{
    clearScreen();

    setColor(11);
    printf("=====\n");
    printf("        БАЗА ДАННЫХ СТРАН МИРА\n");
    printf("=====\n");
    resetColor();

    printf("\n");
    setColor(10);
    printf("ОСНОВНЫЕ ОПЕРАЦИИ:\n");
    resetColor();
    printf("1. Создать новую БД\n");
    printf("2. Добавить страну\n");
    printf("3. Удалить страну\n");
    printf("4. Редактировать страну\n");
    printf("5. Показать таблицу\n");

    printf("\n");
    setColor(10);
    printf("ОБРАБОТКА ДАННЫХ:\n");
    resetColor();
    printf("6. Фильтрация\n");
    printf("7. Сортировка\n");
    printf("8. Поиск страны\n");

    printf("\n");
    setColor(10);
    printf("ФАЙЛОВЫЕ ОПЕРАЦИИ:\n");
    resetColor();
    printf("9. Сохранить БД\n");
    printf("10. Загрузить БД\n");

    printf("\n");
    setColor(10);
    printf("ДОПОЛНИТЕЛЬНО:\n");
    resetColor();
    printf("11. Загрузить демо-данные\n");

    printf("\n");
    setColor(12);
    printf("0. Выход\n");
    resetColor();
    printf("=====\n");
}

```

Файл search.c

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

```

```

#include <string.h>
#include "database.h"
#include "utils.h"

void searchCountry()
{
    char name[MAX_STR];

    inputString("Введите название страны: ", name, MAX_STR);

    int found = 0;
    for (int i = 0; i < count; i++)
    {
        if (strcmp(name, db[i].name) == 0)
        {
            printCountry(db[i]);
            found++;
        }
    }

    if (!found)
        printf("Страна '%s' не найдена\n", name);
}

void searchByCapital()
{
    char capital[MAX_STR];

    inputString("Введите название столицы: ", capital, MAX_STR);

    int found = 0;
    for (int i = 0; i < count; i++)
    {
        if (strcmp(capital, db[i].capital) == 0)
        {
            printf("Страна: %s\n", db[i].name);
            found++;
        }
    }

    if (!found)
        printf("Столица '%s' не найдена\n", capital);
}

```

Файл search.h

```

#ifndef SEARCH_H
#define SEARCH_H

void searchCountry();
void searchByCapital();

#endif

```

Файл sort.c

```
#include "database.h"
#include "utils.h"

// Функция для обмена двух структур
void swapCountries(Country* a, Country* b)
{
    Country temp = *a;
    *a = *b;
    *b = temp;
}

// Сортировка по населению
void sortByPopulation(int ascending)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = i + 1; j < count; j++)
        {
            if (ascending)
            {
                if (db[i].population > db[j].population)
                    swapCountries(&db[i], &db[j]);
            }
            else
            {
                if (db[i].population < db[j].population)
                    swapCountries(&db[i], &db[j]);
            }
        }
    }
}

// Сортировка по ВВП
void sortByGDP(int ascending)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = i + 1; j < count; j++)
        {
            if (ascending)
            {
                if (db[i].gdp > db[j].gdp)
                    swapCountries(&db[i], &db[j]);
            }
            else
            {
                if (db[i].gdp < db[j].gdp)
                    swapCountries(&db[i], &db[j]);
            }
        }
    }
}
```

```

}

// Сортировка по площади
void sortByArea(int ascending)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = i + 1; j < count; j++)
        {
            if (ascending)
            {
                if (db[i].area > db[j].area)
                    swapCountries(&db[i], &db[j]);
            }
            else
            {
                if (db[i].area < db[j].area)
                    swapCountries(&db[i], &db[j]);
            }
        }
    }
}

// Сортировка по ИЧР (чем меньше число, тем лучше)
void sortByHDI(int ascending)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = i + 1; j < count; j++)
        {
            if (ascending)
            {
                if (db[i].hdiRank > db[j].hdiRank)
                    swapCountries(&db[i], &db[j]);
            }
            else
            {
                if (db[i].hdiRank < db[j].hdiRank)
                    swapCountries(&db[i], &db[j]);
            }
        }
    }
}

// Главное меню сортировки
void sortMenu()
{
    if (count == 0)
    {
        printf("База данных пуста. Нечего сортировать.\n");
        return;
    }
}

```

```

while (1)
{
    printf("\n===== МЕНЮ СОРТИРОВКИ =====\n");
    printf("1. По населению\n");
    printf("2. По ВВП\n");
    printf("3. По площади\n");
    printf("4. По ИЧР\n");
    printf("0. Назад\n");

    int choice = inputInt("Выбор: ");

    if (choice == 0) return;

    if (choice >= 1 && choice <= 4)
    {
        printf("\nВыберите направление сортировки:\n");
        printf("1. По возрастанию (от меньшего к  

        большему) \n");
        printf("2. По убыванию (от большего к меньшему) \n");
        int order = inputInt("Выбор: ");

        if (order == 1 || order == 2)
        {
            int ascending = (order == 1);

            switch (choice)
            {
            case 1:
                sortByPopulation(ascending);
                printf("\nСортировка по населению выполнена  

                ");
                printf(ascending ? "(по возрастанию)!\n" :  

                "(по убыванию)!\n");
                break;
            case 2:
                sortByGDP(ascending);
                printf("\nСортировка по ВВП выполнена ");
                printf(ascending ? "(по возрастанию)!\n" :  

                "(по убыванию)!\n");
                break;
            case 3:
                sortByArea(ascending);
                printf("\nСортировка по площади выполнена  

                ");
                printf(ascending ? "(по возрастанию)!\n" :  

                "(по убыванию)!\n");
                break;
            case 4:
                sortByHDI(ascending);
                printf("\nСортировка по ИЧР выполнена ");
                printf(ascending ? "(по возрастанию - лучшие  

                первые)!\n" : "(по убыванию - худшие первые)!\n");
                break;
            }
        }
    }
}

```

```

    }

    printf("\nРезультат сортировки:\n");
    printTableHeader();
    for (int i = 0; i < count; i++)
        printTableRow(i);
    printTableFooter();
}
else
{
    printf("Неверный выбор. Сортировка
отменена.\n");
}
else
{
    printf("Неверный выбор. Попробуйте снова.\n");
}
}
}

```

Файл sort.h

```

#ifndef SORT_H
#define SORT_H

void sortByPopulation(int ascending);
void sortByGDP(int ascending);
void sortByArea(int ascending);
void sortByHDI(int ascending);
void sortMenu();

#endif

```

Файл utils.c

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <windows.h>
#include <locale.h>
#include "utils.h"

void setRussianLocale()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
}

int inputInt(const char* msg)
{
    char buf[100];
    int x;
    while (1)

```

```

    {
        printf("%s", msg);
        if (!fgets(buf, sizeof(buf), stdin)) continue;
        if (sscanf(buf, "%d", &x) == 1) return x;
        printf("Invalid input\n");
    }
}

double inputDouble(const char* msg)
{
    char buf[100];
    double x;
    while (1)
    {
        printf("%s", msg);
        if (!fgets(buf, sizeof(buf), stdin)) continue;
        if (sscanf(buf, "%lf", &x) == 1) return x;
        printf("Invalid input\n");
    }
}

void inputString(const char* msg, char* buffer, int size)
{
    printf("%s", msg);
    fgets(buffer, size, stdin);
    buffer[strcspn(buffer, "\n")] = 0;
}

void inputValidName(const char* msg, char* buffer, int size)
{
    while (1)
    {
        inputString(msg, buffer, size);

        int valid = 1;
        for (int i = 0; buffer[i]; i++)
        {
            char c = buffer[i];
            if (!(c >= 'A' && c <= 'Я' || (c >= 'a' && c <=
'z') ||
                (c >= 'A' && c <= 'Z') || c == ' ' || c == '-'))
            {
                valid = 0;
                break;
            }
        }

        if (valid && strlen(buffer) > 0)
            return;

        printf("Ошибка! Используйте только буквы, пробелы и
дефис.\n");
    }
}

```

```

}

void setColor(int c)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), c);
}

void resetColor()
{
    setColor(7);
}

void clearScreen()
{
    system("cls");
}

void pause()
{
    printf("\nНажмите ENTER для продолжения...");
    getchar();
}

```

Файл utils.h

```

#ifndef UTILS_H
#define UTILS_H

int inputInt(const char* msg);
double inputDouble(const char* msg);
void inputString(const char* msg, char* buffer, int size);
void inputValidName(const char* msg, char* buffer, int size);

void setColor(int color);
void resetColor();
void clearScreen();
void pause();
void setRussianLocale();

#endif

```

Приложение В. Результаты работы программы

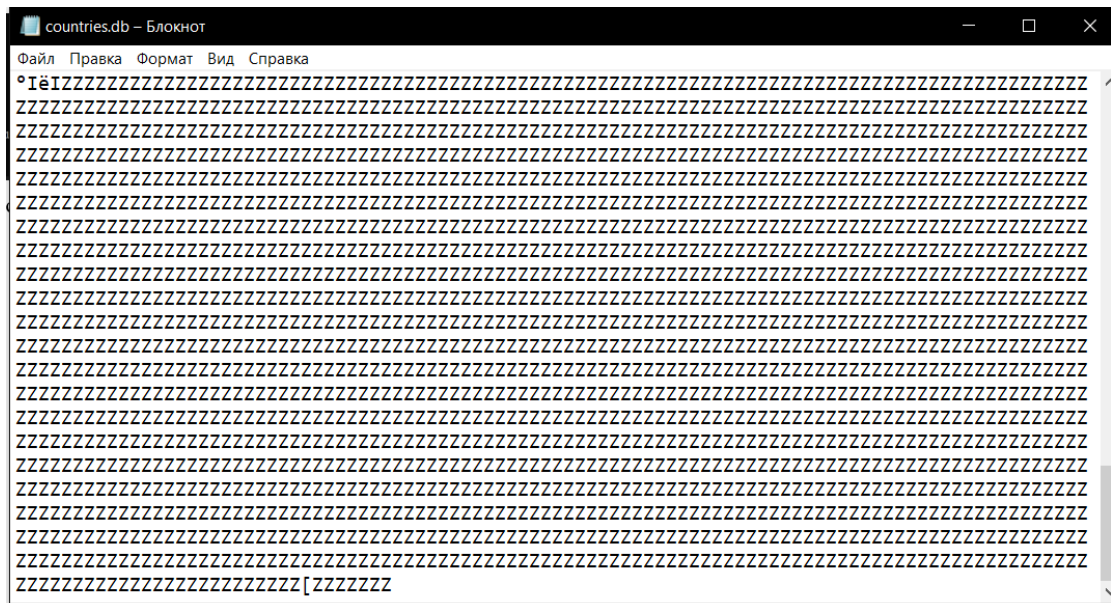
Приложение В.1 – Информационная заставка

```
0. Выход
=====
Выберите действие: 0
Курсовая работа
По дисциплине "Программирование"
На тему: "База данных стран"
Выполнил студент группы 25ВВВ2: Никоноров Семён
Принял: доцент, к.т.н.
Генералова А.А
D:\Users\Пользователь\source\repos\Kurs\Debug\Kurs.exe (процесс 11760) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно...
```

На информационной заставке представлены данные о дисциплине, теме, учащегося и об преподавателе.

Приложение С. Результат работы программы

Файл countres.bd



В файле находится зашифрованная база данных.