Условие задачи:

Создать таблицу, содержащую не менее 40-ка записей (тип – запись с вариантами (объединениями)). Упорядочить данныев ней по возрастанию ключей, двумя алгоритмамисортировки, где ключ – любое невариантное поле (по выбору программиста), используя: а) саму таблицу, б) массив ключей. (Возможность добавления и удаления записей в ручном режиме обязательна). Осуществить поиск информации по варианту.

Задание для поиска информации:

Ввести список литературы, содержащий фамилию автора, название книги, издательство, количество страниц, вид литературы (1: техническая – отрасль, отечественная, переводная, год издания; 2: художественная – роман, пьеса, поэзия; 3: детская – минимальный возраст, сказки, стихи). Вывести список всех романов указанного автора.

Техническое задание:

a) Исходные данные.

1) Но вход программе подается текстовый файл с записями. Записи расположены таким образом, что каждое поле записи находится в отдельной строчке. Максимальное количество записей в файле – 140. Максимальная длина строковой записи - 100 символов. Числовые данные ограничены целым типом.

2) Поля структуры – строки и целые числа. Строки могут состоять из любых символов, поля с числами содержат целые числа.

3) В конце файла нет пустой строки.

b) Задача, реализуемая программой.

Программа выполняет следующие действия:

1) Удаляет строчку по полю имени книги.
2) Добавляет запись в конец файла.
3) Производит поиск романов по имени автора. Тоесть выводит все романы конкретного автора.
4) Выводит отсортированную таблицу ключей.
5) Упорядочивает исходную таблицу и выводит ее.
6) Упорядочивает исходную таблицу по отсортированной таблице ключей.

7) Выводит информацию о затраченном на сортировку времени и использованной памяти.
0) Завершает программу и записывает получившийся массив в тот же текстовый файл, откуда мы взяли информацию для начала программы.

c) Способ обращения к программе.

Запуск программы происходит из командной строки, мы указываем исполняемый файл и файл с записанными структурами.

d) Возможные аварийные ситуации и ошибки пользователя.

Ситуация	Вывод	Код возврата
Пустой файл	Ошибка! Пустой файл.	14
Несуществующий файл	Ошибка! Неправильно введено название файла.	13
Ошибка запуска программы	Ошибка! Неверный формат командной строки.	22222
Ошибка выбора команды	Ошибка! Вы ввели не номер действия.	1
Ошибки чтения полей	Ошибка! Неправильно введен (имя поля).	2-12
Отсутствие совпадений при поиске по полю и удалении	Ошибка! Совпадений не найдено.	15
Ошибка при сохранении файла	Ошибка! Неверно введено подтверждение или опровержение сохранения.	17
Ошибка при заполнении массива ключей	Ошибка! Проблема в заполнении массива ключей.	16
Переполнение массива ключей или массива структур	Ошибка! В файле слишком много записей.	33

Внутренние структуры данных:

В программе используются структуры:

Структура для хранения полей вариантной части структуры, если тип литературы – “художественная”.

```
typedef struct
{
char type[LENGTH];
} s_artistic;
```

Структура для хранения полей вариантной части структуры, если тип литературы - “техническая”.

```
typedef struct
{
char branch[LENGTH];
char country[LENGTH];
int year;
} s_tech;
```

Структура для хранения полей вариантной части структуры, если тип литературы – “детская”.

```
typedef struct
{
int min_age;
char type[LENGTH];
} s_child;
```

Объединение, оно включает в себя три предыдущие структуры.

```
typedef union
{
s_tech tech;
s_artistic artistic;
s_child child;
} u_type;
```

Структура содержит описание невариантных плей и предыдущее объединение.

```
typedef struct
{
char surname[LENGTH];
char book_name[LENGTH];
char publication[LENGTH];
int page_number;
char literature_type[LENGTH];
```

```
u_type type;
} book;
```

Структура описывает поля для таблицы ключей.

```
typedef struct
{
int base_num;
int page_number;
} key_table;
```

Также используются два массива, для хранения полных структур:

```
book all[INITIAL + LENGTH];
```

и для хранения ключей:

```
key_table key[INITIAL + LENGTH];

LENGTH = 100
INITIAL = 40
```

Алгоритм работы программы:

1) Программа выводит краткую информацию о содержании входного файла и меню дейсствий с краткими пояснениями как пользоваться программой.
2) Пользователь вводит номер команды, которую должна выполнить программа.
3) В зависимости от выбора пользователя совершаются разные действия:
    1) Если выбрана команда 1, то программа перемещает поле с введенным названием в конец массива и сокращает его длину.
    2) Если выбрана команда 2, то программа заполняет следующую после последней уже записанной в массив структуры.
    3) Команда 3, программа сравнивает введенное имя автора и имена авторов из массива структур и контролирует чтобы произведение было романом.
    4) 4 - сортирует массив ключей сортировкой пузырьком и выводит его.
    5) 5 - сортирует массив полных структур сортировкой пузырьком и выводит его.
    6) 6 – сортирует массив ключей и выводит массив полных структур согласно отсортированному массиву ключей.
    7) 7 - делает 1000 замеров сортировки для каждого из случаев: сортировка пузырьком для массива полных структур, сортировка пузырьком для массива ключей, быстрая сортировка для массива полных структур, быстрая сортировка для массива ключей. Потом считает на сколько процентов сортировка пузырьком для массива полных структур

медленнее, чем для массива ключей и наоборот, те же замеры делаютс для быстрой сортировки.
0) 0 – сохраняет изменения в исходный файл и завершает программу с кодом 0.

Тесты:

Описание теста	**Результат**
Пустой файл	Ошибка! Пустой файл.
Отсутствие файла	Ошибка! Неправильно введено название файла.
Неправильный ввод при поиске по полю или добавлении структуры в таблицу	Ошибка! Неправильно введено (название поля).
Отсутствие совпадений при поиске по полю	Ошибка! Совпадений не найдено.
Ошибка командной строки	Ошибка!
Ошибка ввода номера действия программы	Ошибка! Вы ввели не номер действия.
Удаление записи по полю названия книги	В памяти будет хранится массив с передвинутым в конец полем(полями в случае еслих несколько), которое(ые) нужно было удалить и значение количества рядов таблицы будет на 1 (несколько) меньше
Добавление записи в конец таблицы	В памяти будет массив с добавленным в конец полем и значение количестваа рядов в таблице увеличится на 1
Провести поиск романов в таблице по полю фамилии автора	Будут выведены романы конкретного автора
Просмотр отсортированной таблицы ключей при неотсортированной исходной	Выведет отсортированную таблицу ключей
Вывод упорядоченной исходной таблицы.	Выведет отсортированную таблицу полных структур
Вывод исходной таблицы в упорядоченном виде, используя упорядоченную таблицу ключей.	Выведет отсортированную таблицу полных структур
Вывод результатов сравнения	Выведет таблицу с результатами

эффективности работы программы при обработке данных в исходной таблице и таблице ключей.	замеров сортировки разных массивов разными способами.
Закончить выполнение программы	Запишет получившийся массив в исходный файл и закончит выполнение программы
Ошибка при ответе на запрос о сохранении массива структур в файл	Ошибка! Неверно введено подтверждение или опровержение сохранения.

Вывод:

В данной лабораторной работе для хранения записей используется структура с объединением внутри. Таким образом мы экономим память, чтобы не использовать большую структуру со всеми возможными полями, мы вариантные поля записываем в объединение.

При сортировке массива с целыми структурами мы используем меньше памяти, но она в несколько раз медленнее сортировки таблицы ключей. Поэтому сильно быстрее сортировать таблицу с двумя полями и выводить основную таблицу соглясно полученному порядку. Также таблица с двумя полями занимает немного места по сравнению с таблицей полных структур.

Способ сортировки крупных таблиц посредством выноса в отдельный массив ключа и его порядка в исходном массиве и сортировки этого маленького массива является наиболее удобным и быстрымб а также не знимает много памяти.

Поэтому мы можем оптимизировать работу программы:

1) Использовать в структуре объединения, что уменьшит размер используемой памяти.

2) Сортировать таблицу можно посредством таблицы ключей, которая имеет только 2 поля, что намного легче и быстрее отсортировать.

3) Во время сортировки ключей мы будем использовать дополнительный массив, который будет занимать некоторое место, но по сравнению с таблицей полных структур эта память очень маленькая.

Результаты замерного эксперимента:

Длина массива	Время СП для полного массива, сек	Время БС для полного массива, сек	Время СП для массива ключей, сек	Время БС для массива ключей, сек	Размер массива полных структур, байт	Размер массива полных ключей и массива структур, байт
15	0.000013	0.000002	0.000001	0.000001	9120	9240
160	0.000925	0.000013	0.000072	0.000005	97280	98560
480	0.007994	0.000043	0.000633	0.000016	291840	295680

Контрольные вопросы:

1.Как выделяется память под вариантную часть записи?

Память выделяется под максимальный элемент объединения, таким образом в любом случае введенные значения поместятся.

2. Что будет, если в вариантную часть ввести данные, несоответствующие описанным?

Программа не проверяет что мы ввели в объединение, поэтому не понятно как в дальнейшем будет обрабатываться запись с неверно введенной вариантной частью.

3. Кто должен следить за правильностью выполнения операций с вариантной частью записи?

За правильным выполнением операций с вариантной частью записи должен следить программист.

4. Что представляет собой таблица ключей, зачем она нужна?

Таблица ключей – таблица с двумя полями – индекс в исходной таблице и поле, по которому будет сортироваться таблица структур. Эта таблица используется для более быстрой сортировки исходной таблицы.

5. В каких случаях эффективнее обрабатывать данные в самой таблице, а когда – использовать таблицу ключей?

В случаях, если таблица должна хранится в памяти в измененном виде, легче изменять саму таблицу, чем делать это посредством таблицы ключей. Если таблица должна выглядет измененной только для пользователя, намного легче и быстрее менять ее посредством таблицы ключей.

6. Какие способы сортировки предпочтительнее для обработки таблиц и почему?

В условиях современных реалий лучше использовать сортровки, которые быстрее. У большинства ЭВМ есть возможность использовать немного больше памяти. Поэтому для нас удобнее и быстрее использовать сортировку с использованием таблицы ключей.