|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**«Записи с вариантами. Обработка таблиц»**

Преподаватель: Силантьева А. В.

Студент: Динь Вьет Ань, ИУ7И - 34Б

**Цель работы:**

Приобрести навыки работы с типом данных «запись» (структура), содержащим вариантную часть (объединение, смесь), и с данными, хранящимися в таблицах, произвести сравнительный анализ реализации алгоритмов сортировки и поиска информации в таблицах, при использовании записей с большим числом полей, и тех же алгоритмов, при использовании таблицы ключей; оценить эффективность программы по времени и по используемому объему памяти при использовании различных структур и эффективность использования различных алгоритмов сортировок.

**Оисание условие задачи:**

Создать таблицу, содержащую записи с вариантной частью. Произвести поиск информации по вариантному полю. Упорядочить таблицу, по возрастанию ключей (где ключ – любое невариантное поле по выбору программиста), используя:

а) исходную таблицу;

б) массив ключей,

(возможность добавления и удаления записей в ручном режиме обязательна).

В таблице содержится информация о студентах. Основная часть: тип жилья (дом, общежитие), имя, пол, возраст, средний балл за сессию, год поступления. Вариантная часть делится на две подгруппы:

Дом: название улицы, номер дома, номер квартиры.

Общежитие: номер общежития, номер комнаты.

Ввести общий список студентов группы. Вывести список студентов, живущих в общежитии указанного года поступления. Реализовать возможность добавления записей в конец таблицы и удаления записи по значению указанного поля.

**Техническое задание**

**Входные данные:**

1. Файл с данными: текстовый файл формата TXT. В файле строковое или целочисленное поле, в зависимости от вводимой информации.

2. Целое число, представляющее собой номер команды: целое число в диапазоне от 0 до 10.

3. Дополнения к таблице: строковое или целочисленное поле, в зависимости от вводимой информации.

**Выходные данные:**

1.Полученная таблица (основная или таблица ключей) в отсортированном или неотсортированном виде (в зависимости от выполненной команды).

2.Характеристика сравнения вариантов сортировки таблицы.

**Bозможные ошибки**

1. Некорректный ввод номера команды.

На входе: число, большее чем 11 или меньшее, чем 0 или пустой ввод.

На выходе: сообщение « Error input»

2. Файл пуст.

На входе: пустой файл.

На выходе: сообщение « The file is empty »

3. Файл не существует или неверное имя

На входе: файл не существует или неверное имя

На выходе: сообщение « File does not exist »

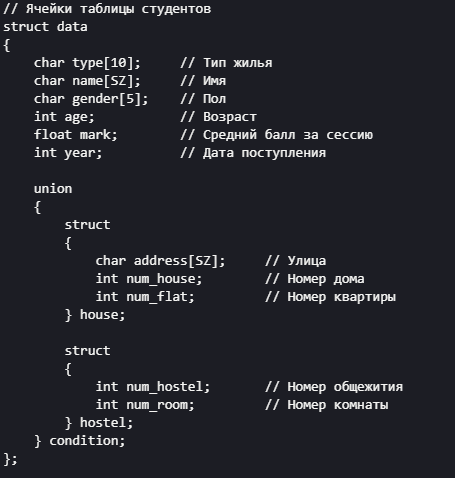
4. Информация о студенте неверна

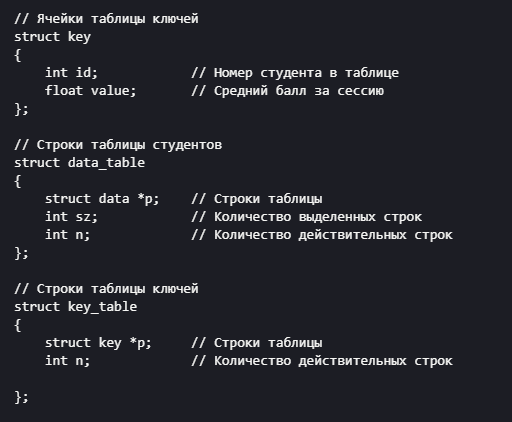
На входе: Информация о студенте неверна

На выходе: сообщение зависит от неверной информации

**Структуры данных:**







**Функции:**

1. Загрузить таблицу из файла (f\_load\_table).

2. Добавить запись в конец таблицы (f\_add).

3. Удалить запись из таблицы (f\_remove).

4. Cохранить таблицу в файл (f\_save\_table).

5. Вывести таблицу на экран. (f\_show\_table).

6. Cоздать ключи (f\_create\_key\_table)

7. Вывести таблицу ключи на экран. (f\_show\_key\_table)

8. Поиск (f\_search)

9. Проверить ввходные данные (f\_check)

10. Отсортировать таблицу ключей способом сортировки пузырьком (f\_sort\_bubble\_key) и способом сортировки вставками (f\_sort\_insert\_key)

11. Отсортировать таблицу по ключам способом сортировки пузырьком (f\_sort\_table\_bubble) и способом сортировки вставками (f\_sort \_table)

**ОПИСАНИЕ АЛГОРИТМА**

1. Пользователь вводит номер команды из предложенного меню.

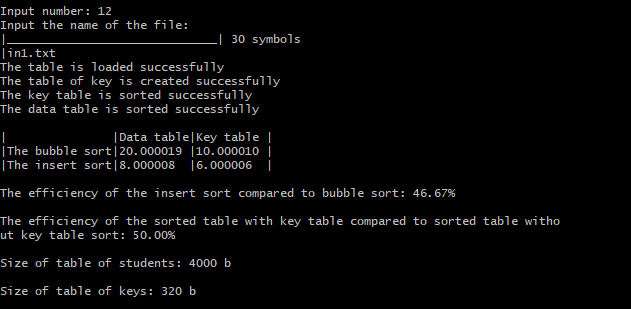
2. Пока пользователь не введет 0 (выход из программы), ему будет предложено вводить номера команд и выполнять действия по выбору.

**Тесты:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название теста** | **Пользовательский ввод** | **Результат** |
| 1 | Некорректный ввод команды | 45 | Error input |
| 2 | Некорректный ввод из файла (файл пустой) | 1 (команда)  in.txt | The file is empty |
| 3 | Некорректный ввод из файла (файл не существует) | 1 (команда)  in0.txt | File does not exist |
| 4 | Тип жилья неверна | adad | Error input |
| 5 | Некорректный ввод для строкового поля (имя или улица) (длинная) | qq...qq (31 буква) | There are more than 30 symbols |
| 6 | Некорректный ввод для строкового поля (имя или улица) (первая не большая) | anton  alex | The first letter must be uppercase |
| 7 | Имя студента состоит из более чем 1 слова | An ton | There must be 1 word |
| 8 | Некорректный ввод численного поля (буква) | q | Error input |
| 9 | Неправильный пол | aaaa | Error input |
| 10 | Средний балл > 5 или < 2 | 6 | Error input |
| 11 | Ввод записей из верного файла (команда 1) | команда 1  in1.txt | The table is loaded successfully |
| 12 | Добавление верной записи в конец файла (команда 4) | Введена команда 4  Запись введена верно | The information about student is added successfully |
| 13 | Удаление из таблицы (команда 5) | Введена команда 5 и указано не существующее значение в поле удаление | The number is not in range |
| 14 | Удаление из таблицы (команда 5) | Введена команда 5 и указано существующее значение в поле удаление. | The element is deleted successfully |
| 15 | Вывод таблицы на экран (команда 3) | Команда введена верно. | Таблица выведена на экран. |
| 16 | Вывод таблицы на экран (команда 3) | Команда введена верно  Таблица пуста | The table is empty |
| 17 | Поиск (команда 11) | Введен год (найден) | вывод на экране |
| 18 | Поиск (команда 11) | Введен год (не найден) | There are not such student |
| 19 | Cоздать ключи | команда 6 | The table of key is created successfully / The table data is empty |
| 20 | Команда сортировки (8, 9) | команда 8  команда 9 | The data table is sorted successfully / The key table is sorted successfully |

**Сравнительный анализ**

Если есть необходимость хранения различных вариативных данных при различных данных, находящихся в основной части, то для того, чтобы не создавать отдельную запись, а хранить всё в одном блоке, целесообразно использовать запись с вариантной частью.



Время сортировки без таблицы ключей

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сортировка пузырьком   |  |  | | --- | --- | | Размерность таблицы | Время(микро сен.) | | 40 | 20.000019 | | 50 | 35.000034 | | Сортировка вставками.   |  |  | | --- | --- | | Размерность таблицы | Время(микро сен.) | | 40 | 8.000008 | | 50 | 10.000010 | |

Время сортировки с таблицей ключей

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сортировка пузырьком   |  |  | | --- | --- | | Размерность таблицы | Время(микро сен.) | | 40 | 10.000010 | | 50 | 11.000010 | | Сортировка вставками.   |  |  | | --- | --- | | Размерность таблицы | Время(микро сен.) | | 40 | 6.000006 | | 50 | 7.000007 | |

Для реализации поставленной задачи были выбраны сортировка

вставками и сортировка пузырьком.

Таблицы ключей используются для ускорения процесса сортировки элементов исходной таблицы. Наиболее эффективным данный метод становится при увеличении исходной таблицы.

**Выводы**

В ходе данной лабораторной работы были приобретены навыки работы с типом данных структура, содержащим вариантную часть, и с данными, хранящимися в таблицах. После лабораторной работы узнать, что таблица ключей позволяет уменьшить время сортировки 2 – 3 раза в всех структурой, но надо память для сохранить дополнительную таблицу ключей (< 10% размера структура).

**ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

**1.Как выделяется память под вариантную часть записи?**

Размер памяти, который выделяется под вариантную часть, равен максимальному по длине полю вариантной части. Эта память является общей для всех полей вариантной части записи.

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

При компиляции тип данных в вариантной части не проверяется. Поведение будет неопределенно из-за того, что невозможно корректно считать данные.

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

За правильностью выполнения операций с вариантной частью должен следить программист.

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

Таблица ключей представляет собой дополнительный массив (структура), содержащий индекс исходного элемента в исходной таблице и выбранный ключ.

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

Когда мы сортируем таблицу ключей, мы экономим время, так как перестановка записей в основной таблице (может содержать большой количество полей) отсутствует. Минус данного подхода в том, что для размещения таблицы ключей требуется дополнительная память. Кроме того, если использовать в качестве ключа символьное поле, то необходимо будет дополнительно обрабатывать данное поле в цикле, что увеличивает время выполнения, так же выбор данных из основной таблицы в порядке, определенном таблицей ключей, замедляет вывод. Если исходная таблица содержит небольшие число полей, то выгоднее обрабатывать данные в самой таблице.

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

Если будет производится сортировка самой таблицы, то необходимо использовать алгоритмы, требующие наименьшее количество операций перестановки. Если же сортировка производится по таблице ключей, то эффективнее использовать сортировки с наименьшей сложностью работы.