МГТУ им. Н.Э.Баумана

Дисциплина “Типы и структуры данных”

Отчет по Лабораторной работе №2 по теме “Записи с вариантами. Обработка таблиц”.

Работу выполнил:

Воякин А.Я.

Группа ИУ7-34Б

Вариант №5

*Описание задачи:*

*Создать таблицу, содержащую не менее 40-ка записей (тип – запись с вариантами). Упорядочить данные в ней по возрастанию ключей, где ключ – любое невариантное поле (по выбору программиста), используя: а) саму таблицу, б) массив ключей (возможность добавления и удаления записей в ручном режиме обязательна).* Ввести репертуар театров, содержащий: название театра, спектакль, режиссер, диапазон цены билета, тип спектакля: детский – для какого возраста, тип (сказка, пьеса, музыкальный); взрослый - тип (пьеса, драма, комедия); музыкальный – композитор, страна, минимальный возраст, продолжительность. Вывести список всех музыкальных спектаклей для детей указанного возраста с продолжительностью меньше указанной.

*Описание ТЗ*

*Введение*:

*Наименование программы* - “Табличный редактор”.

Программа предназначена к применению в учебных целях.

*Основания для разработки*:

Основанием для проведения разработки является Лабораторная работа № 2, выданная в МГТУ им. Н.Э. Баумана в учебных целях.

*Назначение разработки*:

*Функциональное назначение программы*: предоставление пользователю возможностей удаления записей из таблицы по индексу, добавление записей в конец таблицы, сортировки таблицы по ключам, сортировки самой таблицы, сравнения эффективности сортировок.

*Эксплуатационное назначение программы*: программа должна использоваться в ходе проведения Лабораторной работы № 2 в МГТУ им. Н.Э. Баумана, пользователями программы должны являться преподаватели и студенты университета.

*Требования к программе или программному изделию*:

* *Требования к функциональным характеристикам*: программа должна выводить пункты меню и предлагать пользователю ввести пункт меню, пока он не введет корректный, либо пока не введет пункт выхода из программы. Программа должна давать возможность добавлять новую запись в конец таблицы, удалять запись по ее индексу, сортировать ключи и саму таблицу, сравнивать эффективность сортировки по ключам и самой таблицы, эффективность сортировки ключей быстрой сортировкой и сортировкой пузырьком. При некорректном вводе числовых или символьных значений программа не должна завершаться аварийно и должна продолжить просить ввести значение, пока оно не окажется в верном формате. Поля название театра и страна должны содержать не более 35 символов, поля режиссер, композитор, название спектакля – не более 40 символов, остальные значения целочисленные.
* *Требования к надежности*: программа должна воспринимать любой ввод и не завершаться аварийно. При вводе целочисленных значений должны проверяться введенные символы и диапазон возможных значений. При вводе символьных значений проверяется только длина введенной строки. При некорректном вводе пользователю должно быть выведено сообщение об этом и предложено ввести значение снова.
* *Требования к эксплуатации*: Программа не требует особых климатических условий. Программа не требует проведения каких-либо видов обслуживания. Для нормального функционирования программы достаточно одного человека - пользователя.

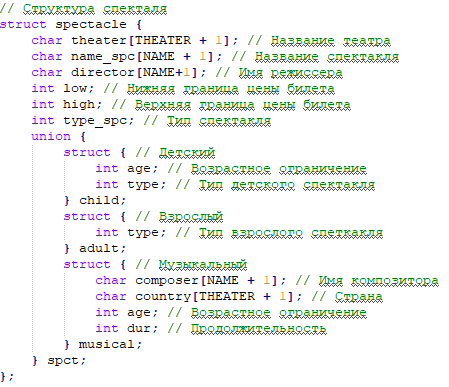
*Описание СД*

Для работы с таблицами были выбраны типы данных структуры и объединения, которые могут содержать в себе поля различных типов. Таблица хранится в виде массива структур, ключи хранятся в виде матриц, в первом столбце которых хранятся индексы таблицы, во втором – значения ключей.

!!! Гораздо рациональнее было бы использовать таблицы ключей вместо матриц, это позволило бы увеличить выгоду по времени для сортировки по ключам. Для выбранной мной структуры на малых размерах таблицы рациональнее использовать сортировку таблицы, нежели сортировку матрицы ключей.

В программе были использованы:

* структура для каждой записи таблицы, которая содержит 3 символьных статических массива char, 3 целочисленных переменных int , вариативную часть из 3 структур, максимальная из которых содержит 2 символьных статических массива char и 2 целочисленных переменных int, размер одной записи – 256 байт



* статический массив структур для хранения таблицы, размер – 10240 байт



* массивы типа char для хранения символьных переменных, 3 по 40 байт, 2 по 35 байт



* целочисленная матрица типа int для хранения индексов таблицы и соответствующих им ключей – 360 байт



* переменные типа int для хранения целочисленных значений структур и других рабочих переменных – 4 байта каждая.



***Входные данные***

* Файл с записанной заранее таблицей.
* Номер пункта меню.
* Строка или целое число, в зависимости от выбранного пункта меню.

***Выходные данные***

* Таблица ключей
* Таблица по записям
* Вывод результатов поиска
* Таблица результатов эффективности сортировок

***Алгоритм***

* Запуск цикла бесконечного вызова меню, который прерывается вводом символа, который не соответствует ни одной из команд
* Команда чтения таблицы
  + посимвольно в цикле считывается файл, пока не будет достигнут конец файла
* Добавление записи
  + последовательно ввести значения всех полей добавляемой записи, в случае некорректных данных – вывести сообщение
  + увеличить счётчик количества записей.
* Удаление записи
  + Прочитать индекс (номер по порядку) удаляемой записи (нумерация с 1). В случае недопустимого индекс вернуть ошибку.
  + Сдвинуть все элементы после данного на один назад
  + Уменьшить счётчик количества записей;
* Поиск музыкальных детских постановок указанного возраста и продолжительности
  + Ввести минимальный допустимый возраст и максимальную продолжительность, в случае некорректных данных вернуть сообщение
  + Последовательно в таблице сравнивать полученные значения со значениями полей записей
  + В случае соответствия вывести запись
  + Если не было найдено соответствующих записей, вывести сообщение
* Сортировка
  + Если следует выполнять сортировку по ключам, составить таблицу ключей на основе исходной таблицы, записывая индексы и значения максимальной цены за билет

Выполнить сортировку по алгоритму сортировки «пузырьком» или быстрой сортировки

*Набор тестов:*

* Некорректные символьные значения

|  |  |
| --- | --- |
| Ввод | Вывод программы |
| Режисcер: Ааа..аа(более 40) | Некорректный ввод, введите еще раз |
| Театр: Аааа..аа(более 35) | Некорректный ввод, введите еще раз |
| Пустой ввод | Некорректный ввод, введите еще раз |

* Некорректные целочисленные значения

|  |  |
| --- | --- |
| Ввод | Вывод программы |
| Возрастное ограничение: ааа | Некорректный ввод, введите еще раз |
| Возрастное ограничение: 30 | Некорректный ввод, введите еще раз |
| Продолжительность: -5 | Некорректный ввод, введите еще раз |
| Пустой ввод | Некорректный ввод, введите еще раз |
| Выберите действие (1,2,3): 4 | Некорректный ввод, введите еще раз |

* Ошибки при действиях в меню

|  |  |
| --- | --- |
| Ввод | Вывод программы |
| Сортировка ключей (массив пуст) | В таблице нет записей |
| Вывод таблицы (массив пуст) | В таблице нет записей |
| Удаление записи (массив пуст) | В таблице нет записей |
| Поиск(нет подходящих записей) | Записей не найдено |
| Добавление записи (массив полон) | Достигнут максимум записей |

* Результаты сравнения сортировки ключей и сортировки таблицы

**Сортировка 40-ка записей.**

|  |  |  |
| --- | --- | --- |
|  | Сортировка ключей(такты) | Сортировка таблицы(такты) |
| Пузырьком | 18700 | 29340 |
| Быстрая | 7590 | 17604 |

Быстрая сортировка быстрее сортировки пузырьком на 60 % и в случае сортировки таблицы, и в случае сортировки ключей.

Сортировка ключей быстрее сортировки таблицы на 55%, при этом матрица ключей занимает лишь 1.9 % памяти относительно таблицы структур.

**Сортировка 160 записей.**

|  |  |  |
| --- | --- | --- |
|  | Сортировка ключей(такты) | Сортировка таблицы(такты) |
| Пузырьком | 220796 | 134780 |
| Быстрая | 27707 | 80868 |

Быстрая сортировка быстрее сортировки пузырьком на 87 % и в случае сортировки таблицы, и в случае сортировки ключей.

Сортировка ключей быстрее сортировки таблицы на 65%, при этом матрица ключей занимает лишь 1.9 % памяти относительно таблицы структур.

**Сортировка 480 записей.**

|  |  |  |
| --- | --- | --- |
|  | Сортировка ключей(такты) | Сортировка таблицы(такты) |
| Пузырьком | 220796 | 134780 |
| Быстрая | 117645 | 431881 |

Быстрая сортировка быстрее сортировки пузырьком на 87 % и в случае сортировки таблицы, и в случае сортировки ключей.

Сортировка ключей быстрее сортировки таблицы на 65%, при этом матрица ключей занимает лишь 1.9 % памяти относительно таблицы структур.

*Вывод:* Использование вариантной записи в структуре помогло сэкономить 12 байт, увеличив ее эффективность по памяти. А увеличив использование памяти всего на 1.9%, мы добились уменьшения времени сортировки на 65%. Изменив способ сортировки таблицы на более эффективный время было снижено на 77%. А сортировка массива ключей более эффективным способом дает уменьшение времени на 91.5% по сравнению с сортировкой таблицы исходным способом. Таким образом, для большого объема данных рациональнее по памяти и по времени использовать таблицу ключей, сортируемую эффективным способом. При увеличении кол-ва записей таблицы использование матрицы ключей становится более выгодным.

*Ответы на вопросы:*

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

Память под вариантную часть записи выделяется по наибольшему полю.

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

Тип данных в вариантной части при компиляции не проверяется, поэтому, контроль за правильностью ее использования возлагается на программиста.

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

За правильностью выполнения любых операций с вариантной частью следит программист.

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

При больших размерах таблиц поиск данных, имеющих указанный ключ, может потребовать больших затрат времени. Если же помимо поиска требуется произвести сортировку данных, то временные затраты многократно возрастут, так как потребуется осуществлять их перестановку. В этом случае можно уменьшить время обработки за счет создания дополнительного массива – таблицы ключей, содержащей индекс элемента в исходной таблице и выбранный ключ

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

При малом объёме, занимаемом таблицей, таблицы эффективнее обрабатывать данные в ней, при большом объёме памяти, занимаемом таблицей, эффективнее использовать таблицу ключей.

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

При сортировке таблиц предпочтительнее использовать сортировки с наименьшим количеством перестановок. При сортировке таблицы ключей предпочтительнее использовать сортировки с наименьшей сложностью работы.