Лабораторная работа №2 по дисциплине

“Типы и структуры данных”

Сорокин Антон ИУ7-32Б

Номер по списку - 24

***Условие задачи***

Создать таблицу, содержащую не менее 40 записей. 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 – пьеса, драма, комедия); музыкальный – композитор, страна, минимальный возраст, продолжительность). Вывести список всех музыкальных спектаклей для детей указанного возраста с продолжительностью меньше указанной.

***Техническое задание***

*Исходные данные*

На вход программе может быть подан файл с исходными данными (записями) или же не подаваться никакой информации, принимая исходную таблицу за пустую.

Формат записи в таблице в файле:

Каждое поле записи разделено символом « **|** ». К примеру, порядок невариативных полей следующий:

<театр>|<постановка>|<режиссёр>|<мин. цена билета>|<макс. цена билета>|<тип>|

Под типом понимается то, для кого предназначена постановка: C – для детей, A – для взрослых. Цены на билеты не могут быть отрицательными, но могут быть нулями, а также могут быть равны между собой. Театр, постановка, режиссёр – строки размером не больше 30.

Если постановка предназначена для взрослых (A), то добавляется ещё поле жанра постановки (P – пьеса, C – комедия, D – драма).

Если постановка предназначена для детей (C), то добавляются ещё два поля: допустимый возраст и жанр (F – сказка, P – пьеса, M - музыкальный). Возраст может принимать значения от 0 до 17 включительно.

Если жанр детской постановки музыкальный, то добавляются ешё четыре поля: имя композитора, страна, возраст и продолжительность. Композитор ограничен 30 символами, страна – 20 символами.

*Результат*

Файл с записями в формате, описанном выше.

*Описание задачи*

Программа выполняет обработку таблицы записей постановок (репертуары театров), которая включает в себя добавление и удаление записей, поиск музыкальных спектаклей для детей указанных возраста и продолжительности, а также сортировка таблицы путём сортировки таблицы ключей и самой таблицы (поле – максимальная цена).

*Аварийное завершение работы программы*

Программа не может завершится аварийно, так как реализована как бесконечный вызов меню после выполнения его команд. При этом может аварийно завершится выполнение команд меню при:

* ошибке открытия файла с исходными данными;
* ошибке выделения памяти под таблицу;
* некорректно введённых данных полей при выполнении добавления записи;
* введении недопустимого индекса при выполнении удаления записи;
* ошибке выделения памяти под таблицу ключей

Также все команды, кроме добавления записи, при пустой таблице не выполняются, при выводится сообщение о пустой таблице.

***Обращение к программе***

Исполняемый файл app.exe создается путем автоматической сборки проекта с помощью файла makefile. Для выполнения работы следует запустить данный исполняемый файл без каких-либо аргументов.

***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
  + поля заполняются в соответствии с находимыми в файле символами вертикальной черты. К примеру, читается название постановки. Как только будет найден символ |, чтение в строку постановки остановится и начнётся запись в строку режиссёра. В случае несоответствию формату строки записи или некорректных данных в самом файле возвращается код ошибки.
  + при нахождении символа перехода на новую строку проверить, выделена ли память под следующие записи. Если нет, выделить дополнительно память под 40 записей. Также увеличить счётчик количества записей.
* Добавление записи
  + проверить, выделена ли память под добавляемую запись. Если нет, выделить дополнительно память под 40 записей;
  + последовательно ввести значения всех полей добавляемой записи, в случае некорректных данных – вернуть код ошибки
  + 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;
  + Есть количество записей кратно 40 (не 0), то перевыделить память под текущее количество записей.
* Поиск музыкальных детских постановок указанного возраста и продолжительности
  + Ввести минимальный допустимый возраст и максимальную продолжительность, в случае некорректных данных вернуть ошибку
  + Последовательно в таблице сравнивать полученные значения со значениями полей записей
  + В случае соответствия вывести запись
  + Если не было найдено соответствующих записей, вывести сообщение
  + Вернуть количество найденных записей
* 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

***Тестовые данные***

*Чтение*

|  |  |  |
| --- | --- | --- |
| Ввод | Результат | Класс |
|  |  |  |

*Добавление*

|  |  |  |
| --- | --- | --- |
| Ввод | Результат | Класс |
|  |  |  |

*Удаление*

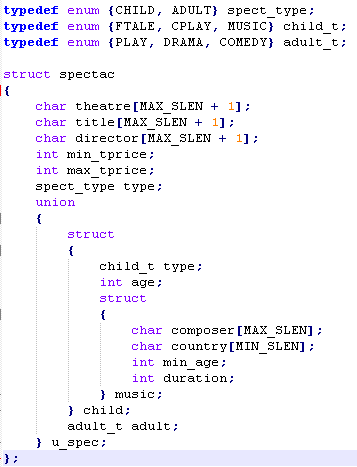
|  |  |  |
| --- | --- | --- |
| Ввод | Результат | Класс |
|  |  |  |

*Поиск*

|  |  |  |
| --- | --- | --- |
| Ввод | Результат | Класс |
|  |  |  |

***Внутренние структуры данных***

Для хранения таблицы записей используется динамический массив данных типа структура с объединением. Количество записей хранится в отдельной целочисленной переменной.

theatre - название театра

title – название постановки

director – имя режиссёра

min\_tprice – мин. цена билета

max\_tprice – макс. цена билета

type – тип постановки (детский/взрослый)

**Объединение u\_spec:**

*при взрослом:*

adult – жанр

*при детском:*

type – жанр

age – допустимый возраст

*при детском музыкальном:*

composer – имя композитора

country – страна

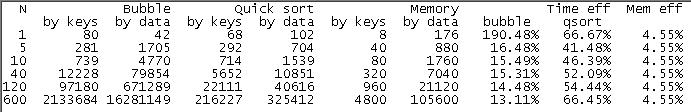
min\_age – минимальный возраст

duration – продолжительность в минутах

Таблица ключей представляет собой массив структур с целыми полями индекса и значения ключа.

Для чтения команд используется символьная переменная.

***Оценка эффективности методов сортировки***



Исходя из полученных данных тестирования методов по памяти и по времени, можно сделать вывод о том, что быстрая сортировка эффективнее, чем сортировка «пузырьком», при количестве элементов, больше 10.

Сортировка через таблицу ключей эффективнее сортировки всей таблицы на всех значениях n, кроме n = 1 при сортировки пузырьком. Эта эффективность составляет падает от 16% при увеличении количества элементов при сортировке «пузырьком», и остаётся в пределах от 40% до 65% при быстрой сортировке.

Эффективность использования ключей по памяти всегда постоянно и составляет 4.55%, так как размер памяти и под таблицу ключей, и под всю таблицу записей зависят прямо пропорционально от количества элементов.

***Вывод***

При сортировки таблицы большого количества записей по одному полю эффективнее сортировать таблицу ключей по значениям данного поля методом быстрой сортировки, при малом количестве записей – также таблицу ключей, но методом сортировки «пузырьком».

***Ответы на вопросы***

1. **Как выделяется память под вариативную часть?**

Под вариативную часть выделяется память размером, равным максимальному размеру среди вариативных полей.

1. **Что будет, если в вариантную часть ввести данные, несоответствующие**

**описанным?**

При компиляции не проверяется тип вводимых данных вариативной части, поэтому возможна перезапись другого поля.

1. **Кто должен следить за правильностью выполнения операций с вариантной**

**частью записи?**

Программист

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

Таблица ключей представляет собой структуру из двух полей: индекса элемента в таблице записей и значение ключа данной записи. Таблица ключей позволяет ускорить процессы сортировки, т. к. обрабатывается в этом случае меньший объём памяти.

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

Данные в самом таблице эффективнее обрабатывать, если при обработке производится только один проход по таблице (поиск). В этом случае создание таблицы ключей не является эффективным. Таблицу ключей эффективнее обрабатывать, если при обработке может потребоваться не один проход по таблице (сортировка).

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

В общем случае (кроме n = 1) эффективнее для сортировки таблицы использовать таблицу ключей. При малом количестве записей (меньшем 10) предпочтительнее использовать простые алгоритмы сортировки, например, «пузырьком», вставками, выбором; в остальных случаях – алгоритм быстрой сортировки.