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.

*Аварийное завершение работы программы*

Программа не может заврешится 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.
* Провести дополнительную операцию деления для одного разряда, уходящего за допустимую сетку. Результат сохранить отдельно как дополнительную цифру.
* Выполнить округление результата с учётом дополнительной цифры.
* Вычислить результирующий порядок числа с учётом порядка исходного числа и добавки.
  + в случае переполнения порядка, вывести ошибку
* Определить, с какой позиции в числе начинаются незначащие нули.
* Вывести цифры числа до этой позиции.

***Тестовые данные***

|  |  |  |
| --- | --- | --- |
| Ввод | Вывод | Класс эквивалентности |
|  | Empty string was entered! | Пустая строка |
| + | Incorrect number format! | Нет мантиссы |
| E +11 | Incorrect number format! | Нет мантиссы |
| +55 | Incorrect number format! | Нет порядка |
| +55E11 | Incorrect number format! | Неверный формат порядка |
| +1234567890123456789012345678901 E 11 | Incorrect number format! | Переполнение мантиссы |
| +1 E +100001 | Incorrect number format! | Переполнение порядка |
| -55 E | Incorrect number format! | Нет порядка |
| +-1 E +15 | Incorrect digit! | Лишние знаки в мантиссе |
| +1 E -+15 | Incorrect digit! | Лишние знаки в экспоненте |
| +1 E +15 | Incorrect number format! | Лишние пробелы |
| +1 E E +15 | Incorrect digit! | Лишний знак экспоненты |
| +1 E +12.5 | Incorrect digit! | Вещественный порядок |
| +1 E +a | Incorrect digit! | Символьный порядок |
| +1343a12 E +15 | Incorrect digit! | Недопустимый символ в мантиссе |
| +1 E +99999 | Exponent overflow! | Переполнение порядка исходного числа |
| +0.01 E -99999 | Exponent overflow! | Переполнение порядка исходного числа |
| 0.1 E 0  (пустая строка) | Empty string was entered! | Пустая строка |
| 0.1 E 0  + | Incorrect number format! | Нет цифр числа |
| 0.1 E 0  +1234567890123456789012345678901 | Incorrect number format! | Введено больше 30 разрядов |
| 0.1 E 0  1a2312 | Incorrect digit! | Недопустимые символы в строке |
| 0.1 E 0  1.2312 | Incorrect digit! | Введена десятичная точка |
| 0.1 E 0  0 | Division by zero! | Деление числа на 0 |
| 0 E 0  0 | Division by zero! | Деление 0 на 0 |
| 0.0 E 55  1 | +0.0 E -00000 | Деление 0 на число, отличное от 0 |
| +0.1 E -99999  10 | Exponent overflow! | Переполнение после деления |
| +123.45 E +01  1 | +0.12345 E +00004 | Деление на 1 |
| +123.45 E +01  -1 | -0.12345 E +00004 | Деление на -1 |
| +0.25 E +01  100 | +0.25 E -00001 | Порядок меняет знак  c + на - |
| +0.25 E +01  10 | +0.25 E -00000 | Порядок становится нулевым |
| +2 E +00  .3 | +0.666666666666666666666666666667 E -00000 | После деления необходимо округление (доп. цифра > 4) |
| 12 E +00  .3 | +0.333333333333333333333333333333 E -00000 | После деления округление не нужно (доп. цифра <= 4) |
| +0.12 E -001  12 | +0.1 E -00002 | Общий случай |

***Внутренние структуры данных***

Для хранения больших чисел используются массивы. Для представления вещественного числа используются два массива: отдельно для мантиссы и отдельно для порядка.

Массив целого числа или мантиссы:

* Кол-во элементов – 31
* Первые 30 элементов – цифры целого числа
* 31ый элемент – знак числа (коды символов ‘+’ или ‘-‘)

Массив порядка:

* Кол-во элементов – 7
* Первые 5 элементов – цифры порядка
* 6ой элемент – знак пордяка (коды символов ‘+’ или ‘-‘)
* 7ой элемент – позиция десятичной точки в мантиссе (используется при чтении)

Результат представляет из себя два массива: массив мантиссы и вещественного числа

.

***Выводы***

По выполнении данной лабораторной работы, можно сделать вывод о том, что для выполнения длинной арифметики следует использовать массивы для записи цифр числа, а для выполнения операций использовать поразрядные методы вычисления («столбиком» и «уголком»).

***Ответы на вопросы***

* Каков возможный диапазон чисел, представляемых в ПК?

Возможный диапазон чисел, представляемых в ПК, зависит от архитектуры самой машины. Так, для 32-разрядной машины диапазоном целых чисел является [-232 ; 232 – 1], а для 64-разрядной машины - [-264 ; 264 – 1].

Для вещественных чисел максимально возможное число разрядов составляет 52 под хранение мантиссы и 11 под разряд. Соответственно диапазон следующий: [3.6 E –4951; 1.1 E +4932]

* Какова возможная точность представления чисел, чем она определяется?

Как правило, до 20 разрядов мантиссы. Определяется архитектурой машины

* Какие стандартные операции возможны над числами?

Арифметические: сложение, вычитание, умножение, деление

Операции сравнения

Логические

* Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?

Массив целых или символов для записи цифр числа

* Как можно осуществить операции над числами, выходящими за рамки машинного представления?

Следует выполнять поразрядные операции сложения и вычитания, которые применяются в алгоритмах сложения, вычитания, умножения и деления.