|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования 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)** |

**Отчет**

**по лабораторной работе №2**

**по теме**

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

**Вариант 5.**

Дисциплина: Типы и структуры данных

Студент ИУ7-31Б:

Косарев Алексей

Проверил:

Т.А.Никульшина

Москва, 2021

1. **Описание условия задачи**

Ввести репертуар театров, содержащий: название театра, спектакль, режиссер, диапазон цены билета, тип спектакля: детский – для какого возраста, тип (сказка, пьеса); взрослый – пьеса, драма, комедия); музыкальный – композитор, страна, минимальный возраст, продолжительность). Вывести список всех музыкальных спектаклей для детей указанного возраста с продолжительностью меньше указанной.

1. **Описание ТЗ**

1. Описание исходных данных

Исходными данными является структурированная информация о театральном спектакле.

Запись содержит:

1. Название театра

2. Название спектакля

3. Диапазон цен (начальная цена и конечная)

4. Тип спектакля (детский, взрослый, мюзикл)

5. Минимальный возраст (для типа “детский” и “взрослый”)

6. Тип спектакля (для типа “детский”: сказка, пьеса; для типа “взрослый”: пьеса, драма, комедия)

7. Имя композитора (для типа “мюзикл”)

8. Название страны (для типа “мюзикл”)

9. Длительность (для типа “мюзикл”)

Поля:

- Максимальная длина названия театра – 50

- Максимальная длина названия спектакля – 50

- Диапазон цен вводится как значения двух чисел (начальная цена и конечная)

- Тип вводится следующим образом:

1 - детский

2 - взрослый

3 - мюзикл

- Минимальный возраст вводится в произвольном порядке (не может быть отрицательным)

- Дополнительный тип вводится следующим образом

для типа “детский”:

1 - сказка

2 - пьеса

для типа “взрослый”:

1 - пьеса

2 - драма

3 - комедия

- Максимальная длина имени композитора – 25

- Максимальная длина названия страны – 50

- Длительность записывается целым числом

2. Описание результатов программы

Меню:

1. Загрузка структур из файла

2. Вывод структур на экран

3. Сортировка изначального массива структур

4. Сортировка массива структур с помощью массива ключей

5. Вывод массива ключей

6. Поиск мюзикла для детей с указанным возрастом и указанной длительностью

7. Добавление структуры в массив

8. Удаление структуры из массива по значению поля

9. Аналитика времени и памяти

0. Выход из программы

1. **Описание задачи, реализуемой в программе**

Программа реализует обработку массива структур, используя таблицу.

1. **Способ обращения к программе**

Обращение к программе происходит через консоль, путём запуска файла с расширением .exe

(./main.exe) и ввода параметров командной строки (название входного текстового файла)

1. **Описание возможных аварийных ситуаций и ошибок пользователя**

#define INCORRECT\_ARGS 1 // Некорректные аргументы командной строки

#define FILE\_OPEN\_ERR 2 // Ошибка открытия файла

#define FILE\_ERR 3 // Некорректный файл

#define FILE\_CLOSE\_ERR 4 // Ошибка закрытия файла

#define EMPTY\_FILE 5 // Пустой файл

#define INCORRECT\_INPUT 6 // Некорректный ввод

#define INCORRECT\_VALUE 7 // Некорректное значение

#define EMPTY\_STRING 8 // Пустая строка

#define THEATRE\_NAME\_OVERFLOW 9 // Переполнение названия театра

#define PLAY\_NAME\_OVERFLOW 10 // Переполнение названия спектакля

#define COMPOSER\_NAME\_OVERFLOW 11 // Переполнение имени композитора

#define COUNTRY\_NAME\_OVERFLOW 12 // Переполнение названия страны

#define THEATRE\_ARRAY\_OVERFLOW 13 // Переполнение массива структур

1. **Описание внутренних структур данных**

typedef struct

{

int min\_age; // Возрастное ограничение

int type\_of\_child\_play; // Тип: 1 - сказка, 2 - пьеса

} child\_play;

typedef struct

{

int type\_of\_adult\_play; // Тип: 1 - пьеса, 2 - драма, 3 - комедия

} adult\_play;

typedef struct

{

char composer\_name[MAX\_COMPOSER\_NAME + 2]; // Композитор

char country\_name[MAX\_COUNTRY\_NAME + 2]; // Страна

int min\_age; // Возрастное ограничение

int play\_duration; // Длительность мюзикла

} musical;

struct theatre

{

char theatre\_name[MAX\_THEATRE\_NAME + 2]; // Название театра

char play\_name[MAX\_PLAY\_NAME + 2]; // Название спектакля

int start\_price; // Начальная цена билета

int end\_price; // Конечная цена билета

int type\_of\_play; // Тип спектакля: 1 - детский, 2 - взрослый, 3 - мюзикл

union

{

child\_play child\_play; // Детский спектакль

adult\_play adult\_play; // Взрослый спектакль

musical musical; // Мюзикл

} type;

};

struct key\_theatre{

char play\_name[MAX\_THEATRE\_NAME + 2]; // Название спектакля

int array\_index; // Индекс в начальном массиве

};

1. **Тесты**

Положительные тесты:

1. Сортировка массива структур по полю “Название театра”
2. Сортировка массива структур по полю “Название спектакля”
3. Сортировка массива структур по диапазону цен
4. Сортировка массива структур по полю “Название спектакля” с помощью массива ключей
5. Поиск мюзикла с существующими подходящими данными
6. Удаление структуры по значению поля “Название театра”
7. Удаление структуры по значению поля “Название спектакля”
8. Удаление структуры по значению диапазона цен

Негативные тесты:

1. Некорректный ввод параметров командной строки
2. Пустая строка на месте любого поля
3. Отрицательное значение цен
4. Отрицательное значение минимального возраста
5. Отрицательное значение длительности
6. Длина названия театра > 50 символов
7. Длина названия спектакля > 50 символов
8. Длина имени композитора > 25 символов
9. Длина названия страны > 25 символов
10. Переполнение массива структур
11. **Временная эффективность и затраты памяти**

Временная эффективность.

Время, мс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | bubble | | qsort | |
| Кол-во  элементов | начальный массив | массив  ключей | начальный массив | массив  ключей |
| 50 | 52 | 32 | 4 | 0 |
| 100 | 132 | 56 | 6 | 0 |
| 150 | 252 | 136 | 8 | 0 |
| 600 | 4001 | 1976 | 28 | 12 |
| 1000 | 11068 | 5572 | 36 | 24 |

Можно увидеть, что время сортировки пузырьком начальной таблицы медленнее на 75-90%, чем сортировка с помощью массива ключей, а время сортировки qsort начальной таблицы больше на 90-100%, чем с помощью массива ключей.

Затраты по памяти.

При обработке массива структур размерности 148 используется 26640 байт памяти. А при использовании массива ключей той же размерности используется 8288 байт памяти.

На массив ключей вместе с массивом структур требуется на 31% больше памяти, чем просто на массив структур.

1. **Вывод**

В процессе выполнения данной лабораторной работы я изучил принципы реализации записей с вариантами и способы обработки таблиц.

Увеличив затраты по памяти (дополнительный массив ключей) примерно на 31%, мы смогли уменьшить время сортировки таблицы примерно на 50%.

1. **Ответы на контрольные вопросы**
2. Как выделяется память под вариативную часть записи?

Объем памяти, необходимый для записи с вариантами складывается из длин полей фиксированной части и максимального по длине поля вариантной части.

Union test{

int a;

int b;

double c;

}

Под данное объединение будет выделено 8 байт (размер double).

1. Что будет, если в вариативную часть ввести данные, несоответствующие с описанными?

Они будут отображаться некорректно.

1. Кто должен следить за правильностью выполнения операций, с вариативной частью записи?

За правильностью выполнения операций над вариативной частью должен следить программист.

1. Что представляет из себя таблица ключей и зачем она нужна?

Таблица ключей представляет из себя структуры, либо массив, каждый элемент которого содержит значение какого-либо ключа и позицию из исходной таблицы.

Таблица ключей нужна повышения эффективности работы с таблицей.

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

В случае, если нам часто приходится сортировать таблицу с большим количеством полей, следует использовать таблицу ключей. Если же полей немного, то можно обойтись и без выделения дополнительной памяти для таблицы ключей.

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

Предпочтительнее те способы сортировки, которые не переставляют уже отсортированные элементы, например, сортировка пузырьком с флагом или метод сортировки вставками.