Склифасовский Денис ИУ7-35

Вариант 20

Лабораторная работа №1

Записи с вариантами, обработка таблиц.

**Цель работы** - приобрести навыки работы с типом данных «запись» («структура»,) содержащим вариантную часть, и с данными, хранящимися в таблицах. Оценить относительную эффективность программы (в процентах) по времени и по используемому объему памяти в зависимости от используемого алгоритма и от объема сортируемой информации.

**Условие задачи (вар 20):**

Ввести список стран, содержащий название страны, количество жителей, столицу, материк, основной вид туризма (экскурсионный - количество объектов, основной вид (природа, история, искусство); пляжный – основной сезон, температура воздуха и воды, время полета до страны; спортивный – вид спорта (горные лыжи, серфинг, восхождения), минимальная стоимость отдыха). Вывести список стран на выбранном материке, где можно заняться указанным видом отдыха.

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)

**Описание функциональности ПО.**

ПО должно уметь

1) Выводить таблицу

2) Добавлять запись в таблицу

3) Удалять запись из таблицы

4) Выводить список стран на выбранном материке, где можно заняться указанным видом спорта

5) Сортировка таблицы

6) Создание таблицы ключей

7) Вывод таблицы ключей

8) Сортировка таблицы ключей

9) Сохранение измененного файла

10) Проверка сортировок

11) Вывод таблицы по ключам

12) Загрузка файла

**Требования к программе**.

Требования к информационным структурам.

1) Длина названия страны не больше 20

2) Длина названия континента не больше 20

3) Длина названия столицы не больше 20

4) Максимальная популяция 2 147 483 64

5) Вид туризма должен описываться словами Tourism, Beach, Sport

6) Для видов туризма должны быть созданы вариативные поля

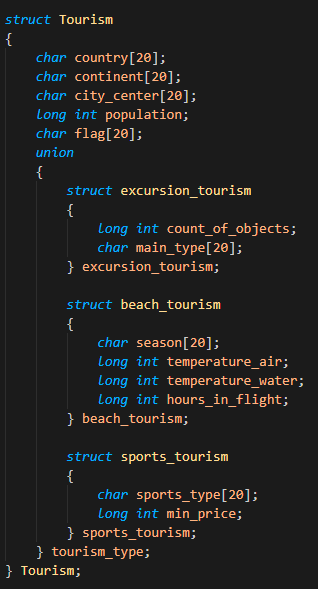
**Требования к надёжности.**

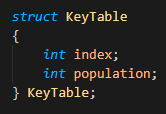
Типы отказов при работе ПО.

1. Ошибки ввода/вывода
2. Превышение допустимого количества записей
3. Неправильный ввод чисел
4. Попытка вывода таблицы по ключам без создания этой таблицы

**Приложение к техническому заданию.**

**Структуры**





**Функции**

*long int read\_file(char \*filename, struct Tourism \*new, long int count);*

Считывание файла

*void display\_table(struct Tourism \*tourists, long int count);*

Вывод таблицы

*void add\_record(long int \*count, struct Tourism \*new);*

Добавление записи

*void delete\_record(struct Tourism \*tourists, long int \*count);*

Удаление записи

*void sort\_tourism\_table(struct Tourism \*tourists, long int count);*

Сортировка таблицы

*void create\_key\_table(struct Tourism \*tourists, struct KeyTable \*keys, long int count);*

Создание таблицы ключей

*void show\_key\_table(struct KeyTable \*keys, long int count);*

Вывод таблицы ключей

*void sort\_key\_table(struct KeyTable \*keys, long int count);*

Сортировка таблицы ключей

*void display\_countries(struct Tourism \*tourists, int count);*

Вывод страны по континенту и виду спорта

*void save\_file(struct Tourism \*tourists, int count);*

Сохранение файла

*void check\_soutring(struct Tourism \*tourists, struct KeyTable \*keys, long int count);*

Проверка сортировок

*void sort\_table\_buble(struct Tourism \*keys, long int count);*

Сортировка таблицы пузырьком

*void sort\_keys\_qsort(struct KeyTable \*keys, long int count);*

Сортировка таблицы методом qsort

*int comp\_key(const void \*a, const void \*b);*

Сравнение для qsort

*void display\_table\_by\_keys(struct Tourism \*tourists, struct KeyTable \*keys, long int count);*

Вывод таблицы по ключам

*int check\_keys\_dimention(struct KeyTable \*keys, long int count);*

Проверка размера таблицы ключей

*int check\_table\_dimention(struct Tourism \*tourists, long int count);*

Проверка размера таблицы

**Тесты**

1. Несуществующий файл ключей

|  |  |
| --- | --- |
| Ввод | Вывод |
| 7 (вывод таблицы ключей) | You should create key table! |
| 8 (сортировка таблицы ключей) | You should create key table! |
| 10 (вывод таблицы по ключам) | You should create key table! |
| 11 (проверка времени сортировок) | You should create key table! |

1. Добавление записи

|  |  |
| --- | --- |
| Ввод | Вывод |
| 2 Russia Europe Moscow 156 Sport Basketball 10000 | Корректный ввод |
| 2 Russia Europe Moscow Six | Invalid input |

1. Удаление записи

|  |  |
| --- | --- |
| Ввод | Вывод |
| 3 Russia Europe Moscow 156 Sport Basketball 10000 | Done! |
| Russia Europe Moscow 156 Sport Basketball 500 | We couldn't find anything |

1. Вывод стран по континентам и спорту

|  |  |
| --- | --- |
| Ввод | Вывод |
| 4 Asia Volleyball | * Russia |
| 4 Asia Swimming | We couldn't find any countries |

**Результаты работы программы**

После выполнения данной работы, стало понятно, что для точного времени сортировки таблиц нужно использовать больше данных. (В моем случае при 156 полях считалось точнее, чем при 39) Так же получилось, что память, выделенная под массив структур больше в 14,5 раз, чем массив ключей.

Result of sorting table with qsort is > 6

Result of sorting table with buble is > 84

Result of sorting key table with qsort is > 4

Result of sorting key table with buble is > 74

Procenting dimention 14.5

Dimention of structs > 18096 1248

**Вывод**

Сортировка с использованием ключей оказалась эффективнее. Объем памяти под структуру оказался 116 б. Объем памяти под структуру ключй оказался 8 б. Объем занимаемой дополнительной таблицей ключей относительно исходной (116 + 8) / 116 = 1.0689 (Увеличен на 6.89 %)

Вывод: Для сортировки больших баз данных выгоднее использовать дополнительную таблицу ключей. При использовании дополнительной 6.89 % от исходной таблицы происходит выигрыш во времени более чем в два раза.

Контрольные вопросы.

1. Как выделяется память под вариантную часть записи?

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

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?

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

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

Таблицы ключей содержат индекс элемента в исходной таблице и выбранный ключ. Сортировка таблицы ключей экономит время, поскольку перестановка записей в исходной таблице отсутствует. При подобной организации вычислительного процесса разделяются понятия хранения, структурирования, и обработки данных.

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

Для размещения таблицы ключей требуется дополнительная память. Вывод данных из основной таблицы в порядке, определенном таблицей ключей, замедляет вывод этих данных. Следовательно, если объём памяти критичен, исходная таблица состоит из небольшого количества полей, намного чаще происходит запрос на чтение данных, чем на создание/обновление/удаление, таблица ключей окажется не эффективной. Но таблица ключей ускоряет поиск записей по id, так как становиться возможен бинарный поиск.