|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**«ЗАПИСИ С ВАРИАНТАМИ, ОБРАБОТКА ТАБЛИЦ»**

Студент Лазутин Александр Владимирович

Группа ИУ7 – 33Б

Проверяющий Всеволод Алексей Иванов

*2022г.*

Оглавление

[Описание условия задачи 3](#_Toc116071830)

[Описание технического задания 3](#_Toc116071831)

[Входные данные: 3](#_Toc116071832)

[1.Файл с данными: 3](#_Toc116071833)

[2. Целое число, представляющее собой номер команды: ​ 4](#_Toc116071834)

[3. Дополнительный ввод: 4](#_Toc116071835)

[Выходные данные: 4](#_Toc116071836)

[Функции программы: 4](#_Toc116071837)

[Обращение к программе: 4](#_Toc116071838)

[Аварийные ситуации: 5](#_Toc116071839)

[Описание структуры данных 6](#_Toc116071840)

[Описание алгоритма 8](#_Toc116071841)

[Набор тестов 8](#_Toc116071842)

[Оценка эффективности 11](#_Toc116071843)

[Ответы на контрольные вопросы 13](#_Toc116071844)

[Вывод 14](#_Toc116071845)

# Описание условия задачи

Создать таблицу, содержащую не менее 40-ка записей (тип – запись с

вариантами (объединениями)). Упорядочить данные в ней по

возрастанию ключей, двумя алгоритмами сортировки, где ключ – любое

невариантное поле (по выбору программиста), используя: а) саму

таблицу, б) массив ключей. (Возможность добавления и удаления

записей в ручном режиме обязательна). Осуществить поиск

информации по варианту.

Вариант 14:

Ввести список стран, в которых можно отдохнуть, содержащий название страны, количество жителей, столицу, материк, нужна ли прививка или ПЦР, 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 спорта.

# Описание технического задания

## Входные данные:

### 1.Файл с данными:

Текстовый файл формата txt. Разделителем в файле является пробел.

Каждая новая запись таблицы в обязательном порядке должна находиться на новой строке файла.

Запись содержит :

* Страна – до 50 символов
* Кол-во людей
* Столица – до 50 символов
* Материк – до 50 символов
* ПЦР
* Вид туризма:
  + Экскурсионный (количество объектов, основной вид)
  + Пляжный (основной сезон, температура воздуха и воды, время полета до страны)
  + Спортивный (вид спорта, минимальная стоимость отдыха)

### 2. Целое число, представляющее собой номер команды: ​

Число типа int в диапазоне от 0 до 10

### 3. Дополнительный ввод:

Поле типа int или char в зависимости от требования

## Выходные данные:

Текущее состояние таблицы, результаты сравнения эффективности сортировок, результаты поиска по заданным полям, информация о выполнение определенной задачи.

## Функции программы:

1. Загрузить список стран из файла –

int read\_file(FILE \*file\_in, country\_info\_t \*array, size\_t \*size);

1. Добавить страну в список –

int add\_country(country\_info\_t \*array, int \*size);

1. Удалить информацию о стране по самой стране –

short int delete\_by\_country(country\_info\_t \*array, int \*size);

1. Удалить информацию о стране по ее номеру в списке –

void del\_by\_pos(country\_info\_t \*array, int size, int flat);

1. Найти все страны по указанному материку и виду спорта –

int search(country\_info\_t \*array, int size);

1. Отсортировать таблицу пузырьком по странам –

int bubble\_sort\_country(country\_info\_t \*array, int size);

1. Отсортировать таблицу ключей пузырьком по странам –

int bubble\_sort\_keys(keys\_t \*key, int size);

1. Отсортировать таблицу qsort по типу путешествия –

void qsort\_table(country\_info\_t \*array, int size);

1. Отсортировать таблицу ключей qsort по странам –

void qsort\_table\_keys(keys\_t \*key, int size);

1. Вывести сравнение сортировок пузырьком и qsort –

short int compare\_sorts(FILE \*file);

1. Вывести таблицу –

void print\_string(country\_info\_t array, int index);

1. Вывести таблицу ключей –

void print\_table\_key();

1. Вывести отсортированную таблицу по таблице ключей

void print\_string(country\_info\_t array, int index)

1. Очистить таблицу
2. Выйти из программы

## Обращение к программе:

Запускается через терминал коммандой ./app.exe

## Аварийные ситуации:

1. Неверно введён пункт меню

Входные данные: не число или число, большее 13 или меньшее 0

Выходные данные: сообщение «Ошибка: пункт меню введён неверно»

2. Неверно введено имя файла

Входные данные: строка с несуществующим файлом

Выходные данные: сообщение «Ошибка: неверно указано имя файла»

3. В записи находятся недопустимые данные

Входные данные: файл с неверным данными

Выходные данные: сообщение «Ошибка: неверные записи в файле»

4. Количество возможных записей переполнено

Входные данные: файл, в котором больше записей, чем допустимо

Выходные данные: сообщение «Ошибка: количество записей переполнено»

5. Неверно введёна страна

Входные данные: строка выше 50 символов ил не строка

Выходные данные: сообщение «Ошибка: неверно введёна страна»

6. Неверно введено кол-во жителей

Входные данные: не число или число, меньшее 1

Выходные данные: сообщение «Ошибка: неверно введено кол-во людей»

7. Неверно введёна столица

Входные данные: строка выше 50 символов ил не строка

Выходные данные: сообщение «Ошибка: неверно введёна столица»

8. Неверно введён материк

Входные данные: строка выше 50 символов ил не строка

Выходные данные: сообщение «Ошибка: неверно введён материк»

9. Неверно введен ПЦР

Входные данные: не число или число, меньшее 0 или большее 1

Выходные данные: сообщение «Ошибка: неверно введен ПЦР»

10. Неверно введен вид туризма

Входные данные: не число или число, меньшее 1 или большее 3

Выходные данные: сообщение «Ошибка: неверно введен вид туризма»

11. Неверно введено кол-во исторических объектов

Входные данные: не число

Выходные данные: сообщение «Ошибка: неверно введено кол-во исторических объектов»

12. Неверно введен вид

Входные данные: строка больше 50 символов

Выходные данные: сообщение «Ошибка: неверно введен вид»

13. Неверно введен сезон

Входные данные: строка больше 50 символов

Выходные данные: сообщение «Ошибка: неверно введен сезон»

14. Неверно введена температура воздуха

Входные данные: не число

Выходные данные: сообщение «Ошибка: неверно введена температура воздуха»

15. Неверно введена температура воды

Входные данные: не число

Выходные данные: сообщение «Ошибка: неверно введена температура воды»

16. Неверно введено время перелета

Входные данные: не число или число, меньше 1

Выходные данные: сообщение «Ошибка: неверно введено время перелета»

17. Неверно введен спорт

Входные данные: не строка

Выходные данные: сообщение «Ошибка: неверно введен спорт»

18. Неверно введена минимальная стоимость

Входные данные: не число или число, меньше 0

Выходные данные: сообщение «Ошибка: неверно введена минимальная стоимость»

19. Файл пустой

Входные данные: пустой файл

Выходные данные: сообщение «Ошибка: файл пуст»

# Описание структуры данных

typedef struct

{

int number\_objects;

char main\_view[MAX\_LEN];

} excursion\_t;

Поля структуры excursion\_t:

1. int number\_objects – кол-во исторических объектов

2. char main\_view[MAX\_LEN] – вид

typedef struct

{

char season[MAX\_LEN];

int temperature\_air;

int temperature\_water;

int flight\_time;

} beach\_t;

Поля структуры beach\_t:

1. char season[MAX\_LEN] – сезон
2. int temperature\_air – температура воздуха
3. int temperature\_water – температура воды
4. int flight\_time – время перелета

typedef struct

{

char view\_sports[MAX\_LEN];

int minimum\_cost;

} sports\_t;

Поля структуры sports\_t:

1. char view\_sports[MAX\_LEN] – вид спорта

2. int minimum\_cost – минимальная стоимость

Тип данных country\_info\_t представляет собой структуру, содержащую название страны, кол-во людей, столицу, материк, ПЦР, вид туризма. Так же эта структура объединяется с одной из 3 других, в зависимости от вида туризма.

typedef struct

{

char country[MAX\_LEN];

int count\_people;

char capital[MAX\_LEN];

char mainland[MAX\_LEN];

int PCR;

int tourism;

union

{

excursion\_t excursion\_holiday;

beach\_t beach\_holiday;

sports\_t sports\_holiday;

} holiday\_kind\_t;

} country\_info\_t;

keys\_t — структура, которая хранит в себе ключи для поля «страна»

основной таблицы

typedef struct keys

{

char country[50];

int id;

} keys\_t;

Поля структуры:

1. char country[50] — площадь квартиры

2. int id - номер квартиры в основной таблице

# Описание алгоритма

1. Выводится меню программы.
2. Пользователь вводит пункт меню, который отвечает за какое-то из действий.
3. Ввод осуществляется до того момента, пока не будет введён 0, который является выходом из программы, или пользователь некорректно введет данные.

# Набор тестов

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название теста** | **Пользовательский ввод** | **Результат** |
| 1 | Некорректный ввод пункта меню | i | Ошибка: пункт меню введён неверно |
| 2 | Некорректный ввод пункта меню | 15 | Ошибка: пункт меню введён неверно |
| 3 | Неверно введено имя файла | no.txt | Ошибка: неверно введено имя файла |
| 4 | В файле находятся недопустимые данные | bad.txt | Ошибка: неверные записи в файле |
| 5 | Количество возможных записей переполнено | Файл, хранящий более чем 100 записей | Ошибка: количество записей переполнено |
| 6 | Неверно введёна страна | Строка с более, чем 50 символов | Ошибка: неверно введёна страна |
| 7 | Неверно введено кол-во людей | -1 | Ошибка: неверно введено кол-во людей |
| 8 | Неверно введен материк | 1 | Ошибка: неверно введен материк |
| 9 | Неверно введена столица | 1 | Ошибка: неверно введена столица |
| 10 | Неверно введен ПЦР | 2 | Ошибка: неверно введен ПЦР |
| 11 | Неверно введен вид туризма | 0 | Ошибка: неверно введен вид туризма |
| 12 | Неверно введено количество исторических объектов | ) | Ошибка: неверно введено количество кол-во исторических объектов |
| 13 | Неверно введен вид | 1234412 | Ошибка: неверно введен вид |
| 14 | Неверно введен сезон | 32 | Ошибка: неверно введен сезон |
| 15 | Неверно введена температура воздуха или воды | ыва | Ошибка: неверно введена температура воздуха\воды |
| 16 | Неверно введено время перелета | -123 | Ошибка: неверно введено время перелета |
| 17 | Неверно введен вид спорта | 432 | Ошибка: неверно введен вид спорта |
| 18 | Добавление верной записи в конец файла | Верная запись | Страна была успешно добавлена |
| 19 | Ввод записей из файла с правильными записями | Правильный файл | Список был успешно прочитан |
| 20 | Удаление страны | Название страны, которая есть в таблице | Успешно удалены страны из таблицы |
| 21 | Удаление страны | Название страны, которой нет в таблице | Не найдены указанные страны |
| 22 | Поиск по условию | Название материка и вида спорта  (такие записи есть) | Таблица с подходящими странами |
| 23 | Поиск по условию | Название материка и вида спорта  (такие записи нет) | Ничего не было найдено |
| 24 | Ввод пункта меню с сортировкой | Полная таблица, где есть хотя бы 1 запись | Таблица успешно отсортирована |
| 25 | Вывод любой таблицы | Таблица хотя бы с одной записью | Вывод таблицы с шапкой |
| 26 | Ввод нуля | Пункт меню 0 | Выход из программы, очистка консоли |
| 27 | Пустой файл | Пустой файл | Таблица пуста |

# Оценка эффективности

Время сортировки:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Количество записей | Таблица | | Таблица ключей | |
| Сортировка «bubble», секунды | Сортировка «qsort», секунды | Сортировка «bubble», секунды | Сортировка «qsort», секунды |
| 25 | 0.0000054111 | 0.0000017174 | 0.0000038826 | 0.0000015014 |
| 50 | 0.0000222451 | 0.0000029434 | 0.0000159443 | 0.0000028857 |
| 75 | 0.0000923229 | 0.0000086977 | 0.0000636611 | 0.0000065337 |
| 100 | 0.0001477897 | 0.0000075306 | 0.0000819791 | 0.0000066611 |

Для получение более достоверных значений я сумировал измерения каждой сортировки 20 раз. После чего полученное значение делил на 20, получая усредненное время каждой сортировки.

Объем занимаемой памяти:

|  |  |  |
| --- | --- | --- |
| Количество записей | Объем таблицы,  байты | Объем таблицы ключей,  байты |
| 25 | 5700 | 1400 |
| 50 | 11400 | 2800 |
| 75 | 17100 | 4200 |
| 100 | 22800 | 5600 |

Чтобы найти размеры основной таблицы и таблицы ключей я использовал функцию sizeof().

Таблица соотношений памяти и времени:

|  |  |  |  |
| --- | --- | --- | --- |
| Количество записей | % памяти, занимаемый таблицей ключей от всей таблицы | На сколько % сортировка таблицы ключей быстрее сортировки всей таблицы (“bubble”) | На сколько % сортировка таблицы ключей быстрее сортировки всей таблицы (“qsort”) |
| 25 | ~24% | ~30% | ~13% |
| 50 | ~24% | ~30% | ~2% |
| 75 | ~24% | ~30% | ~25% |
| 100 | ~24% | ~45% | ~12% |

Для получение информации о сравнении основной таблицы и таблицы ключей я принимал время сортировки основной таблицы за 100%, а таблицу ключей за Х, после чего находил, на сколько процентов таблица ключей работает эффективнее.

Получение процента памяти, занимаемого таблицей ключей от всей таблицы, я делал аналогично нахождению процента эффективности для времени.

# Ответы на контрольные вопросы

1.Как выделяется память под вариантную часть записи?

Максимальное по длине поле вариантной части - размер памяти, который выделяется под вариантную часть. Эта память является общей для всех полей вариантной части записи

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

В вариантной части при компиляции тип данных не проверяется. Невозможно считать корректно данные, поэтому поведение будет неопределенным

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

Эта задача лежит на программисте

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

Таблица ключей представляет собой структуры из двух массивов, один под индексы, другой под поля для хранения. Использование таблицы ключей при сортировке, ускоряет ее работу, но требует дополнительной памяти

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

Если исходная таблица содержит небольшое количество полей, то лучше сортировать именно её. Но если в структуре большое количество полей, то стоит создать дополнительную таблицу ключей, которая займет дополнительную память, но ускорит работу сортировки

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

Если сортировка производится по таблице ключей, то эффективнее использовать сортировки с наименьшей сложностью работы

Если будет производится сортировка самой таблицы, то необходимо использовать алгоритмы, требующие наименьшее количество операций перестановки.

# Вывод

В данной лабораторной работе используется два основных понятия

1) Вариантная запись – помогает экономить память, так как выделятся она под самое динное поле из имеющихся, но правильность данных, которые подаются на вход структуре должны быть проверены самим программистом, так как компилятор не сможет отследить ошибку.

2) Использование таблицы ключей помогает оптимизировать сортировку таблицы с большимм количеством полей, что даст хороший прирост во времени, но потребуетсяя пожертвовать дополнительной памятью. Что касаемо таблиц с небольшим количеством полей, то таблица ключей особо не ускорит работу сортировки, но память затратит.