Отчет по 2 лабораторной работе

По дисциплине «Типы и структуры данных»

Подготовил Жабин Дмитрий

Группа ИУ7-34Б

Вариант 7

**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.

**Задача**

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). Найти всех друзей, которых необходимо поздравить с днем рождения в ближайшую неделю.

Сравнить различные алгоритмы сортировки массива при использовании таблиц записей с большим числом полей и таблиц ключей. Оценить эффективность (по времени и по используемому объему памяти) при различной реализации программы, то есть в случаях (1) и (2).

**Внешняя спецификация программы.**

**Входные данные:** текстовый файл, представляющий таблицу записей (не более 100) с вариантной частью.

Таблица - список абонентов, содержащий фамилию, имя, телефон, адрес, статус (личный – дата рождения: день, месяц, год; служебный – должность, организация).

Фамилия - строка (не более 30 символов)

Имя - строка (не более 20 символов)

Телефон – целое число

Адрес - строка (не более 40 символов)

Статус – строка (personal/work)

Дата рождения – три целых числа (дд мм гггг), разделенных пробелом, точкой или запятой

Должность - строка (не более 20 символов)

Организация - строка (не более 20 символов)

Поля записи в файле разделены пробелом, точкой или запятой.

**Выходные данные:** список записей, удовлетворяющих условиям (все друзья, которых необходимо поздравить с днем рождения в ближайшую неделю). Таблица записей, таблица ключей, таблица записей по ключам. Информация об эффективности программы.

**Задачи программы:**

1. Ввести таблицу абонентов из файла

2. Записать таблицу абонентов в файл

3. Добавить абонента в таблицу

4. Удалить абонента из таблицы

5. Вывести таблицу абонентов

6. Вывести таблицу абонентов, используя таблицу ключей

7. Вывести таблицу ключей

8. Упорядочить таблицу абонентов по фамилии

9. Упорядочить таблицу ключей по фамилии

10. Вывести информацию об эффективности программы

11. Вывести всех друзей, у которых день рождения в ближайшую неделю

**Способ обращения к программе:** Запуск программы в консоли (./app.exe).

**Возможные ошибки**

1) Открытие несуществующего файла

Сообщение о невозможности открыть файл, программа продолжает работать.

2) Открытие файла с некорректным форматом данных

Сообщение о некорректности входных данных, программа продолжает работать.

**Структуры данных.**

Для хранения данных выбран массив следующих структур.

listp list[MAX]

key\_listp list\_surname[MAX]

1) Абонент

typedef struct

{

char surname[30]; //фамилия

char name[20]; //имя

int phone; //телефон

char address[40]; //адрес

enum //статус (личный/служебный)

{

personal,

work

}st;

union //личные/служебные данные в зависимости от статуса

{

struct data dofb;

struct office ofc;

} type;

} listp;

2) Ключ

typedef struct

{

int index\_src; //индекс в исходной таблице

char surname[30]; //фамилия

} key\_listp;

3) Статус - личный

struct data //дата рождения

{

int d;

int m;

int y;

};

4) Статус - служебный

struct office

{

char position[20]; //должность

char organization[20]; //организация

};

**Функции**

Во всех перечисленных ниже функциях **cnt – текущая длина массива структур** (кол-во записей в таблице)

**void del(int \*cnt)** – удаление записи из таблицы

**void fscan\_list(int \*cnt)** – ввод таблицы из файла

**void fprint\_list(void)** – запись таблицы в файл

**void scan\_key\_list(void)** – заполнение таблицы ключей

**void print\_key\_list(void)** – вывод таблицы ключей

**void scan\_list(int \*cnt)** – добавление абонента в таблицу

**void print\_list(void)** – вывод таблицы

**void print\_list\_using\_keys(void)** – вывод таблицы по ключам

**void bubble\_sort(void \*a, int n, int size, int (\*compare)(const void \*, const void \*))** - сортировка пузырьком; a - массив, n - его длина, size - размер элемента в байтах, compare - функция сравнения элементов

**void sort\_list(int cnt)** – сортировка таблицы

**void sort\_key\_list(int cnt)** – сортировка таблицы ключей

**void spec(int cnt)** – поиск друзей с ДР в ближайшую неделю

**unsigned long long tick(void)** – функция замера времени

**void check\_ef(int len)** – вывод информации об эффективности программы; len - текущая длина массива структур

**Алгоритм**.

Пользователь выбирает любой из предложенных пунктов меню для работы с таблицей. В зависимости от его выбора осуществляется одно из действий: ввод таблицы из файла, вывод таблицы в файл, добавление абонента в таблицу, удаление абонента из таблицы, вывод основной таблицы, вывод основной таблицы, используя таблицу ключей, вывод таблицы ключей, сортировка основной таблицы по фамилии, сортировка таблицы ключей по фамилии, сравнение эффективности двух способов обработки данных, сравнение скорости двух типов сортировки, вывод всех друзей, у которых в ближайшую неделю будет день рождения, выход из программы.

**Тесты.**

|  |  |  |
| --- | --- | --- |
| Содержимое файла | Результат | Пояснение |
|  | File is empty | Пустой файл |
|  | Can’t open file | Файл не существует |
| Balabanov,Petr,abcd,Street3,personal,4.11.2000 | Wrong input data | Номер телефона должен быть целым числом |
| Balabanov,Petr,abcd,Street3,other,4.11.2000 | Wrong input data | Статус имеет только два возможных значения (личный/служебный) |
| Balabanov,TOOLONGNAMEMORETHAN20,  abcd,Street3,personal,4.11.2000 | Wrong input data | Слишком длинное имя (более 20 символов) |
| Balabanov,Petr,123456,Street3,personal,4.11.2000  Ivanov,Ivan,52671,Street5,personal,07.01.1995  Karpov,Lev,4316,Street4,work,teacher,bmstu  Petrov,Semen,568,Street8,personal,29.10.1990  01.11 | Balabanov | Файл успешно прочитан |
| Balabanov,Petr,123456,Street3,personal,4.11.2000  Ivanov,Ivan,52671,Street5,personal,07.01.1995  Karpov,Lev,4316,Street4,work,teacher,bmstu  Petrov,Semen,568,Street8,personal,29.10.1990  01.05 | There are no birthdays next week | Файл успешно прочитан |

**Оценка эффективности**

|  |  |  |
| --- | --- | --- |
| Метод сортировки | Сортируемая таблица | Количество тиков |
| Быстрая | Абонентов | 10731516 |
| Быстрая | Ключей | 3371828 |
| Пузырьком | Ключей | 10007454 |
| Пузырьком | Абонентов | 17815788 |

Для хранения таблицы ключей необходимо дополнительно 25% памяти.

**Вывод**

Таблица ключей сортируется примерно в 3 раза быстрее, чем исходная таблица; быстрая сортировка примерно в 3 раза быстрее, чем сортировка пузырьком.

В случае большого количества записей или большого размера памяти, необходимой для хранения каждой записи, имеет смысл использовать таблицу ключей, также важен выбор ключевого поля.

**Ответы на вопросы**

***1.*** **Как выделяется память под вариантную часть записи?**

Выделяется область памяти, равная размеру 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, ошибка не будет обнаружена.

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

Программист.

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

Таблица ключей содержит индекс элемента в исходной таблице и выбранный ключ. За счет небольших дополнительных затрат памяти позволяет ускорить процесс поиска и сортировки элементов исходной таблицы.

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

Использовать таблицу ключей эффективнее в случае большого количества записей или большого размера памяти, необходимой для хранения каждой записи.

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

При сортировке таблиц эффективнее использовать таблицу ключей. При этом быстрая сортировка работает быстрее, чем сортировка пузырьком за счет меньшего количества перестановок.