|  |  |
| --- | --- |
| 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 **ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04** Программная инженерия

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 2 |

**Название:**

Записи с вариантами. Обработка таблиц

**Дисциплина:** Типы и структуры данных

|  |  |  |  |
| --- | --- | --- | --- |
| Студент | ИУ7-35Б |  | А. В. Толмачев |
|  | (Группа) |  | (И.О. Фамилия) |
|  |  |  |  |

Москва, 2022

**Описание условия задачи**

Ввести список абонентов, содержащий фамилию, имя, телефон, адрес, статус (личный – дата рождения: день, месяц, год; служебный – должность, организация). Найти всех друзей, которых необходимо поздравить с днем рождения в ближайшую неделю.

Создать таблицу, содержащую не менее 40 записей. Упорядочить данные в ней по возрастанию ключей, двумя алгоритмами сортировки, где ключ – фамилия, используя: а) саму таблицу, б) массив ключей.

**Описание ТЗ**

1. **Описание исходных данных**

Исходными данными является следующая информация:

1. Фамилия
2. Имя
3. Номер телефона
4. Адрес
5. Тип статуса
6. Если тип «личный»:
   1. Дата рождения
7. Если тип «служебный»:
   1. Должность
   2. Организация

Исходные данные могут быть обработаны программой либо посредством помещения в файл, в котором каждая строка записи располагается на отдельной строке, либо вводом с помощью соответствующего поля меню – после открытия существующего файла с корректными данными.

Ограничения к исходным данным:

1. Допускается использование только английского языка
2. Максимальная длина фамилии – 20 символов
3. Максимальная длина имени – 20 символов
4. Имя и фамилия могут содержать 1 дефис, для записи двойной фамилии/имени
5. Имя и фамилия не могут содержать любые другие символы, кроме букв из 1 дефиса
6. Номер должен вводиться без кода страны, т. е. состоять из 10 цифр
7. Максимальная длина адреса – 60 символов. Адрес не валидируется, допускается ввод в произвольном формате
8. Максимальная длина должности – 20 символов
9. Максимальная длина организации – 35 символов
10. Поле не может быть «пропущено», т. е. оставлено пустым
11. Дата вводится через точку в формате dd.mm.yyyy.
12. Допустимо опускать незначащий первый 0 у даты
13. Минимально возможный год – 1900
14. Максимально возможный год – 9999
15. Введенное количество дней не должно превышать максимально возможное количество дней в этот день в этот год
16. Чтобы указать статус необходимо ввести “P” – для указания личного статуса, “O” – для указания официального статуса
17. Если после удаления данных из файла, количество структур в нем становится равным 0, то дальнейшая работа с этим файлом считается невозможной.

Ограничения к входному файлу:

1. Передача программе пустого файла считается некорректной ситуацией
2. Исходные данные должны быть записаны в файле следующим образом:
   1. Каждое поле записи располагается с новой строки
   2. После последнего поля последней записи должна быть пустая строка
3. Путь до файла с данными должен указываться относительно папки с исполняемым файлом

Без успешного прочтения программой файла невозможно осуществлять любые другие действия.

1. **Описание задачи, реализуемой программой**

Программа выполняет следующие действия над таблицей:

1 - считать данные из файла

2 - добавить запись

3 - вывод на экран

4 - сортировка исходной таблицы (пузырьком)

5 - сортировка исходной таблицы (быстрая)

6 - вывод таблицы ключей

7 - сортировка таблицы ключей (пузырьком)

8 - сортировка таблицы ключей (быстрая)

9 - вывод упорядоченной исходной таблицы по упорядоченной таблице ключей

10 - вывод эффективности

11 - поиск друзей, которых надо поздравить с днем рождения в ближайшую неделю

12 - удалить записи по фамилии

1. **Способ обращения к программе**

Для запуска программы необходимо из консоли вызвать исполняемый файл ./app.exe.

1. **Описание возможных аварийных ситуаций и ошибок пользователя**
2. Ошибка при открытии файла
3. Ошибка при выделении динамической памяти
4. Ошибка при перевыделении динамической памяти
5. Ошибка при чтении файла

**Описание внутренних структур данных**

В программе содержится структура данных для хранения записи таблицы – информации об абоненте.

**typedef** **struct**

{

**char** firstname[MAX\_FIRSTNAME\_LEN + 1];

**char** lastname[MAX\_LASTNAME\_LEN + 1];

**char** phone\_number[LEN\_PHONE\_NUMBER + 1];

**char** adress[MAX\_LEN\_ADRESS + 1];

**status\_t** status;

**union** {

**private\_t** private;

**offical\_t** offical;

} variate\_status;

} **subscribers\_t**;

Данная структура содержит следующие поля:

1. firstname – строка, фамилия абонента
2. lastname – строка, имя абонента
3. phone\_number – строка, номер телефона абонента
4. adress – строка, адрес абонента
5. status – поле перечисляемого типа, содержит информацию об использованной вариативной части

**typedef** **enum**

{

private = 0,

offical = 1

} **status\_t**;

1. variate\_status - объединение, содержит вариативную часть: структуры с личной и служебной информацией:
   1. private – структура с личной информацией

**typedef** **struct**

{

**date\_birth\_t** date\_birth;

} **private\_t**;

Содержит структуру с датой рождения

**typedef** **struct**

{

**unsigned** **short** **int** day;

**unsigned** **short** **int** month;

**unsigned** **short** **int** year;

} **date\_birth\_t**;

* 1. offical – структура с служебной информацией

**typedef** **struct**

{

**char** job\_title[MAX\_LEN\_JOB\_TITLE + 1];

**char** organization[MAX\_LEN\_ORGANIZATION + 1];

} **offical\_t**;

**Набор тестов**

Негативные тесты:

1. Пустой файл
2. Пустой ввод любого из полей при добавлении записи из программы
3. Некорректный ввод пути до файла (такой файл отсутствует)
4. Некорректный ввод типа статуса
5. Некорректный ввод даты:
   1. Число дней превышает максимально возможное для введенного месяца и года
   2. Число дней меньше 1
   3. Число месяцев меньше 1
   4. Число месяцев больше 12
   5. Число лет меньше 1900
   6. Число лет больше 9999
   7. Введенная строка содержит символы, отличные от «.» и цифр
   8. Длина введенной строки превышает 10 символов
6. Некорректный ввод имени/фамилии:
   1. Введенная строка содержит символы, отличные от букв, символа «-» в корректном количестве
   2. Длина введенной строки превышает 20 символов
7. Некорректный ввод адреса – длина введенной строки превышает 60 символов
8. Некорректный ввод номера телефона
   1. Строка с номером телефона содержит символы, отличные от цифр
   2. Длина введенной строки не равна 10 символам
9. Некорректный выбор действия

Позитивные тесты:

1. Корректный ввод всех полей при добавлении записи
2. Корректно заполненный файл
3. Сортировка файла быстрой сортировкой
4. Сортировка файла пузырьковой сортировкой
5. Сортировка таблицы ключей

**Временная эффективность**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | qsort | qsort with key | bubble sort | bubble sort with key |
| 10 | 3.09 | 3.04 | 9.04 | 7.51 |
| 60 | 26.37 | 18.52 | 159.1 | 94.15 |
| 150 | 73.86 | 61.19 | 968.89 | 706.67 |
| 600 | 249.93 | 209.58 | 15603.9 | 11015.8 |

Для всех замеров было взято среднее из 100 измерений, в микросекундах.

Асимптотика быстрой сортировки: O(n\*log(n))

Асимптотика сортировки пузырьком O(n2)

Объемы занимаемой памяти:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | qsort | qsort with key | bubble sort | bubble sort with key |
| 10 | 1840 | 2160 | 1840 | 2160 |
| 60 | 9936 | 11664 | 9936 | 11664 |
| 150 | 27600 | 32400 | 27600 | 32400 |
| 600 | 110400 | 129600 | 110400 | 129600 |

Для массива структур память выделяется динамически.

**Вывод**

В данной лабораторной работе были получены практические навыки по работе с структурами с вариативной частью, массивом таких структур.

Были рассмотрены два варианта сортировки таблицы: сортировка исходной таблицы и сортировка таблицы ключей, и вывод сортированных значений по ней. Создание таблицы ключей и ее последующая сортировка оказались выгоднее, чем сортировка исходной таблицы: при дополнительных затратах, на хранение таблицы ключей, по памяти в 17%, был получен выигрыш по времени в 37% в случае сортировки пузырьком и в случае быстрой сортировки - 24%.

**Контрольные вопросы**

1. Как выделяется память под вариантную часть записи?

Объем памяти для объединения выделяет по длине максимального поля. Т.е., если есть объединение с полями int, int, double – будет выделено sizeof(double). Поля объединения разделяют одну область памяти. Т.е. при изменении одного поля, изменится значение другого.

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

В этом случае будет некорректное отображение данных, т.к. эти данные будут интерпретироваться программой в соответствии описанию.

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

За правильностью выполнения операций с вариантной частью записи должен следить программист.

1. Что представляет собой таблица ключей, зачем она нужна?Таблица ключей собой представляет массив структур, каждая из которых содержит значение ключа из исходной таблицы, и индекс этого элемента в исходной таблице. Она нужна для того, чтобы сортировать таблицу ключей, а не исходную, чтобы сэкономить время.
2. В каких случаях эффективнее обрабатывать данные в самой таблице, а когда – использовать таблицу ключей?

Эффективнее использовать таблицу ключей – если таблица состоит из записей с большим числом полей и ее требуется часто сортировать. В этом случае при небольших дополнительных затратах по памяти будет получен значительный выигрыш по времени.

Если же запись содержит малое количество полей или таблица имеет маленький размер, то эффективнее использовать исходную таблицу.

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

Для обработки таблиц более предпочтительны способы сортировки, которые не переставляют отсортированные ранее элементы, как, например, сортировка пузырьком или вставками, т.к. перестановка элементов занимает много времени.