|  |  |
| --- | --- |
|  | **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 по предмету

“Типы и структуры данных”

Записи с вариантами. Обработка таблиц.

Выполнила:

Кузнецова Анастасия

Группа:

ИУ7-31Б

Вариант 14

Проверил(а):

# **Цель работы:** приобрести навыки работы с типом данных «запись» (структура), содержащим вариантную часть (объединение, смесь), и с данными, хранящимися в таблицах, произвести сравнительный анализ реализации алгоритмов сортировки и поиска информации в таблицах, при использовании записей с большим числом полей, и тех же алгоритмов, при использовании таблицы ключей; оценить эффективность программы по времени и по используемому объему памяти при использовании различных структур и эффективность использования различных алгоритмов сортировок.

# **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). 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.

**2. Описание ТЗ:**

1. Описание исходных данных и результатов

*Входные данные:*

На вход подаётся имя файла, содержащего записи с информацией об абонентах.

Ограничения для входных данных:

* файл должен существовать
* файл не должен быть пустым
* в файле должно быть не менее 40 записей
* поля записи должны быть записаны в порядке: фамилия абонента, имя, телефон, адрес, вариантное поле: день, месяц, год рождения или должность, компания
* Каждое поле должно быть записано без пробелов
* Поля фамилия и имя должны содержать не более 20 символов
* Поля телефон и должность должны содержать не более 15 символов
* Поля адрес и компания должны содержать не более 30 символов
* Поле месяц должно содержать не более 9 символов
* Поле день рождения – целое число от 1 до 31 включительно
* Поле год рождения – целое число от 1800 до 2021 включительно

Дополнительные входные данные в зависимости от выбора пользователя (Пользователь вводит целое число от 0 до 9 включительно):

2. Фамилия (строка не более 20 символов), имя (строка не более 20 символов), телефон (строка не более 15 символов), адрес (строка не более 30 символов), статус (личный или служебный) – целое число 0 или 1. Если выбран личный статус, то день рождения (целое число от 1 до 31 включительно), месяц рождения (строка не более 9 символов), год рождения (целое число от 1800 до 2021 включительно)

3. Целое число от 1 до 9 (выбор пользователя, по какому полю будет происходить удаление), значение поля (строка для фамилии, имени, телефона, адреса, месяца; целое число для дня, года рождения)

4. Месяц следующей недели (строка не более 9 символов), первый день следующей недели (целое число)

*Выходные данные:*

Программа выводит меню и в зависимости от выбора пользователя выводит:

1. Вывод исходной таблицы
2. Запись добавляется и в таблицу, и в файл
3. Запись удаляется и из таблицы, и из файла
4. Вывод всех записей, подходящих под условие
5. Вывод отсортированной таблицы ключей, вывод исходной несортированной таблицы
6. Вывод отсортированной исходной таблицы
7. Вывод отсортированной с помощью таблицы ключей исходной таблицы
8. Вывод таблицы сравнения эффективности работы программы при обработке данных в исходной таблице и в таблице ключей (вывод затраченного времени в обоих случаях и памяти)
9. Вывод результатов использования различных алгоритмов сортировок (Вывод затраченного времени для каждого случая)

При любом некорректном вводе программа выводит сообщение об ошибке.

1. Описание задачи, реализуемой программой

Программа выполняет задачу, запрошенную пользователем. В программе реализовано меню с выборами действий:

1. Просмотр исходной таблицы
2. Добавление одной записи в конец таблицы
3. Удаление записи по значению указанного поля
4. Поиск друзей, у которых день рождения в ближайшую неделю
5. Просмотр отсортированной таблицы ключей при несортированной исходной таблице
6. Вывод исходной отсортированной таблицы
7. Вывод отсортированной исходной таблицы, используя отсортированную таблицу ключей
8. Вывод результатов сравнения эффективности работы программы при обработке данных в исходной таблице и в таблице ключей
9. Вывод результатов использования различных алгоритмов сортировок
10. Способ обращения к программе

Программу следует вызывать из командной строки с помощью команды:

./a.exe

1. Описание возможных аварийных ситуаций и ошибок пользователей

* Файл не существует
* Файл пустой
* В файле менее 40 записей
* Некорректные данные в файле (см. 2а)
* Некорректный ввод при выборе 2, 3 или 4 действия (см. 2а)

**3. Описание внутренних СД**

Каждая запись хранится в виде структуры:

struct personal\_status  
{  
 int day;  
 char month[SIZE\_MONTH];  
 int year;  
};  
  
struct service\_status  
{  
 char position[SIZE\_POS + 1];  
 char company[SIZE\_COMPANY + 1];  
};  
  
union sub\_status  
{  
 struct personal\_status personal;  
 struct service\_status service;  
};  
  
typedef enum{KIND\_PERSONAL, KIND\_SERVICE} kind\_status;  
  
typedef struct  
{  
 char last\_name[SIZE\_LASTNAME + 1];  
 char name[SIZE\_NAME + 1];  
 char telephone[SIZE\_TEL + 1];  
 char address[SIZE\_ADDRESS + 1];  
 union sub\_status status;  
 kind\_status kind;  
} subscriber;

Записи читаются из файла в массив структур:

subscriber arr\_subscribers[ARR\_SIZE];

Таблица ключей хранится также как массив структур:

typedef struct  
{  
 int index;  
 char last\_name[SIZE\_LASTNAME];  
} type\_key\_table;

type\_key\_table key\_table[ARR\_SIZE];

Для замеров времени используются числа:

clock\_t start;

clock\_t finish;

**4. Набор тестов**

Позитивные тесты:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание | Input | Output |
| 1 | Поиск абонентов, у которых день рождения в ближайшую неделю | ./a.exe subscribers.txt  Choice: 4  Month: October  First day: 4 | Kjnscakn Kjnascl 89102938847 Kjsncj,7 6 october 1976  Kjbckjb KJlckn 89203948857 JHncslms,2 9 october 1980 |

Негативные тесты:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание | Input | Output |
| 1 | Аргументов командной строки меньше допустимого | ./a.exe | Key error |
| 2 | Аргументов командной строки больше допустимого | ./a.exe subscribers.txt odihb | Key error |
| 3 | Файл не существует | ./a.exe ldjvh | File error |
| 4 | Файл пустой | ./a.exe subscribers.txt | File error |
| 5 | Длина фамилии больше 20 символов | ./a.exe subscribers.txt  Khbkcdsjbsalkcnalkcansld lashvf 89302938847 Khclkns,12 12 may 1920  … | Overflow last name |
| 6 | Длина имени больше 20 символов | ./a.exe subscribers.txt  Khbkcdsj lashvfsldvknlhavjkasdlak 89302938847 Khclkns,12 12 may 1920  … | Overflow name |
| 7 | Длина телефона больше 15 символов | ./a.exe subscribers.txt  Khbkcdsjb lashvf 8930293884738264187648 Khclkns,12 12 may 1920  … | Overflow telephone |
| 8 | Длина адреса больше 30 символов | ./a.exe subscribers.txt  Khbkcd lashvf 89302938847 Khclknssldvjhksjdbvkjdbvkshvbksjbvkbsk,12276318 12 may 1920  … | Overflow address |
| 9 | Длина месяца больше 9 символов | ./a.exe subscribers.txt  Khbkcdsj lashvf 89302938847 Khclkns,12 12 makjasjshlasay 1920  … | Overflow month |
| 10 | Длина должности больше 15 символов | ./a.exe subscribers.txt  Khbkcds lashvf 89302938847 Khclkns,12 lsblskdjbvkdvsbdlk dnvsjn  … | Overflow position |
| 11 | Длина компании больше 30 символов | ./a.exe subscribers.txt  Khbkcd lashvf 89302938847 Khclkns,12 ksvudh lahdvlskhbslkfjbslkfbslfkjbslf  … | Overflow company |
| 12 | Вместо дня введены буквы | ./a.exe subscribers.txt  Khbkcd lashvf 89302938847 Khclkns,12 aaa may 1920  … | Input error |
| 13 | Вместо года введены буквы | ./a.exe subscribers.txt  Khbkcdsj lashvf 89302938847 Khclkns,12 12 may aaa  … | Input error |
| 14 | День меньше 1 или больше 31 | ./a.exe subscribers.txt  Khbkcds lashvf 89302938847 Khclkns,12 50 may 1920  … | Incorrect day of birth |
| 15 | Год меньше 1800 или больше 2021 | ./a.exe subscribers.txt  Khbkcds lashvf 89302938847 Khclkns,12 12 may 3000  … | Incorrect year of birth |
| 16 | В файле меньше 40 записей | ./a.exe subscribers.txt  Khbkcdsj lashvf 89302938847 Khclkns,12 12 may 1920 | Too few records |

**5. Оценка эффективности программы по времени и по используемому объему памяти в зависимости от используемого алгоритма сортировки и от сортируемой таблицы.**

В таблицу ключей вынесены фамилии абонентов. Я сравнила время работы программы при сортировке исходной таблицы и при сортировке таблицы ключей. Я выбрала два алгоритма сортировки: сортировку пузырьком и быструю сортировку. Замеры проводила на 100 повторах, брала среднее значение. Результаты представлены ниже:

|  |  |  |
| --- | --- | --- |
|  | Сортировка исходной таблицы | Сортировка таблицы ключей |
| Память (в байтах) | 14400 | 16800 |
| Время сортировки пузырьком (в микросекундах) | 70 | 35 |
| Время быстрой сортировки (в микросекундах) | 20 | 10 |

Сортировка исходной таблицы на 50 % медленнее, чем сортировка таблицы ключей.

Обработка данных с таблицей ключей занимает на 16,67% больше памяти, чем только с исходной таблицей.

**6. Выводы по проделанной работе**

# В результате выполненной лабораторной работы я приобрела навыки работы с типом данных «запись», содержащим вариантную часть, и с данными, хранящимися в таблицах, произвела сравнительный анализ реализации алгоритмов сортировки и поиска информации в таблицах, при использовании записей с большим числом полей, и тех же алгоритмов, при использовании таблицы ключей; оценила эффективность программы по времени и по используемому объему памяти при использовании различных структур и эффективность использования различных алгоритмов сортировок.

**7. Ответы на 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.Объем памяти, необходимый для вариантной части, равен 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, поэтому, контроль за правильностью ее использования возлагается на программиста. В процессе выполнения программы некорректные данные могут привести к ошибке.

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

За правильностью выполнения операций с вариантной частью записи должен следить программист. При обработке записей с вариантами программисту необходимо следить за правильностью хранения и обработки данных, содержащихся в вариантной части.

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

Таблица ключей – это таблица, содержащая индекс элемента в исходной таблице и выбранный ключ. Если мы сортируем таблицу ключей, вместо исходной таблицы, то экономится время, поскольку перестановка записей в исходной таблице, которая иногда может содержать достаточно большое число полей, отсутствует. Однако для размещения таблицы ключей требуется дополнительная память.

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

Эффективнее обрабатывать небольшие объёмы данных в самой таблице, чтобы не занимать лишнюю память и не вводить новые структуры данных. Но когда полей в каждой записи таблицы много и объём данных большой, то эффективнее использовать таблицу ключей. так как в этом случае не будет тратиться время на перестановку записей в исходной таблице.

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

Так как перестановка записей в таблице занимает много времени, то предпочтительнее использовать такие способы сортировки, которые не переставляют уже отсортированные записи в таблице, например, сортировка вставками.