Отчёт

1. **Дисциплина: Типы и структуры данных**
2. **Лабораторная работа №2**
3. **по теме: «Записи с вариантами. Обработка таблиц»**
4. Работу выполнил:
5. студент группы ИУ7-32Б
6. Иванов Всеволод

# Описание условия задачи

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)

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.

Провести количественную оценку времени и памяти, затрачиваемой при сортировках таблицы и массива ключей, сортировками различных сложностей.

# Описание технического задания

Программа для редактирования, сортировки ввода и вывода на экран и в файл таблицы записей, содержащих информацию о абонентах. Записи являются вариативными в зависимости от статуса абонента. Область применения — таблицы до 100 штук записей. Сроки выполнения — 2 недели.

Основой для разработки программы послужила потребность в хранении и обработке связанных данных различных типов. Конкретно для этой программы была поставлена цель реализовать ряд функций для обработки информации о абонентах.

Для реализации этой задачи использовался тип запись, содержащий поля с различной основной информацией об абонентах (Фамилия, Имя, Телефон, Адрес, Статус), а также и вариативной (личный: дата рождения, служебный: должность и организация).

Программа должна:

* Содержать указание формата и диапазона данных при вводе и (или) добавлении записей
* Содержать указание операций, производимых программой
* Иметь пояснения при выводе результата
* Иметь возможность добавления записей в конец таблицы и удаления записи по значению указанного поля
* Выводить отсортированную таблицу ключей при несортированной исходной таблице
* Выводить упорядоченную исходную таблицу
* Выводить исходную таблицу в упорядоченном виде, используя упорядоченную таблицу ключей
* Отображать результаты сравнения эффективности работы программы при обработке данных в исходной таблице и в таблице ключей
* Отображать результаты использования различных алгоритмов сортировок.

# Этап постановки

Исходными данными является файл с таблицей информации об абонентах. Результатом является файл с отредактированной таблицей и сравнение разных методов сортировок.

Обращение к программе происходит выбором функций программы путём введения их кодов в консоли. Вывод результатов работы программы также производится в консоль.

Сортировка таблицы производится по фамилии абонентов

Возможными ошибками являются:

1) Отсутствие выбранного файла

2) Некорректный формат данных вводимой записи

2.1) Превышение установленного ограничения по длине символьных строк

2.2) Ввод несуществующего статуса

2.3) Несуществующая дата

2.4) Недопустимые символы в номере телефона

3) Удаление поля с несуществующим индексом

4) Выбор для удаления несуществующего поля

5) Переполнение таблицы

6) Сортировка пустой таблицы

Особых требований для технических средств, аппаратной совместимости, программных средств не имеется.

# Описание СД

Для хранения абонентов была создана запись sub\_t состоящая из:

* Массива 21 символов surname, хранящего фамилию
* Массива 21 символов name, хранящего имя
* Массива 17 символов phone, хранящего телефонный номер
* Массива 21 символов address, хранящего адрес
* Массива 21 символов status, хранящего статус
* Вариативного поля info, состоящего из структуры, хранящей день рождения (целые числа day, month, year) и структуры, хранящей должность и организацию (массивы 21 символов post, org)

Для хранения ключей была создана запись key\_t, состоящая из

* Целого числа ind, хранящего индекс поля в исходной таблице
* Массива 21 символов key, хранящего ключ для сортировки (фамилию)

Хранение таблицы абонентов и ключей осуществляется с помощью массивов записей sub\_t и key\_t соответственно

# Описание алгоритма

В для сортировки были выбраны алгоритмы bubble sort и quick sort. Для сортировки с использованием ключей производится создание таблицы ключей по исходной таблице, и сортировка ключей. После этого возможен вывод неупорядоченной исходной таблицы по упорядоченным ключам.

Оценка времени работы сортировок производится путём замера времени перемешивания таблицы в случайном порядке, создания таблицы ключей (если такая используется) и самой сортировки. После этого производится замер времени, затраченного не на сортировку, и вычитается из общего времени. Ввиду того, что функция clock() не способна зарегистрировать время выполнения одной сортировки, сортировка производится 100000 раз, а потом вычисляется среднее время сортировки.

# Набор тестов

Негативные тесты

|  |  |  |
| --- | --- | --- |
| Ввод | Проверяемый случай | Вывод |
| not\_file.txt | Несуществующий файл | Ошибка: файл невозможно открыть |
| 29.02.2001 | 29 февраля невисокосного года | Ошибка: дата некорректна |
| 32.03.2000 | Несуществующая дата | Ошибка: дата некорректна |
| a.03.2000 | Ввод символов в поле даты | Ошибка: неправильный формат ввода |
| afaf aafa | Ввод символов в поле команды | Ошибка: неправильный формат ввода |
| \*выбор добавления записи\* | Переполнение таблицы (добавление 101 записи) | Ошибка: таблица имеет максимальный допустимый размер |
| +7-1a2-231-23-12 | Использование недопустимых символов в номере телефона | Ошибка: номер содержит недопустимые символы |
| 123456789012345678901 | Превышение ограничения на количество полей в строчном поле (фамилия, имя, телефон, адрес, должность, организация) | Ошибка: неправильный формат ввода |
| 50 | Использование несуществующей команды | Ошибка: команда не сущесвует |
| gender | Использование удаления по несуществующему полю | Ошибка: неправильный формат ввода |
| File name: text.err  11 (scan file) | Сканирование несуществующего файла | Ошибка: файл невозможно открыть |

Тепличные тесты

|  |  |  |
| --- | --- | --- |
| Ввод | Проверяемый случай | Вывод |
| Команда: 00 | Выход из программы | Выход из программы  \*завершение работы\* |
| Команда: 01 | Вывод функций | Таблица доступных функций |
| Команда: 10  text.txt | Смена файла | Файл text.txt отрыт успешно |
| Команда: 11 | Сканирование файла | Считано 40 записей |
| Команда: 12 | Печать в файл | Файл text.txt был успешно записан |
| Команда: 20 | Печать таблицы | Таблица абонентов |
| Команда: 21  Egor  Shatunov  +7-915-231-2313  Petrovskaya5  personal  26.12.1991 | Добавление новой записи | Запись была успешно добавлена в конец таблицы |
| Команда: 22  status  official | Удаление записи по значению указанного поля | Удалено 8 записей |
| Команда: 30, 31 | Сортировка по таблице (bubble, quick) | Отсортированная таблица записей |
| Команда: 32, 33 | Сортировка по таблице ключей (bubble, quick) | Отсортированная таблица ключей  Неотсортированная таблица записей, выведенная в порядке таблицы ключей |
| Команда: 40 | Сравнение затраченного времени и памяти сортировок | Время и память, затраченные на сортировку таблицы  Сравнение в процентах эффективности алгоритмов сортировки и объектов сортировки |
| Команда: 41  30.12.2001 | Отображение людей с днём рождения в следующую неделю | Таблица людей с днями рождения 31.12-06.01 |

Граничные тесты

|  |  |  |
| --- | --- | --- |
| Ввод | Проверяемый случай | Вывод |
| Команда: 21  Gregor-Gregor-gregor  Zalatoglavovichensky  +7-915-231-23-13  Petrovskaya\_st.,53/2  official  PR\_zam\_director\_curr  Very\_long\_names\_ink. | Добавление новой записи с использованием предельных значений для каждого поля (фамилия, имя, адрес, должность, организация – 20 символов, телефон – 16) | Запись была успешно добавлена в конец таблицы |
| …  29.02.2004 | Добавление новой записи с использованием предельных значений для даты | Запись была успешно добавлена в конец таблицы |
| …  31.03.2004 | Добавление новой записи с использованием предельных значений для даты | Запись была успешно добавлена в конец таблицы |
| Новая запись | Достижение максимального размера таблицы | Запись была успешно добавлена в конец таблицы |

# Сравнение эффективностей

(Сортировка производилась по полю фамилии абонента)

Таблица из 10 записей:

Сортировка quicksort с ключами и без:

**Время** с ключами: меньше на 30%

**Память** с ключами: больше на 14.2%

Сортировка bubble sort с ключами и без:

**Время** с ключами: меньше на 20%

**Память** с ключами: больше на 14.2%

Сортировка bubble sort и quicksort с ключами:

**Время** quicksort: меньше на 18%, **память** одинаковая.

Таблица из 40 записей:

Сортировка quicksort с ключами и без:

**Время** с ключами: меньше на 30%

**Память** с ключами: больше на 14.2%

Сортировка bubble sort с ключами и без:

**Время** с ключами: меньше на 30%

**Память** с ключами: больше на 14.2%

Сортировка bubble sort и quicksort с ключами:

**Время** quicksort: меньше на 30%, **память** одинаковая.

Таблица из 90 записей:

Сортировка quicksort с ключами и без:

**Время** с ключами: меньше на 35%

**Память** с ключами: больше на 14.2%

Сортировка bubble sort с ключами и без:

**Время** с ключами: меньше на 30%

**Память** с ключами: больше на 14.2%

Сортировка bubble sort и quicksort с ключами:

**Время** quicksort: меньше на 80%, **память** одинаковая.

**Итог сравнительного анализа:**

Алгоритм сложности n\*log2(n) (quicksort) показывает значительное преимущество во времени сортировки при достаточно больших размерах таблицы по сравнению с алгоритмом сложности n^2 (bubblesort).

Сортировка с использованием таблицы ключей для обоих алгоритмов показывает примерно одинаковый выигрыш по времени в 30% при разных размерностях по сравнению с сортировкой самой таблицы. Объём затраченной памяти неизменно больше на 14.2%. Таким образом, преимуществом в сортировке таблице ключей является меньшее время, а недостатком дополнительно затрачиваемая память.

# Ответы на вопросы

1. Как выделяется память под вариантную часть записи?

Под вариативную запись отводится память, равная максимальной занимаемой памяти среди всех входящих в неё полей данных.

2.Что будет, если в вариантную часть ввести данные, несоответствующие описанным? 3.Кто должен следить за правильностью выполнения операций с вариантной частью записи?

Так как на этапе трансляции тип данных в вариативной части не проверяется, то существует возможность успешно вносить в неё данные, несоответствующие описанным. Поэтому контроль за правильностью выполнения операций с вариативной частью должен следить программист

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

Таблица ключей представляет массив структур, а которых содержатся два поля: индекса поля в таблице и значение ключа данного поля. Таблица ключей нужна для уменьшения времени сортировки за счёт более быстрых перестановок (ключей по сравнению с исходными записями)

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

В случае, когда требуется уменьшить время сортировки, эффективнее обрабатывать данные с помощью таблицы ключей. В случае, когда требуется уменьшить объём занимаемой памяти, эффективнее обрабатывать данные с помощью самой таблицы.

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

Предпочтительнее способ сортировки, который использует таблицу ключей, так как она даёт ощутимо более быструю работу при незначительной занимаемой дополнительной памяти.

# Выводы по проделанной работе

Из проведённого анализа следует, что преимуществом в сортировке таблице ключей является меньшее время, а недостатком дополнительно затрачиваемая память. При этом время выигрыш во времени является существенным, в то время как дополнительная занимаемая память несущественна.