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 Цветков Иван Алексеевич

Группа ИУ7 – 33Б

2020 г.

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.

Мое задание по 20-ому варианту:

Ввести список квартир, содержащий адрес, общую площадь, количество комнат, стоимость квадратного метра, первичное жилье или нет (первичное – с отделкой или без нее; вторичное – время постройки, количество предыдущих собственников, количество последних жильцов, были ли животные). Найти все вторичное 2-х комнатное жилье в указанном ценовом диапазоне без животных.

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 :

1. Адрес (город) — до 100 символов
2. Площадь квартиры
3. Количество комнат
4. Стоимость одного квадратного метра
5. Класс жилья:
    1. Первичное (отделка — да или нет)
    2. Вторичное (год постройки, количество прошлых владельцев, количество прошлых жильцов, животные — да или нет)

2. Целое число, представляющее собой номер команды:

число типа int в диапазоне от 0 до 13

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.

Функции программы:

1. Загрузить список квартир из файла
2. Добавить квартиру в конец списка
3. Удалить квартиры по значению площади
4. Найти все двухкомнатное вторичное жильё без живтотных в указанном ценовом диапазоне
5. Отсортировать таблицу пузырьком
6. Отсортировать таблицу ключей пузырьком
7. Отсортировать таблицу qsort
8. Отсортировать таблицу ключей qsort
9. Вывести сравнение сортировок пузырьком и qsort
10. Вывести таблицу
11. Вывести таблицу по таблице ключей
12. Вывести таблицу ключей
13. Очистить таблицу

0 — 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 введено 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 введён город

Входные данные: строка выше 100 символов

Выходные данные: сообщение «Ошибка: неверно введён город»

6. Неверно введена площадь квартиры

Входные данные: не число или число, меньшее 1 или большее 100000

Выходные данные: сообщение «Ошибка: неверно введена площадь квартиры»

7. Неверно введено количество комнат

Входные данные: не число или число, меньшее 1 или большее 100

Выходные данные: сообщение «Ошибка: неверно введено количество комнат»

8. Неверно введена цена за 1 квадратный метр

Входные данные: не число или число, меньшее 1 или большее 1000000

Выходные данные: сообщение «Ошибка: неверно введена цена за 1 квадратный метр»

9. Неверно введен класс квартиры

Входные данные: не число или число, меньшее 1 или большее 2

Выходные данные: сообщение «Ошибка: неверно введен класс квартиры»

10. Неверно введено наличие декора

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

Выходные данные: сообщение «Ошибка: неверно введено наличие декора»

11. Неверно введен год постройки

Входные данные: не число или число, меньшее 1900 или большее 2020

Выходные данные: сообщение «Ошибка: неверно введен год постройки»

12. Неверно введено количество прошлых владельцев

Входные данные: не число или число, меньшее 1 или большее 1000

Выходные данные: сообщение «Ошибка: неверно введено количество прошлых владельцев»

13. Неверно введено количество прошлых жильцов

Входные данные: не число или число, меньшее 1 или большее 1000

Выходные данные: сообщение «Ошибка: неверно введено количество прошлых жильцов»

14. Неверно введено наличие животного

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

Выходные данные: сообщение «Ошибка: неверно введено наличие животного»

15. Неверно введена начальная цена для поиска

Входные данные: не число или число, меньшее 0

Выходные данные: сообщение «Ошибка: неверно введена начальная цена для поиска»

16. Неверно введена конечная цена для поиска
    Входные данные: не число или число, меньшее 0
    Выходные данные: сообщение «Ошибка: неверно введена конечная цена для поиска»
17. Конечная цена не может быть меньше начальной
    Входные данные: конечная цена больше начальной
    Выходные данные: сообщение «Ошибка: конечная цена не может быть меньше начальной»
18. Файл пустой
    Входные данные: пустой файл
    Выходные данные: сообщение «Ошибка: файл пуст»

Описание структуры данных

Тип данных table_t представляет собой структуру, содержащую массивы стуктр appartment_t flats_arr[1000], key_t keys[1000], размер таблицы, который находится в целочисленном поле size и maxsize, которое является численным полем и хранит в себе максимально возможное количество записей.

```
typedef struct table
{
    appartment_t *flats_arr;
    keys_t *keys;
    int size;
    int maxsize;
} table_t;
```

appartment_r – информация об одной квартире

```
typedef struct
{
    char address[MAX_ADDRES_NAME + 1];
    int square;
    int rooms;
    int one_m_cost;
    int is_primary;
    primary_check_t flat;
} appartment_r;

MAX_ADDRES_NAME = 100
```

Поля структуры:
1. address[MAX_ADDRES_NAME + 1] — адрес (город) квартиры
2. int square — площадь квартиры
3. int rooms - количество комнат
4. int one_m_cost — стоимость 1 квадратного метра квартиры
5. int is_primary — первичная или вторичная квартира
6. primary_check_t flat — вариативное поле, которое представляет собой структуру и хранит информацию о квартире в зависимости от её класса

key_t — структура, которая хранит в себе ключи для одного из полей основной таблицы

```
typedef struct keys
{
    int square;
    int id;
} key_t;
```

Поля структуры:
1. int square — площадь квартиры
2. int id - номер квартиры в основной таблице

Тип prim_sec_r является вариантной частью:

```
typedef union
{
    primary_t primary;
    not_primary_t secondary;
} primary_check_t;
```

primary_t и not_primary_t — отвечают за хранение информации по каждому классу жилья (первичное или вторичное):

```
typedef struct
```

```
{
    int decorated;
} primary_t;
```

Поля структуры:
1. int decorated — есть ли декор у квартиры (1 — да, 0 - нет)

typedef struct

```
{
    int build_year;
    int previous_owners;
    int previous_rezidents;
    int animals;
} not_primary_t;
```

Поля структуры:
1. int build_year — год постройки здания с квартирой
2. int previous_owner — количество прошлых владельцев
3. int previous_rezidents - количество прошлых жильцов
4. int animals — ыбло ли животное в квартире (1 — да, 0 - нет)

Описание алгоритма

1. Выводится меню программы
2. Пользователь вводит пункт меню, который отвечает за какое-то из действий, определённых программой

3. Ввод осуществляется до того момента, пока не будет введён 0, который является выходом из программы

НАБОР ТЕСТОВ

	Название теста	**Пользовательский ввод**	**Результат**
1	Некорректный ввод пункта меню	iu	Ошибка: пункт меню введён неверно
2	Некорректный ввод пункта меню	15	Ошибка: пункт меню введён неверно
3	Неверно введено имя файла	no_file.txt	Ошибка: неверно введено имя файла
4	В файле находятся недопустимые данные	some_file.txt	Ошибка: неверные записи в файле
5	Количество возможных записей переполнено	Файл с более, чем 1000 строкового	Ошибка: количество записей переполнено
6	Количество возможных записей в таблице переполнено	При добавлении записи стало 1001 записей в табилце	Ошибка: количество записей переполнено
7	Неверно введён адрес (город)	Строка с более, чем 100 символов	Ошибка: неверно введён город
8	Неверно введена площадь квартиры	Iu	Ошибка: неверно введена площадь

		или -100	квартиры
9	Неверно введено количество комнат	Iu или -1001	Ошибка: неверно введено количество комнат
10	Неверно введена цена за 1 квадратный метр	Iu или -17	Ошибка: неверно введена цена за 1 квадратный метр
11	Неверно введен класс квартиры (1 или 2)	Iu или 0	Ошибка: неверно введен класс квартиры
12	Неверно введен год постройки	Iu или -100	Ошибка: неверно введен год постройки
13	Неверно введено количество прошлых владельцев	Iu или -100	Ошибка: неверно введено количество прошлых владельцев
14	Неверно введено количество прошлых жильцов	Iu или -100	Ошибка: неверно введено количество прошлых жильцов
15	Неверно введено наличие животного	Iu	Ошибка: неверно

	(0 или 1)	или -2	введено наличие животного
16	Неверно введена начальная цена для поиска	Iu или -23	Ошибка: неверно введена начальная цена для поиска
17	Неверно введена конечная цена для поиска	Iu или -25	The table is empty.
18	Конечная цена не может быть меньше начальной	Конечная цена больше начальной	Ошибка: конечная цена не может быть меньше начальной
19	Добавление верной записи в конец файла	Верная запись	Квартира была успешно добавлена
20	Ввод записей из файла с правильными записями	Правильный файл	Список квартир из файла был успешно загружен
21	Удаление квартиры по значению площади, которая есть в таблице	Значение площади, которое есть в таблице	Успешно удалены квартиры из таблицы
22	Удаление квартиры по значению площади, которой нет в таблице	Значение площади, которой нет в таблице	Не найдены квартиры с такой площадью
23	Поиск по условию (такие записи есть)	Начальная и конечная цена	Таблица с подходящими квартирами

24	Поиск по условию (таких записей нет)	Начальная и конечная цена	Ничего не было найдено по заданным границам цены
25	Ввод пункта меню по любой из сортировок (Пункты меню 5 — 8)	Полная таблица или таблица ключей, где есть хотя бы 1 запись	Таблица успешно отсортирована
26	В любом пункте, где нужно использование таблицы	Таблица без записей (размера 0)	Таблица пустая
27	Запуск сравнения сортировок	Имя валидного файла	Информация о сортировке каждой таблицы
28	Вывод любой таблицы	Таблица хотя бы с одной записью	Таблица с «шапкой»
29	Ввод нуля	Пункт меню 0	Таблица успешно очищена
30	Ввод нуля	Пункт меню 0	Выход из программы, очистка консоли
31	Пустой файл	Пустой файл	Ошибка: Файл пуст

Оценка эффективности

Сортировка каждой таблицы будет измеряться в тактах процессора на процессоре с частотой 35000000 Гц

Время сортировки:

Количество записей	Полная таблица		Таблица ключей	
	Сортировка «bubble»	Сортировка «qsort»	Сортировка «bubble»	Сортировка «qsort»
60	232492	102952	113760	20272
120	1006626	146136	412320	27588
240	3665518	200220	1559056	52168
480	14610038	453128	9414448	104386

Объем занимаемой памяти:

Количество записей	Полная таблица	Таблица ключей

60	8160	480
120	16320	960
240	32640	1920
480	63360	3840

Таблица соотношений памяти и времени:

Количество записей	% памяти, занимаемый таблицей ключей от всей таблицы	Во сколько раз сортировка таблицы ключей быстрее сортировки всей таблицы (“bubble”)	Во сколько раз сортировка таблицы ключей быстрее сортировки всей таблицы (“qsort”)
60	~6%	~2 раз	~5 раз
120	~6%	~2.4 раз	~5.3 раз
240	~6%	~2.4 раз	~3.8 раз
480	~6%	~1.5 раз	~4.3 раз

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