Лабораторная работа №3. Отчет.

**Записи с вариантами. Обработка таблиц.**

Кононенко С.Д. ИУ7-31

**Условие 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 - общая площадь квартиры), используя:

а) саму таблицу,

б) массив ключей

Ввести список квартир, содержащий адрес, общую площадь, количество комнат, этаж, стоимость квадратного метра (первый – есть ли подвал; последний – наличие чердака, его площадь и возможность приватизации; другой – наличие балкона и его площадь). Найти все квартиры, с возможностью приватизации

чердака.

**Тех. Задание**

**Входные данные:**

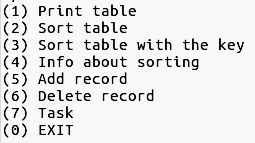
1. Цифры 0-7 для выбора пункта меню работы с программой
2. Строка длиной не более 50 символов для ввода адреса новой записи и числа для заполнения информации о площади, количестве комнат, этаже, стоимости квадратного метра и др.
3. Числа для выбора удаляемой из таблицы записи

**Выходные данные:**

1. Список всех записей в порядке по номеру.
2. Список из записей отсортированных по полю - общая площадь квартиры.
3. Время сортировки таблицы двумя методами сортировки.
4. Время сортировки таблицы ключей двумя методами сортировки.
5. Информация о затраченной памяти на таблицу записей и таблицу ключей.
6. Список записей, удовлетворяющих критериям поиска (по заданию - все квартиры, с возможностью приватизации чердака).

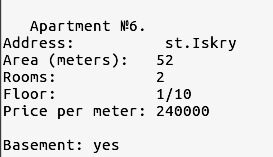
**Работа с программой:**

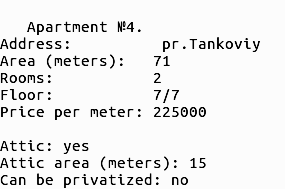
Работа с программой осуществляется с помощью меню:



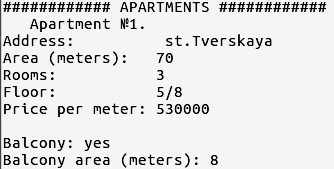
**Вывод записей на экран:**

Первые 5 строк - адрес(address), площадь(area), количество комнат(rooms), этаж(floor) и цена за квадратный метр(price per meter) присутствуют во всех записях. Остальные поля зависят от этажа:

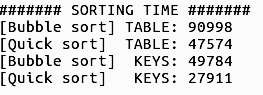
* для первого этажа - наличие подвала(basement);
* для последнего - наличие чердака(attic), при его наличии - площадь(attic area) и возможность его приватизации(can be privatized);



* для остальных этажей - наличие балкона(balcony) и при его наличии - площадь балкона(balcony area).



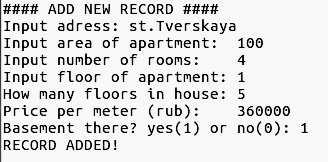
**Сортировка таблицы записей и сортировка таблицы записей с помощью таблицы ключей:** Сортировка таблицы записей без и с помощью таблицы ключей - упорядочивание элементов таблицы по возрастанию площади квартиры(area). Упорядоченная таблица выводится на экран.

**Информация о сортировки таблицы:** таблица записей без и с помощью таблицы ключей сортируется с помощью двух разных методов сортировки - сортировки методом пузырька и быстрой сортировки qsort. Время сортировки выводится на экран:

*первые две строки - сортировка полной таблицы записей,*

*вторые две строки - сортировка таблицы по ключам*

**Добавление новой записи в таблицу:**

Для добавления новой записи пользователю предлагается ввести данные о квартире, следуя запросам. При правильно заданных данных выводится сообщение “RECORD ADDED!”

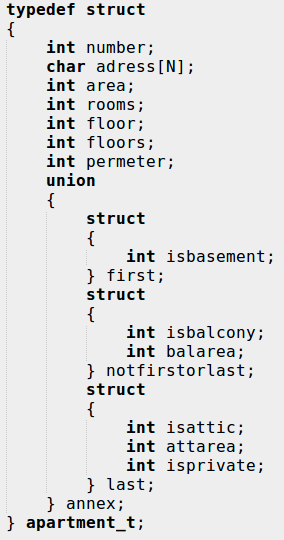
**Удаление записи из таблицы:**

Для удаления записи пользователь должен ввести порядковый номер удаляемой записи. При успешном выборе - сообщение “RECORD DELETED!”

**Поиск всех квартир с возможностью приватизации чердака:**

Пункт меню (7)Task выводит на экран все записи о квартирах, располагающихся на последнем этаже, имеющим чердак, который можно приватизировать.

**Завершение работы с программой:** Пункт меню (0) EXIT завершает работу с программой.

**Структуры данных**

**Структура записи:**

*number* - порядковый номер;

*address* - адрес квартиры;

*area* - площадь квартиры;

*rooms* - количество комнат;

*floor* - этаж;

*floors* - количество этажей в доме;

*permeter* - цена за кв.метр;

*isbasement* - наличие подвала;

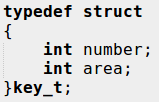
*isbalcony* - наличие балкона;

*balarea* - площадь балкона;

*isattic* - наличие чердака;

*attarea* - площадь чердака;

*isprivate* - возможность приватизации



**Структура записи с ключом:**

*number* - порядковый номер;

*area* - площадь квартиры;

**Сравнение сортировки таблицы из 40 элементов:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Таблица записей** | **Таблица ключей** | **%** |
| **Сортировка пузырьком** | 90998 | 46784 | 1,95 |
| **Qsort** | 47574 | 21911 | 2,17 |

Время сортировки полной таблицы занимает в среднем в **2-2.5 раза больше** времени, чем сортировка с помощью таблицы ключей.

Сортировка была произведена двумя разными по эффективности методами: метод пузырька имеет квадратичную сложность О(n^2).

Быстрая сортировка имеет сложность O(n\*log(n))

Qsort быстрее сортировки пузырьком в **2-3** (при большом количестве сортируемых элементов), следовательно эффективнее для сортировки информации использовать Qsort.

Один элемент таблицы записей занимает 88 байт.

Один элемент таблицы ключей занимает 8 байт.

Используя в **1,1 раз больше памяти** (на таблицу ключей), сортировка занимает **в 2-2.5 раза меньше времени**.

Эффективность использования таблицы ключей для сортировки информации доказана.

**Вопросы к лабораторной работе:**

*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.

*Что будет, если в вариантную часть ввести данные, не соответствующие описанным?*

Тип данных в вариативной части не проверяется при компиляции, поэтому контроль за правильностью ее использования возлагается на программиста.

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

Контроль за правильностью выполнения операций с вариантной частью записи возлагается на программиста.

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

При больших размерах таблиц поиск данных, имеющих указанный ключ, может потребовать больших затрат времени. Если же помимо поиска требуется произвести сортировку данных, то временные затраты многократно возрастут, так как потребуется осуществлять их перестановку. За счет создания дополнительной таблицы ключей можно уменьшить время обработки главной таблицы. Таблицы ключей содержит индекс элемента в исходной таблице и выбранный ключ.

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

Если каждая запись таблицы состоит из небольшого количества полей, то полезнее обрабатывать данные в таблице.

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

Для обработки таблицы предпочтительнее использовать алгоритмы сортировки, предполагающие наименьшее количество операций перестановки - имеющие наименьшую алгоритмическую сложность.