



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 ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ**

### **ЗАПИСИ С ВАРИАНТАМИ. ОБРАБОТКА ТАБЛИЦ**

Студент Жаринов М. А.

Вариант 7

Группа ИУ7-32Б

Название предприятия НУК ИУ МГТУ им. Н. Э. Баумана

Студент \_\_\_\_\_ Жаринов М. А.

Преподаватель \_\_\_\_\_ Барышникова М. Ю.

**2024**

### **Описание условия задачи**

Ввести список абонентов, содержащий фамилию, имя, номер телефона, адрес (улица, дом), статус абонента:

1. Друзья:

а. Дата рождения: день, месяц, год

2. Коллеги:

а. должность

б. организация

Вывести список всех друзей, которых необходимо поздравить с днем рождения в ближайшую неделю (текущая дата вводится пользователем).

Упорядочить данные в таблице по возрастанию ключей, двумя алгоритмами сортировки, где ключ – любое невариантное поле, используя:

а) саму таблицу,

б) массив ключей.

Реализовать возможность добавления и удаления записей в ручном режиме, просмотр таблицы, просмотр таблицы в порядке расположения таблицы ключей.

## **Техническое задание**

### **Исходные данные:**

**Номер пункта меню** – Целое число от 1 до 12.

**Текстовый файл с таблицей структур:** Файл, содержащий поля таблицы структур, разделенные символом переноса строки.

**Прочие данные, запрашиваемые при выборе соответствующих пунктов меню:** Данные, по формату совпадающие с соответствующими полями структуры.

### **Выходные данные:**

Исходные и отсортированные таблицы структур и ключей, прочие данные полученные путем вызовы других пунктов меню.

### **Реализуемые задачи:**

Пункт меню 1 – Выход из программы

Пункт меню 2 – Чтение таблицы из файла

Пункт меню 3 – Сохранение таблицы в файл

Пункт меню 4 – Печать таблицы структур

Пункт меню 5 – Печать таблицы ключей

Пункт меню 6 – Печать отсортированной таблицы структур

Пункт меню 7 – Печать отсортированной таблицы ключей

Пункт меню 8 – Печать таблицы структур в отсортированном виде с помощью таблицы ключей

Пункт меню 9 – Добавление записи в конец таблицы структур

Пункт меню 10 – Удаление записи по введенному значению ключа

Пункт меню 11 – Вывод всех друзей с днем рождения в ближайшую неделю

Пункт меню 12 – Проведение сравнительного анализа различных методов и алгоритмов сортировок

### **Способ обращения к программе:**

Строка запуска программы ./app.exe

После запуска с помощью меню нужно заполнить таблицу или запустить анализ сортировок.

**Аварийные ситуации:**

1. EOF в вводе. Код ошибки 1

Все остальные ошибочные ситуации не являются аварийными и вызывают повторное приглашение к вводу или возвращение в меню.

## Внутренние структуры данных

Сведения о датах хранятся в структуре date\_t.

```
typedef struct
{
    int day;
    int month;
    int year;
} date_t;
```

Поле day – Номер дня

Поле month – Номер месяца

Поле year – Год

Ключи в таблице ключей хранятся в структуре key\_elem\_t.

```
typedef struct
{
    char value[LASTNAME_LEN + 1];
    int index;
} key_elem_t;
```

Поле value – Строка – фамилия абонента

Поле index – Индекс записи в исходной таблице структур

Записи в таблице абонентов хранятся в структуре subscriber\_t.

```
typedef struct
{
    char lastname[LASTNAME_LEN + 1];
    char name[NAME_LEN + 1];
    char number[NUMBER_LEN + 1];
    char address[ADDRESS_LEN + 1];
    enum
    {
        FRIEND = 0,
        COLLEAGUE
    } status;
    union
    {
        date_t birthday;
        struct
```

```

{
    char job[JOB_LEN + 1];
    char organisation[ORGANISATION_LEN + 1];
} position;
} info;
} subscriber_t;

```

Поле `lastname`— строка – фамилия абонента

Поле `name`— строка – имя абонента

Поле `number`— строка – номер телефона абонента

Поле `address`— строка – адрес абонента

Перечисляемый тип `status`— характеризует статус абонента(FRIEND – друзья, COLLEAGUE - коллеги)

Объединение `info`— информация о абоненте как друге или коллеге в зависимости от его статуса (вариантная часть) – Содержит следующие поля:

Поле `birthday`— дата рождения абонента-друга

Поле `position`— структура сведений о абоненте-коллеге. Содержит следующие поля:

Поле `job`— строка – должность абонента-коллеги

Поле `organisation`— строка – организация абонента-друга

Для упрощения работы с таблицами они хранятся в структурах.

Таблица ключей хранится в структуре `key_table_t`.

```

typedef struct
{
    key_elem_t keys[MAX_TABLE_LEN];
    int len;
} key_table_t;

```

Поле `keys` – массив записей - ключей

Поле `len` – длина таблицы

Таблица записей хранится в структуре `table_t`.

```

typedef struct
{
    subscriber_t subscribers[MAX_TABLE_LEN];

```

```
int len;  
} table_t;
```

Поле `subscribers` – массив записей

Поле `len` – длина таблицы

## Описание алгоритма

1. Программа запрашивает номер элемента в меню, считывает его и выполняет соответствующее действие

Пункт меню 1 – Выход из программы

Пункт меню 2 – Чтение таблицы из файла

1. Программа запрашивает имя файла и считывает его.
2. Программа производит чтение указанного файла в таблицу структур.

Пункт меню 3 – Сохранение таблицы в файл:

1. Программа запрашивает имя файла и считывает его.
2. Программа производит запись в указанный файл таблицы структур.

Пункт меню 4 – Печать таблицы структур:

1. Программа проверяет, что таблица не пустая и печатает заголовок таблицы.
2. Программа печатает все записи таблицы.

Пункт меню 5 – Печать таблицы ключей:

1. Программа проверяет, что таблица не пустая.
2. Программа обновляет таблицу ключей по таблице структур.
3. Программа печатает заголовок и все записи таблицы ключей.

Пункт меню 6 – Печать отсортированной таблицы структур:

1. Программа проверяет, что таблица не пустая.
2. Программа сортирует таблицу структур.
3. Программа печатает заголовок и все записи таблицы структур.

Пункт меню 7 – Печать отсортированной таблицы ключей:

1. Программа проверяет, что таблица не пустая.
2. Программа обновляет таблицу ключей по таблице структур.
3. Программа сортирует таблицу ключей.
4. Программа печатает заголовок и все записи таблицы ключей.

Пункт меню 8 – Печать таблицы структур в отсортированном виде с помощью таблицы ключей:

1. Программа проверяет, что таблица не пустая.



2. Программа обновляет таблицу ключей по таблице структур.
3. Программа сортирует таблицу ключей.
4. Программа печатает заголовок таблицы.
5. Программа печатает записи из таблицы структур в порядке, сохраненном в таблице ключей.

Пункт меню 9 – Добавление записи в конец таблицы структур:

1. Программа проверяет, что таблица не максимального размера.
2. Программа запрашивает очередное поле структуры и записывает его в таблицу.

Пункт меню 10 – Удаление записи по введенному значению ключа:

1. Программа проверяет, что таблица не пустая.
2. Программа запрашивает значение поля Фамилия к поиску.
3. Программа ищет первую запись с таким значением поля и удаляет ее.

Пункт меню 11 – Вывод всех друзей с днем рождения в ближайшую неделю:

1. Программа проверяет, что таблица не пустая.
2. Программа запрашивает текущую дату.
3. Программа находит дату ближайшего дня рождения очередного друга и считает, через сколько дней оно будет.
4. Программа печатает запись в случае, если день рождения ближе, чем через неделю.

Пункт меню 12 – Проведение сравнительного анализа различных методов и алгоритмов сортировок:

1. Программа считывает таблицу из файла с данными для тестирования и проверяет их корректность.
2. Программа замеряет время сортировки таблицы структур и соответствующей таблицы ключей сортировкой пузырьком и быстрой сортировкой.
3. Программа печатает информацию о средней скорости сортировок и относительном выигрыше времени.

## Основные функции:

Функция для обработки выбранного пункта меню. Принимает указатели на таблицу структур и ключей и номер выбранного пункта меню. Возвращает флаг необходимости завершения программы

```
bool process_menu(menu_item_t menu_item, table_t *table,  
key_table_t *key_table)
```

Функция чтения таблицы из файла. Принимает указатель на таблицу структур

```
void read_table(table_t *table)
```

Функция сохранения таблицы в файл. Принимает указатель на таблицу структур

```
void save_table(const table_t *table);
```

Функция обновления таблицы ключей. Принимает указатели на таблицу структур и ключей

```
void create_key_table(const table_t *table, key_table_t  
*key_table);
```

Функция печати таблицы структур в соответствии с отсортированной таблицей ключей. Принимает указатели на таблицу структур и ключей

```
static void print_table_with_key_table(const table_t *table,  
key_table_t *key_table);
```

Функция печати таблицы ключей. Принимает указатель на таблицу ключей

```
void print_key_table(key_table_t *key_table);
```

Функция удаления записи из таблицы структур. Принимает указатель на таблицу структур

```
void delete_record_from_table(table_t *table);
```

Функция добавления записи в таблицу структур. Принимает указатель на таблицу структур

```
void add_record_to_table(table_t *table);
```

Функция печати друзей с днем рождения в ближайшую неделю.  
Принимает указатель на таблицу структур

```
void print_friends_with_bday(const table_t *table);
```

**Функция анализа различных методов и алгоритмов сортировок**

```
void run_analysis(void);
```

## Набор тестов

### Обработка пунктов меню

Описание теста	Вход	Результат
Неположительное число	0	Для выбора пункта меню введите целое число от 1 до 12
Число больше 12	13	Для выбора пункта меню введите целое число от 1 до 12
Максимальный пункт меню	12	[сведения о скорости сортировок]
Минимальный пункт меню	1	[выход из программы]
Символ вместо числа	a	Для выбора пункта меню введите целое число от 1 до 12
Выбор пунктов меню, которые требуют данные в таблице при пустой таблице	4	Нельзя производить данное действие над пустой таблицей! Прочтите таблицу из файла или добавьте записи вручную!

### Чтение файла

Описание теста	Ввод	Содержимое файла	Результат
Не существующий файл	2 not_exist.txt	-	Введите корректное имя файла!
Слишком длинное имя файла	2 hsdgfiwegbiesgbc hsbchsbchsbchsd bcihdsbchsdcbcdsh cbsdicbskcbsdcb.t	-	Ошибка ввода! Введите имя файла не длиннее 30 символов!

	xt		
Пустой файл	2 empty.txt	-	Пустой файл!
Файл с 1 структурой	2 table.txt	Zharinov Mikhail 8966160243 Kasatkina 0 09 02 2006	Таблица успешно считана
Файл с 1000 структурами	2 big_table.txt	[1000 корректных структур]	Таблица успешно считана
Файл с некорректной структурой	2 t.txt	Divaev Alexander +876875 Novatorov 19 0b 22 8 2005	Некорректная структура файла!

### Чтение полей

Описание теста	Ввод	Вывод
Слишком длинный номер	7961160319888	Введите номер не длиннее 11 символов!
Невозможный день	32	Введите день от 1 до 31!
Невозможный месяц	13	Введите месяц от 1 до



		Novatorov 19 0b 22 8 2005	таблицы
Заполненная таблица	[1000 записей]	Divaev Alexander +876875 Novatorov 19 0b 22 8 2005	Таблица уже максимального размера!

### Результаты сравнения

Для анализа я измерил средние скорости сортировки при помощи сортировки пузырьком и быстрой сортировки qsort таблицы записей и ключей при размерах 40, 200, 500 и 1000 элементов и получил следующие результаты (в наносекундах):

Размер	Сортировка таблицы пузырьком	Сортировка таблицы ключей пузырьком	Сортировка таблицы с помощью qsort	Сортировка таблицы ключей с помощью qsort
40	9250	6034	1416	1362
200	219889	135083	9446	9662
500	1377770	842814	31031	29817
1000	5480962	3387698	68246	64835

Выигрыши по времени составили:

Размер	Выигрыш от сортировки таблицы ключей при сортировке пузырьком(1- 2)	Выигрыш от сортировки таблицы ключей при сортировке qsort(3-4)	Выигрыш от qsort при сортировке таблицы(1-3)	Выигрыш от qsort при сортировке таблицы ключей(2-4)
40	34.77%	3.81%	84.69%	77.43%
200	38.57%	-2.29%	95.70%	92.85%
500	38.83%	3.91%	97.75%	96.46%
1000	38.19%	5.00%	98.75%	98.09%

Дополнительные затраты памяти при использовании таблицы ключей составили 23.53%



## Ответы на контрольные вопросы

1. Как выделяется память под вариантную часть записи?

Под вариантную часть выделяется размер памяти, являющийся максимальным из полей объединения. В данном случае максимальное из 12 и 52 байт – это 52 байта. Следовательно, использование объединения снижает затраты памяти на 12 байт на запись

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

Если в вариантную часть ввести данные, несоответствующие описанному статусу записи, то это приведет к перезаписыванию мусором соответствующих статусу вариантных полей записи

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

За правильностью выполнения операций с вариантной частью записи должен следить программист, реализующий такие операции

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

Таблица ключей представляет собой массив пар ключ-индекс в исходной таблице. Она позволяет ускорить операции, которые требуют перемещение элементов таблицы, например сортировку.

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

Использовать таблицу ключей эффективнее тогда, когда размер ее элементов значительно меньше элементов самой таблицы

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

Для обработки таблиц предпочтительнее использовать быстрые сортировки, так как это позволяет ускорить сортировку. Если размер ключа значительно меньше размера всей записи, то имеет смысл сортировать таблицу ключей, а не саму таблицу.

## **Вывод**

При выполнении над большими таблицами таких операций, как сортировка, стоит использовать сортировку не всей таблицы, а только таблицы ключей. Это позволяет уменьшить время сортировки примерно на 40% при использовании сортировки пузырьком и на 5% при использовании быстрой сортировки. Данный эффект достигается за счет того, что при сортировке обмениваются элементы меньшего размера. При использовании данной по варианту структуре записи, это приводит к дополнительным затратам памяти в 23.5%.