

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт»
Национальный Исследовательский Университет

Факультет №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

КУРСОВОЙ ПРОЕКТ

По дисциплине «Практикум на ЭВМ»

Задание: «Сортировка и поиск»

Студент: Хренникова А. С.

Группа: М80-108-19

Преподаватель: Поповкин А. В.

Подпись:

Оценка:

Дата:

Содержание

Задача.....	3
Общие сведения о линейном выборе с обменом	4
Общие сведения о программе	5
Общий метод решения.....	6
Описание переменных, функций	7
Входные и выходные данные.....	10
Входные данные	10
Листинг программы	11
Пример работы программы.....	16
Заключение	22
Список использованных источников	23

Задача

Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице.

Программа должна вводить значения элементов неупорядоченной таблицы и проверять работу процедуры сортировки в трёх случаях: (1) элементы таблицы с самого начала упорядочены; (2) элементы таблицы расставлены в обратном порядке; (3) элементы таблицы не упорядочены. В последнем случае можно использовать встроенные процедуры генерации псевдослучайных чисел.

Для каждого вызова процедуры сортировки необходимо печатать исходное состояние таблицы и результаты сортировки. После выполнения сортировки программа должна вводить ключи и для каждого из них выполнять поиск в упорядоченной таблице методом дихотомии (с помощью процедуры двоичного поиска) и печатать найденные элементы, если они присутствуют в таблице.

В процессе отладки и тестирования рекомендуется использовать команды обработки текстовых файлов ОС UNIX и переадресацию ввода-вывода. Тестовые данные необходимо заранее поместить в текстовые файлы. В качестве текста для записей таблицы взять фрагмент стихотворения. Каждый элемент таблицы, содержащий ключ и текст записи, распечатывать в отдельной строке.

Вариант метода сортировки (в терминах Н. Вирта): **линейный выбор с обменом**

Вариант структуры таблицы: тип ключа – **строковый**, размер ключа – **5 байт**; хранение данных и ключей – **отдельно**; число элементов таблицы – **8-12**.

Общие сведения о линейном выборе с обменом

Линейный выбор

Метод предполагает использование рабочего массива, в который помещается отсортированный массив. Количество просмотров определяется количеством элементов массива. Сортировка посредством линейного выбора сводится к следующему:

- 1) Найти наименьший элемент, переслать его в рабочий массив и заменить его в исходном массиве величиной, которая больше любого реального элемента.
- 2) Повторить шаг (1). На этот раз будет выбран наименьший из оставшихся элементов.
- 3) Повторять шаг (1) до тех пор, пока не будут выбраны все n элементов.

Линейный выбор с обменом

В этом методе рабочий массив не используется. Общая схема алгоритма следующая. Просмотрим элементы a_1, a_2, \dots, a_n найдем среди них минимальный элемент и переставим его на первое место. Теперь рассмотрим элементы a_2, \dots, a_n найдем среди них минимальный элемент и поставим его на второе место. И так до тех пор, пока не останется один элемент.

Стандартный обмен (метод "пузырька")

Просматриваем элементы a_1, a_2, \dots, a_n и попутно меняем местами те соседние элементы, для которых выполнено неравенство $a_i > a_{i+1}$. В результате первого просмотра максимальный элемент станет последним ("он вниз - пузыри вверх"). На следующем просмотре аналогичную процедуру проведем над элементами a_1, a_2, \dots, a_{n-1} — и т.д. Сортировку необходимо закончить, если будет выполнено одно из двух условий:

- 1) После очередного прохода не сделано ни одного обмена;
- 2) Сделан проход для элементов a_1, a_2 .

Общие сведения о программе

Необходимое программное и аппаратное обеспечение: ОС семейства UNIX (Linux Ubuntu), среда программирования Си (язык Си, компилятор gcc), процессор с 64-битной архитектурой (как на лабораторном компьютере).

Система программирования: GUN C.

Местонахождение файлов на домашнем компьютере: /home/lina_tucha/dir/kp8/kurs91.c. Сам файл компилируется с помощью написания «gcc kurs91.c -o 345» в командной строке интерпретатора команд.

Общий метод решения

Программа по команде должна выводить меню и выполнять все его пункты, в зависимости от запроса, выдавать результат в интерактивном режиме. Список возможных действий включает в себя: чтение файла, печать таблицы (ключи и соответствующие им строки, разделяемые вертикальной чертой), перестановка элементов в обратном порядке, перестановка элементов в случайном порядке, сортировка методом линейного выбора с обменом, поиск элементов таблицы по заданному ключу (выводит на экран все элементы с такими ключами).

Описание переменных, функций

Структура

Таблица 1 - Описание key – элемент массива

Имя	Тип	Назначение
key[6]	char	Ключ строки
string	char*	Строка изображения

Переменные и константы подпрограмм

Таблица 2 – Описание void print(key keys[], int len) - печать таблицы

Имя	Тип	Вид	Назначение
keys[]	key	Входной параметр	Массив
len	int	Входной параметр	Размер массива
i	int	Локальная переменная	Переменная цикла

Таблица 3 – Описание void reverse(key keys[], int len) – перестановка элементов таблицы в обратном порядке

Имя	Тип	Вид	Назначение
keys[]	key	Входной параметр	Массив
len	int	Входной параметр	Размер массива
i	int	Локальная переменная	Переменная цикла
temp	key	Локальная переменная	Переменная, используемая для сортировки пузырьком

void usage() – выводит меню на экран

Таблица 4 – Описание void swap(key *a, key *b) – меняет два элемента местами

Имя	Тип	Вид	Назначение
a	key*	Входной параметр	Элемент массива
b	key*	Входной параметр	Элемент массива
t	key	Локальная переменная	Переменная, для замены элемента массива

Таблица 5 – Описание void binary_search(key keys[], int len, char *target) – поиск элемента по ключу

Имя	Тип	Вид	Назначение
keys[]	key	Входной параметр	Массив
len	int	Входной параметр	Размер массива
target	char*	Входной параметр	Искомый ключ
first	int	Локальная переменная	Левая граница поиска (индекс)
last	int	Локальная переменная	Правая граница поиска (индекс)
mid	int	Локальная переменная	Индекс срединного элемента

Таблица 6 – Описание void sort(key array[], int len) – сортировка таблицы

Имя	Тип	Вид	Назначение
array[]	key	Входной параметр	Массив
len	int	Входной параметр	Размер массива
i	int	Локальная переменная	Переменная цикла
j	int	Локальная переменная	Переменная цикла
ind	int	Локальная переменная	Массив

Таблица 7 - Описание void deorder(key keys[], int len) – перемешивание элементов таблицы

Имя	Тип	Вид	Назначение
keys[]	key	Входной параметр	Массив
len	int	Входной параметр	Размер массива
i	int	Локальная переменная	Переменная цикла
temp	key	Локальная переменная	Переменная, используемая для сортировки пузырьком
r	int	Локальная переменная	Сгенерированное число($0 < r < len$)

Таблица 8 – Описание `int eq<code>(char *lhs, char *rhs)` – сравнение ключей(первый равен второму\первый меньше второго\первый больше второго)

Имя	Тип	Вид	Назначение
lhs	char*	Входной параметр	Ключ элемента массива
rhs	char*	Входной параметр	Искомый ключ
i	int	Локальная переменная	Переменная цикла

Таблица 9 – Описание переменных и констант функции `main`

Имя	Тип	Назначение
keys[]	key	Массив ключей
string[] []	char	Массив строк изображения
n	int	Количество строк таблицы
exist	int	Характеристика файла: открыт ли файл
sorted	int	Характеристика таблицы: отсортирована ли
table	FILE*	Ссылка на входной файл
cmd[]	char	Вводимая с клавиатуры команда
name[]	char	Имя входного файла
inp	char	Ключ, по которому ведётся поиск в таблице
i	int	Индекс искомого ключа

Входные и выходные данные

Входные данные

Входные данные должны быть представлены текстовыми файлами в которых каждая строка это: ключ, далее пробел (рекомендуется, но можно и без него), и остальная часть строки – текстовые данные. В конце каждой строки – перевод строки (рассматривается как признак её окончания).

Названия файлов должны быть на латинице и без пробелов. При вводе команд с клавиатуры название файла пишется полностью, с расширением. Файлы должны лежать в том же каталоге, что и выполняемая программа.

Выходные данные

Выходные данные для программы –результаты выполнения пунктов меню, а именно: программа по команде печатает: таблицу в виде ключ-вертикальная_черта-строка_текста, найденные при поиске по ключу элементы (в том же формате). Также программа выводит на экран пригласительные и информационные сообщения.

Листинг программы

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct
{
    char key[6];
    char *string;
} key;

void usage()
{
    printf("Usage:\n\to - open file\n\tp - print\n\ts - sort\n\td - deorder\n\tr - reverse\n\tb - binary search\n\th
- help\n\tq - quit\n");
}

void print(key keys[], int len)
{
    printf("-----\n");
    for (int i = 0; i < len; i++)
    {
        printf("|%5s| %-65s\n", keys[i].key, keys[i].string);
    }
    printf("-----\n");
}

void deorder(key keys[], int len)
{
    key temp;
    int r;
    for (int i = 0; i < len; i++)
    {
        r = rand() % len;
        temp = keys[r];
        keys[r] = keys[i];
        keys[i] = temp;
    }
}

void reverse(key keys[], int len)
{
    key temp;
    for (int i = 0; i < len / 2; i++)
    {
        temp = keys[len - i - 1];
        keys[len - i - 1] = keys[i];
        keys[i] = temp;
    }
}

int eq(char *lhs, char *rhs)
{
    for (int i = 0; i < 5; i++)
    {
        if (lhs[i] != rhs[i])
```

```

        {
            return 0;
        }
    }
    return 1;
}

int lt(char *lhs, char *rhs)
{
    for (int i = 0; i < 5; i++)
    {
        if (lhs[i] < rhs[i])
        {
            return 1;
        }
        if (lhs[i] > rhs[i])
        {
            return 0;
        }
    }
    return 0;
}

int gt(char *lhs, char *rhs)
{
    for (int i = 0; i < 5; i++)
    {
        if (lhs[i] > rhs[i])
        {
            return 1;
        }
        if (lhs[i] < rhs[i])
        {
            return 0;
        }
    }
    return 0;
}

int binary_search(key keys[], int len, char *target)
{
    int first, last, mid;
    first = 0;
    last = len - 1;
    mid = (first + last) / 2;
    while (first <= last)
    {
        if (lt(keys[mid].key, target))
        {
            first = mid + 1;
        }
        else if (eq(keys[mid].key, target))
        {
            return mid;
        }
        if (eq(keys[mid+1].key, target))
            return mid+1;
        else continue;
    }
}

```

```

    }
    else
    {
        last = mid - 1;
    }
    mid = (first + last) / 2;
}
return -1;
}

void swap(key *a, key *b)
{
    key t = *a;
    *a = *b;
    *b = t;
}

void sort(key array[], int len)
{
    for (int i = 0; i < len; i++)
    {
        int ind = i;
        for (int j = i + 1; j < len; j++)
        {
            if (lt(array[j].key, array[ind].key))
            {
                ind = j;
            }
        }
        swap(&array[i], &array[ind]);
    }
}

int main()
{
    FILE *table;
    int exist = 0;
    int sorted = 1;
    char cmd[2];
    char name[40];

    int n;
    key keys[40];
    char strings[40][65];

    usage();
    while (1)
    {
        printf("> ");
        scanf("%s", cmd);

        if (!strcmp(cmd, "o"))
        {
            printf("Enter file name: ");
            scanf("%s", name);
            // strcpy(name, "test1.txt");
            table = fopen(name, "r");

```

```

char t;
int is_eof = 0;
n = 0;
while (fscanf(table, "%s %[^\\n]", keys[n].key, strings[n]) == 2)
{
    keys[n].string = strings[n];
    n++;
}
exist = 1;
sorted = 1;
fclose(table);
printf("...done!\\n");
}
else if (!strcmp(cmd, "p"))
{
    if (exist)
    {
        print(keys, n);
    }
    else
    {
        printf("Table wasn't opened!\\n");
    }
}
else if (!strcmp(cmd, "s"))
{
    if (exist)
    {
        sort(keys, n);
        sorted = 1;
        printf("...done!\\n");
    }
    else
    {
        printf("Table wasn't opened!\\n");
    }
}
else if (!strcmp(cmd, "d"))
{
    if (exist)
    {
        deorder(keys, n);
        sorted = 0;
        printf("...done!\\n");
    }
    else
    {
        printf("Table wasn't opened!\\n");
    }
}
else if (!strcmp(cmd, "r"))
{
    if (exist)
    {
        reverse(keys, n);
        sorted = 0;
        printf("...done!\\n");
    }
}

```

```

    }
    else
    {
        printf("Table wasn't opened!\n");
    }
}
else if (!strcmp(cmd, "b"))
{
    if (exist)
    {
        if (sorted)
        {
            int num;
            char chr;
            char inp[6];
            printf("Enter the key: ");
            scanf("%s", inp);
            int i = binary_search(keys, n, inp);
            if (i == -1)
            {
                printf("Not found!\n");
            }
            else
            {
                printf("%s\n", keys[i].string);
            }
        }
        else
        {
            printf("Table isn't sorted!\n");
        }
    }
    else
    {
        printf("Table wasn't opened!\n");
    }
}
else if (!strcmp(cmd, "h"))
{
    usage();
}
else if (!strcmp(cmd, "q"))
{
    return 0;
}
else
{
    printf("Unknown command!\n");
}
}
return 0;
}

```

Пример работы программы

```
lina_tucha@LAPTOP-44CRFC1U:~/dir/kp9$ gcc kurs91.c -o 345
```

```
lina_tucha@LAPTOP-44CRFC1U:~/dir/kp9$ ./345
```

```
Usage:
```

```
o - open file
p - print
s - sort
d - deorder
r - reverse
b - binary search
h - help
q - quit
```

```
> o
```

```
Enter file name: test7.txt
```

```
...done!
```

```
> p
```

```
-----
|asfgv| --- {-----} | |
|etghh| --- | {-----} |
|ethdg| --- {---} {---} |
|etpgl| --- {---} |
|rghjj| --- {---} {---} |
|ropgg| | {-----} {---} |
|rplgk| | {-----} {---} |
|skgpd| | {-----} {---} |
|tpkbd| | {-----} {---} |
|yvoog| | {-----} {---} |
|zgggh| {-----} |
-----
```

```
> d
```

```
...done!
```

```
> p
```

```
-----
|ethdg| --- {---} {---} | |
|yvoog| | {-----} {---} |
|etpgl| --- {---} |
|tpkbd| | {-----} {---} |
|ropgg| | {-----} {---} |
|zgggh| {-----} |
|skgpd| | {-----} {---} |
|rplgk| | {-----} {---} |
|etghh| --- | {-----} |
|rghjj| --- {---} {---} |
|asfgv| --- {---} {---} |
-----
```

```
> s
```

```
...done!
```

```
> p
```

```
-----
|asfgv| --- {---} {---} |
|etghh| --- | {-----} |
-----
```



```

|ethdg| .....|
|etpgl| .....|
|rghjj| .....|
|ropgg| .....|
|rplgk| .....|
|skgpd| .....|
|tppbd| .....|
|yvoog| .....|
|zgggh| .....|

```

```

> r
...done!
> p

```

```

|zgggh| .....|
|yvoog| .....|
|tppbd| .....|
|skgpd| .....|
|rplgk| .....|
|ropgg| .....|
|rghjj| .....|
|etpgl| .....|
|ethdg| .....|
|etghh| .....|
|asfgv| .....|

```

```

> s
...done!
> b
Enter the key: skgpd
| .....|
> o
Enter file name: test8.txt
...done!
> p

```

```

|aaaaa| .....|
|aaaab| .....|
|aaaac| .....|
|aaaad| .....|
|aaaae| .....|
|aaaaf| .....|
|aaaag| .....|
|aaaah| .....|
|aaaai| .....|

```

```

> b
Enter the key: aaaad
| .....|
> o
Enter file name: test3.txt
...done!
> p

```

aaaaa	_____	\$\$\$	
aaaaa	_____	\$ _ \$	
aaaaa	_____	\$\$\$	
aaaab	_____	\$ _ \$	
aaaca	_____	\$ _ \$	
aaaia	_____	\$\$\$_\$\$\$	
frjgn	_____	\$\$ _\$\$\$ _\$\$\$	
fuosj	_____	\$\$ _\$\$\$\$\$\$\$ _ \$	
gaklk	_____	\$ _____ \$	
gbkff	_____	\$ _____ \$	
gbkfg	_____	\$ _____ \$	
ggpef	_____	\$ _\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
heoif	_____	\$ _ \$ _____ \$	
iffgj	_____	\$ _ \$ _\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
irgeg	_____	\$ _ \$ _ \$ _____ \$\$\$	
jefpg	_____	\$ _ \$ _ \$ _\$\$\$ _\$\$\$ _\$	
jonrg	_____	\$ _ \$ _ \$ _\$\$\$ _\$\$\$ _\$	
jtihp	_____	\$ _ \$ _ \$ _____ \$\$\$	
klrig	_____	\$ _ \$ _\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
klyod	_____	\$ _\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
kscst	_____	\$ _____ \$	
kwopg	_____	\$ _\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
lolol	_____	\$ _ \$ _ \$ _ \$ _ \$	
loool	_____	\$ _\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
loooo	_____	\$ _ \$ _ \$ _ \$ _ \$	
ooooo	_____	\$ _\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
pfrgg	_____	\$\$\$ _____ \$\$\$	
pofjg	_____	\$\$ _\$\$\$ _____\$\$\$\$ _\$	
ptkrf	_____	\$\$ _____\$\$\$\$\$\$\$\$ _\$\$\$	
pwdkq	_____	\$ _\$\$ _____\$\$\$\$ _\$	
rpvbg	_____	\$\$\$\$\$ _\$\$\$\$\$\$\$\$ _\$\$\$\$\$\$\$ _\$ \$	
soggo	_____	\$ _\$\$\$ _\$\$\$\$\$\$\$ _\$ \$\$	
tpogh	_____	\$ _ \$ _ \$ _____ \$ _ \$	
tthpl	_____	\$ _ \$ _ \$ _\$\$\$\$\$\$\$\$\$\$\$\$ _\$ _\$	
tthpm	_____	\$\$\$ _\$ _\$ _\$\$\$\$\$\$\$\$\$\$\$\$ _\$ _\$ _\$	
tthpm	_____	\$ _\$\$\$\$ _\$ _\$ _____ \$ _\$ _\$	
tthpn	_____	\$\$\$ _\$ _\$ _\$ _____\$\$ _\$ _\$	
wwlfl	_____	\$ _ \$ _ \$ _\$ _____ \$ _\$\$\$ _\$	
wwlfl	_____	\$ _\$\$\$ _\$ _\$ _____\$\$\$ _\$ _\$ _\$	

> r
...done!
> p

wwlfl	\$ _\$\$\$ _\$ _\$ _____	\$\$\$ _\$ _\$ _\$
wwlfl	\$ _\$ _\$ _\$ _____	\$ _\$\$\$ _\$
tthpn	\$\$\$ _\$ _\$ _\$ _____	\$\$\$ _\$ _\$
tthpm	\$ _\$\$\$\$ _\$ _\$ _____	\$ _\$ _\$ _\$
tthpm	\$\$\$ _\$ _\$ _\$ _\$\$\$\$\$\$\$\$\$\$\$\$	\$ _\$ _\$ _\$
tthpl	\$ _\$ _\$ _\$ _\$\$\$\$\$\$\$\$\$\$\$\$	\$ _\$ _\$
tpogh	\$ _\$ _\$ _\$ _____	\$ _\$ _\$
soggo	\$ _\$\$\$ _\$ _\$ _\$\$\$\$\$\$\$	\$ _\$
rpvbg	\$\$\$ _\$ _\$ _\$ _\$\$\$\$\$\$\$	\$ _\$
pwdkq	\$ _\$\$\$ _\$ _____	\$\$\$ _\$
ptkrf	\$\$\$ _\$ _\$ _\$ _\$\$\$\$\$\$\$\$	\$\$\$
pofjg	\$\$\$ _\$ _\$ _\$ _\$\$\$\$\$\$\$	\$\$\$
pfrgg	\$\$\$ _\$ _\$ _\$ _____	\$\$\$

ooooo	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
loooo	_____	\$	_____	\$	_____
loool	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
lolol	_____	\$	_____	\$	_____
kwopg	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
kscst	_____	\$	_____	\$	
klyod	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
klig	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
jtihp	_____	\$	_____	\$	_____
jonrg	_____	\$	_____	\$	_____
jefpg	_____	\$	_____	\$	_____
irgeg	_____	\$	_____	\$	_____
iffgj	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
heoif	_____	\$	_____	\$	
ggpef	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
gbkfg	_____	\$	_____	\$	
gbkff	_____	\$	_____	\$	
gaklk	_____	\$	_____	\$	
fuosj	_____	\$	_____	\$\$\$\$\$\$\$\$	
frjgn	_____	\$	_____	\$\$\$\$	
aaaia	_____	\$	_____	\$\$\$	
aaaca	_____	\$	_____	\$	
aaaab	_____	\$	_____	\$	
aaaaa	_____	\$	_____	\$\$\$	
aaaaa	_____	\$	_____	\$	
aaaaa	_____	\$	_____	\$\$\$	

> b

Table isn't sorted!

Enter the key: klyod

_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	
-------	----	-------	----------------------------------	--

> s

...done!

> d

...done!

> p

jonrg	_____	\$	_____	\$	_____	\$\$\$	_____	\$\$\$	_____	\$\$	
aaaaa	_____	\$	_____	\$\$\$							
jefpg	_____	\$	_____	\$	_____	\$\$\$	_____	\$\$\$	_____	\$\$	
aaaaa	_____	\$	_____	\$							
klig	_____	\$	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$					
klyod	_____	\$	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$					
tthpn	\$\$\$	_____	\$	_____	\$\$\$	_____	\$	_____	\$\$\$	_____	\$
gbkfg	_____	\$	_____	\$							
fuosj	_____	\$	_____	\$\$\$	_____	\$\$\$	_____	\$			
lolol	_____	\$	_____	\$	_____	\$	_____	\$	_____	\$	
ooooo	_____	\$	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$							
wwlfl	\$	_____	\$\$\$	_____	\$	_____	\$\$\$	_____	\$	_____	\$
aaaia	_____	\$	_____	\$\$\$	_____	\$\$\$					
pwdkq	_____	\$	_____	\$\$\$	_____	\$\$\$	_____	\$\$\$	_____	\$\$\$	
tpogh	_____	\$	_____	\$	_____	\$	_____	\$	_____	\$	
aaaab	_____	\$	_____	\$							
aaaaa	_____	\$	_____	\$\$\$							
ptkrf	_____	\$	_____	\$\$\$	_____	\$\$\$	_____	\$\$\$			
pfrgg	_____	\$	_____	\$\$\$	_____	\$\$\$					
tthpm	\$	_____	\$\$\$	_____	\$	_____	\$	_____	\$\$\$	_____	\$

```

|ggpef| _____ $ _____ $$$$$$$$$$$$$$ |
|pofjg| _____ $$ _____ $$$ _____ $$$$ _____ $$ |
|kwopg| _____ $ _____ $$$$$$$$$$$$$$ |
|irgeg| _____ $ _____ $ _____ $ _____ $$$ |
|loooo| _____ $ _____ $ _____ $ _____ $ _____ $ |
|jtihp| _____ $ _____ $ _____ $ _____ $$$ |
|frjgn| _____ $$ _____ $$$ _____ $$$ |
|iffgj| _____ $ _____ $ _____ $$$$$$$$$$$$$$ |
|gaklk| _____ $ _____ $ |
|kscst| _____ $ _____ $ |
|rpvbg| _____ $$$$ _____ $$$$$$$$ _____ $$$$ _____ $ _____ $ |
|tthpm| _____ $$$ _____ $ _____ $ _____ $$$$$$$$$$$$$$ _____ $ _____ $ _____ $ |
|gbkff| _____ $ _____ $ |
|wwlfl| _____ $ _____ $ _____ $ _____ $ _____ $ _____ $$$ _____ $ |
|heoif| _____ $ _____ $ _____ $ |
|loool| _____ $ _____ $$$$$$$$$$$$$$ |
|soggo| _____ $ _____ $ _____ $ _____ $$$$ _____ $ _____ $ |
|tthpl| _____ $ _____ $ _____ $ _____ $$$$$$$$$$$$$$ _____ $ _____ $ _____ $ |
|aaaca| _____ $ _____ $ |

```

```

-----
> s
...done!
> b
Enter the key: frjgn

```

```

_____ $$ _____ $$$ _____ $$$
> p

```

```

-----
|aaaaa| _____ $$$ |
|aaaaa| _____ $ _____ $ |
|aaaaa| _____ $$$ |
|aaaab| _____ $ _____ $ |
|aaaca| _____ $ _____ $ |
|aaaia| _____ $$$ _____ $$$ |
|frjgn| _____ $$ _____ $$$ _____ $$$ |
|fuosj| _____ $$ _____ $$$$ _____ $ |
|gaklk| _____ $ _____ $ |
|gbkff| _____ $ _____ $ |
|gbkfg| _____ $ _____ $ |
|ggpef| _____ $ _____ $$$$$$$$$$$$$$ |
|heoif| _____ $ _____ $ _____ $ |
|iffgj| _____ $ _____ $ _____ $$$$$$$$$$$$$$ |
|irgeg| _____ $ _____ $ _____ $ _____ $$$ |
|jefpg| _____ $ _____ $ _____ $ _____ $$$ _____ $$$ |
|jonrg| _____ $ _____ $ _____ $ _____ $$$ _____ $$$ |
|jtihp| _____ $ _____ $ _____ $ _____ $$$ |
|klrig| _____ $ _____ $ _____ $$$$$$$$$$$$$$ |
|klyod| _____ $ _____ $$$$$$$$$$$$$$$$$$ |
|kscst| _____ $ _____ $ |
|kwopg| _____ $ _____ $$$$$$$$$$$$$$$$$$ |
|lolol| _____ $ _____ $ _____ $ _____ $ |
|loool| _____ $ _____ $$$$$$$$$$$$$$$$$$ |
|loooo| _____ $ _____ $ _____ $ _____ $ |
|ooooo| _____ $ _____ $$$$$$$$$$$$$$$$$$ |
|pfrgg| _____ $$$ _____ $$$ |
|pofjg| _____ $$ _____ $$$ _____ $$$$ _____ $$ |
|ptkrf| _____ $$ _____ $$$$$$$$$$ _____ $$$ |
|pwdkq| _____ $ _____ $ _____ $$$$ _____ $ |

```

rpvbg	_____	\$\$\$\$\$	_____	\$\$\$\$\$\$\$\$\$	_____	\$\$\$\$\$\$\$\$\$	_____	\$	\$	
soggo	__	\$	_____	\$	_____	\$\$\$\$\$	_____	\$	\$	
tpogh	_	\$	_____	\$	_____	_____	_____	\$	\$	
tthpl	_	\$	_____	\$	_____	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$	\$	\$	
tthpm	\$	_____	\$\$\$\$\$	_____	\$	_____	_____	\$	\$	
tthpm	_	\$\$\$	_____	\$	_____	_____	\$\$\$\$\$\$\$\$\$\$\$\$\$	\$	\$	
tthpn	\$\$\$	_____	\$	_____	\$\$\$	_____	_____	\$	\$	
wwlfl	\$	_____	\$	_____	\$	_____	_____	\$	\$	
wwlfl	\$	_____	\$\$\$	_____	\$	_____	_____	\$\$\$	\$	

>q

Заключение

Цель задания достигнута – реализована программа, считывающая строки текста из файла и выполняющая их сортировку по ключам. Есть возможность получения по ключу нужных строк из отсортированной таблицы – используя метод дихотомии. Проверена устойчивость сортировки линейным выбором с обменом – данная сортировка не является устойчивой (переставляет элементы с одинаковым ключом в порядке, отличном от их изначального хронологического расположения). Асимптотика: $O(n^2)$ в лучшем, среднем и худшем случае.

Список использованных источников

1. РосДиплом, Оформление таблиц в дипломной работе, особенности и требования ГОСТ/Электронный диплом/Режим доступа: <https://www.rosdiplom.ru/rd/pubdiplom/view.aspx?id=288>
2. Диплом Журнал, Оформление курсовой работы по ГОСТу 2019(образец)/Электронный диплом/Режим доступа: <https://journal.duplom.ru/kurovaya/oformlenie-kurovoj-raboty-po-gostu-2019-obrazec/>
3. Vyuchit.work – универсальная методичка/Электронный диплом/Режим доступа: <https://vyuchit.work/samorazvitie/sekretyi/oformlenie-risunkov-po-gostu.html>
4. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. :Пер. с англ. – М. : Издательский дом «Вильямс», 2009. – 304 с. : ил. – Парал. тит. англ.
5. Студопедия/Электронный диплом/Режим доступа: https://studopedia.ru/3_80780_lineyniy-vibor-s-obmenom.html