

## Задание 9: Сортировка и поиск

### Цель работы

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

### Задание

Метод сортировки: метод простой вставки

Структура таблицы: тип ключа — вещественный, длина ключа в байтах — 4, хранение данных и ключей — отдельно, число элементов таблицы — 8-10.

### Алгоритм работы программы

#### Чтение файлов:

У пользователя запрашивается сначала имя файла с ключами, затем имя файла со значениями. Если файл не открывается, то программа выводит соответствующее сообщение и прекращает работу. Файлы открываются для чтения, после чего, пробега по файлу с ключами, определяются размер таблицы и максимальная длина хранимой строки. Затем считывается файл с ключами, если ключей недостаточно, то программа выводит соответствующее сообщение пользователю и прекращает работу. После всего этого программа считывает оба файла в массив структур, содержащих поля ключ и значение.

#### Перемешивание:

При перемешивании для каждой пары ключ-значение из массива определяется индекс пары, с которой её необходимо поменять местами, таким образом каждая пара гарантировано переместится по массиву на случайную позицию хотя бы один раз.

#### Реверс:

Пробега по первой половине массива с парами ключ-значение каждый элемент меняется с элементом, находящимся для него симметрично относительно середины массива.

#### Сортировка:

Сортировка вставкой осуществляется следующим образом: при обходе массива, начиная со второго элемента, мы имеем отсортированный подмассив слева от текущего положения и не отсортированный подмассив справа от него, текущий элемент вставляется на нужное место в отсортированном подмассиве, после чего осуществляется переход к следующему элементу. Обход массива начинается со второго элемента, потому что имеющийся подмассив слева состоит из одного элемента, а значит уже отсортирован.

#### Поиск:

В отсортированном массиве, начиная с середины, сравниваем искомый ключ с ключом в данном элементе массива. Если искомый ключ больше, то следует искать в правом подмассиве, если меньше -то в левом, если равен - то значение было найдено. Если при поиске разница в индексах равна единице, то если элемент, находящийся в правой границе подмассива не равен ключу, то такого ключа в массиве нет.

## Функции и структуры

Для хранения пар ключ-значение используется структура record.

char * readname()	Читает строку до переноса строки.
char * readstring(FILE* file, int length, int j)	Читает строку из файла до переноса строки
void InsertionSort(record * table, int TableSize)	Сортировка вставкой
void mix(record * table, int TableSize)	Перемешивание массива
void reverse(record *table, int TableSize)	Реверс массива
void PrintTable(record * table, int TableSize)	Печать таблицы
void bs(record *table, int TableSize)	Бинарный поиск

## Код программы

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

typedef struct record record;

struct record{
    float key;
    char * value;
};

char * readname(){
    int i=0;
    char b=' ', *a=(char*)malloc(sizeof(char)*256);
    scanf("%c",&b);
    while((b!='\n')&&(i!=256)){
        a[i]=b;
        i++;
        scanf("%c",&b);
    }
    a[i]='\0';
    return(a);
}

char * readstring(FILE* file, int length, int j){
    int i=0, eof;
```

```

char b=' ', *a=(char*)malloc(sizeof(char)*(length+1));
eof=fscanf(file,"%c",&b);
do{
    a[i]=b;
    i++;
    eof=fscanf(file,"%c",&b);
}while((b!='\n')&&(i!=length)&&(eof>=0));
a[i]='\0';
return(a);
}

void InsertionSort(record * table, int TableSize){
    record rec;
    int j;
    for (int i=1; i<TableSize; i++){
        rec.key = table[i].key;
        rec.value=table[i].value;
        j = i-1;
        while ((j >= 0) && (table[j].key > rec.key)){
            table[j+1].key=table[j].key;
            table[j+1].value=table[j].value;
            j = j - 1;
        }
        table[j+1].key = rec.key;
        table[j+1].value=rec.value;
    }
}

void mix(record * table, int TableSize){
    int RandPosition;
    record rec;
    for(int i=0; i<TableSize; i++){
        RandPosition=rand()%TableSize;
        table[i].key+=table[RandPosition].key;
        table[RandPosition].key=table[i].key-table[RandPosition].key;
        table[i].key-=table[RandPosition].key;
        rec.value=table[i].value;
        table[i].value=table[RandPosition].value;
        table[RandPosition].value=rec.value;
    }
}

void reverse(record *table, int TableSize){
    record rec;
    for(int i=0; i<TableSize/2; i++){
        table[i].key+=table[TableSize-i-1].key;
        table[TableSize-i-1].key=table[i].key-table[TableSize-i-1].key;
        table[i].key-=table[TableSize-i-1].key;
        rec.value=table[i].value;
        table[i].value=table[TableSize-i-1].value;
        table[TableSize-i-1].value=rec.value;
    }
}

void PrintTable(record * table, int TableSize){
    for(int i=0; i<TableSize; i++){
        printf("%.2f ", table[i].key);
        printf("%s\n", table[i].value);
    }
}

void bs(record *table, int TableSize){

```

```

float key, eps;
char ctrl;
TableSize--;
int check=1, i=TableSize/2;
printf("Enter key:");
scanf("%f",&key);
for(int i=1; i<TableSize; i++){
    if(table[i].key<table[i-1].key) check=0;
}
if(check){
    check=0;
    while(key!=table[i].key){
        printf("hui");
        if(key<table[i].key){
            TableSize=i;
            i=check+(TableSize-check)/2;
        }
        else{
            check=i;
            i=check+(TableSize-check)/2;
        }
        if(TableSize-check==1){
            if(table[TableSize].key!=key){
                check=-1;
                printf("There isn't key %.2f",key);
                break;
            }
            else i=TableSize;
        }
        printf("%d %d %d",check,i,TableSize);
        scanf("%c",&ctrl);
    }
    if(check!=-1) printf("%f %s",table[i].key, table[i].value);
}
else{
    printf("Binary search doesn't work because record isn't sorted");
}
}

int main(){
    char menu=' ', c;
    int eof, j=0, maxstring=0;
    float k;
    int TableSize = 0;
    printf("Enter name of file with keys: ");
    FILE *f1=fopen(readname(),"rt"), *f2;
    if(f1==NULL){
        printf("There is no such file");
        return -1;
    }
    printf("\nEnter name of file with values: ");
    f2=fopen(readname(), "rt");
    if(f2==NULL){
        printf("There is no such file");
        return -1;
    }
    eof=fscanf(f2,"%c", &c);
    while(eof>=0){
        if(c=='\n') {
            j++;
            TableSize++;
            if(j>maxstring) maxstring=j;
        }
    }
}

```

```

        j=0;
    }
    else j++;
    eof=fscanf(f2,"%c", &c);
}
if(c!='\n') TableSize++;
j=0;
eof=fscanf(f1,"%c", &c);
while(eof>=0){
    if(c=='\n') {
        j++;
    }
    eof=fscanf(f1,"%c", &c);
}
if(c!='\n') j++;
if(j<TableSize){
    printf("Too few keys.\nNumber of keys: %d\nNumber of values: %d\n",j,
TableSize);
    return -1;
}
record *table=(record *)malloc(TableSize*(sizeof(record)));
fseek(f1, 0, SEEK_SET);
fseek(f2, 0, SEEK_SET);
for(int i=0; i<TableSize; i++){
    table[i].value=(char *)malloc(maxstring*sizeof(char));
}
for(int i=0; i<TableSize; i++){
    fscanf(f1, "%f", &table[i].key);
    table[i].value=readstring(f2, maxstring, i);
}
fclose(f1);
fclose(f2);
printf("    m - Mix records\n\
r - Reverse records\n\
s - Sort records\n\
p - Print records\n\
b - Binary search by the key\n\
q -Quit\n");
while(menu!='q'){
    if(menu!='\n')printf("Enter operation:");
    scanf("%c", &menu);
    switch(menu){
        case 'm':
            mix(table, TableSize);
            PrintTable(table, TableSize);
            break;

        case 'r':
            reverse(table, TableSize);
            PrintTable(table, TableSize);
            break;

        case 's':
            InsertionSort(table, TableSize);
            PrintTable(table, TableSize);
            break;

        case 'b':
            bs(table, TableSize);
            break;

        case 'p':
            PrintTable(table, TableSize);
            break;

    }
    printf("\n\n");
}

```

}

## Вывод программы

Enter name of file with keys: keys.txt

Enter name of file with values: values.txt

m - Mix records

r - Reverse records

s - Sort records

p - Print records

b - Binary search by the key

q -Quit

Enter operation:p

```

1.00      ,aad8888888888888baa,.
2.00      ,ad8888888888888888888888888888ba,.
3.00      ,ad8888888888888888888888888888888888ba,
4.00      ,ad888888888P""""      """"Y888888888888ba.
5.00      ,d88888888P""      ""Y8888888888ba
6.90      a888888888"      ""Y888888888b,
7.80      ,888888888b,      ""Y88888888b,
7.90      d888888888888b,      "Y8888888b,
8.50      ,8888888" "888888b,      "Y8888888b
8.60      ,888888" "Y88888b,      "Y888888b
11.00     ,888888" "Y88888b,      "888888b
12.00     ,888888" a, 8a"Y88888b,      `888888a
13.00     ,888888" `8, `8) "Y88888b,      ,adPPRg, `888888,
14.00     888888" 8) j8 "Y88888b,      ,ad8888888888b Y888888b
15.00     d88888P ,8' ,8' "Y88888b, ,gPPR888888888888 `888888,
16.90     888888" ,8' ,8' "Y88888b,,ad8"" `Y888888888P )88888)
17.80     888888 8) 8) "Y888888" (8888888"" (88888)
17.90     888888 8, 8, ,ad8Y88888b, d888"" d88888)
18.50     888888 `8, `8, ,ad8"" "Y88888b,,ad8"" 888888)
18.60     888888 `8, `` ,ad8"" "Y88888b" ,888888'
21.00     Y88888, ,gPPR8b ,ad8Y88888b, d888888
22.00     `88888b dP:::Yb ,ad8"" "Y88888b, ,888888P
23.00     888888, 8):::(8 ,ad8"" "Y88888b, d888888'
24.00     `888888, Yb:::,d888"" "Y88888b, d888888P
25.00     Y888888, "8ggg8P" "Y88888b, ,d888888P
26.90     Y88888b, "Y88888b, ,8888888"
27.80     Y88888b, "Y88888b, d8888888"
27.90     Y888888, "Y8888888888888P'
28.50     "888888b, "8888888888"
28.60     Y888888b, ,888888888"
31.00     Y8888888ba, ,a8888888888"
32.00     "Y8888888888ba,_, ,ad8888888888P"
33.00     "Y888888888888888baa,____,aad888888888888""
34.00     "Y8888888888888888888888888888888888"" Normand
35.00     ""Y888888888888888888888888P"" Veilleux
36.00

```

Enter operation:r

[illegible]



Enter operation:s

Enter operation:b

```
34.000000      "Y88888888888888888888888888888888"" Normand
```

There isn't key 34.50

[illegible]





```
33.00      "Y888888888888bbaa,,_____,,aadd888888888888""
28.50      "8888888b,      "88888888888"
13.00      ,888888'   `8, `8) "Y888888b,      ,adPPRg,      `888888,
3.00      ,ad8888888888888888888888888888888888888888888888ba,
35.00      ""Y888888888888888888888888P""      Veilleux
22.00      `888888b      dP:::::Yb      ,ad8""      "Y888888b,      ,888888P
```

Enter operation:b

Enter key:1

Binary search doesn't work because record isn't sorted

Enter operation:q

## Заклучение

В ходе данной работы мной была реализована программа на языке Си для сортировки таблицы по ключу и бинарного поиска в ней.