Московский авиационный институт

(Национальный исследовательскийуниверситет)

Курсовой проект

по курсу

«Языки и методы программирования»

2 семестр

Задание

Выполнил: Тимирчев Янис

Группа: М8О-111Б-21

Руководитель: Никулин С.П.

Оценка:

Дата:

Москва

2022г

Задание

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

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 процедурыгенерации 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 с помощью процедуры двоичного поиска и печатать найденные элементы, если они присутствуют в таблице.

Метод сортировки: Сортировка слиянием

Структура таблицы:

Тип ключа: целый

Длина ключа байтах: 8

Хранение данных и ключей: вместе

Структура проекта.

Проект состоит из 4 файлов:

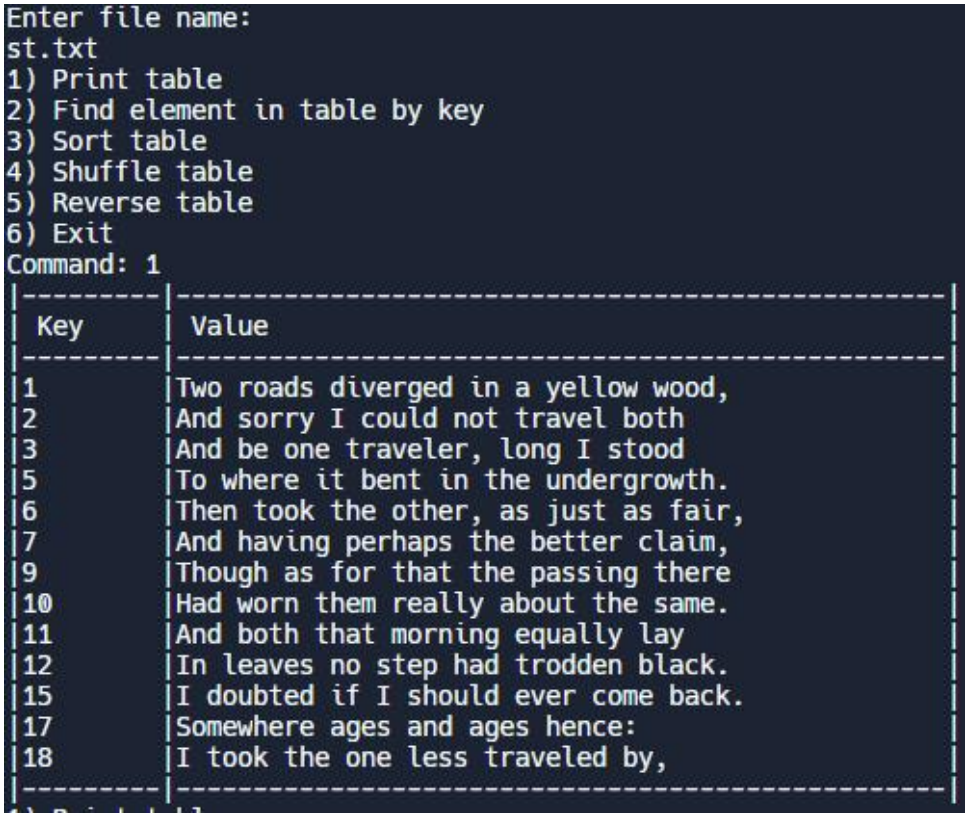
1) **main.c** - Меню, считывание файла.

2) **table.c** - Функции таблицы

3) **table.h** - заголовочный файл с описанием структуры.

4) **st.txt** - входный файл с данными.

Формат программы и примеры работы.



Начальная таблица

```
|---------|-------------------------------------------------|
| Key     | Value                                           |
|---------|-------------------------------------------------|
|1        |Two roads diverged in a yellow wood,             |
|2        |And sorry I could not travel both                |
|3        |And be one traveler, long I stood                |
|5        |To where it bent in the undergrowth.             |
|6        |Then took the other, as just as fair,            |
|7        |And having perhaps the better claim,             |
|9        |Though as for that the passing there             |
|10       |Had worn them really about the same.             |
|11       |And both that morning equally lay                |
|12       |In leaves no step had trodden black.             |
|15       |I doubted if I should ever come back.            |
|17       |Somewhere ages and ages hence:                   |
|18       |I took the one less traveled by,                 |
|---------|-------------------------------------------------|
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 2
Enter key
10
Element found "Had worn them really about the same." by key 10
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 2
Enter key
17
Element found "Somewhere ages and ages hence:  " by key 17
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 2
Enter key
1
Element found "Two roads diverged in a yellow wood," by key 1
```

Нахождение по ключу

Сортировка отсортированной таблицы

```
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 4
Table shuffled
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 1
|---------|-------------------------------------------------
| Key     | Value
|---------|-------------------------------------------------
|10       |Had worn them really about the same.
|2        |And sorry I could not travel both
|3        |And be one traveler, long I stood
|5        |To where it bent in the undergrowth.
|11       |And both that morning equally lay
|15       |I doubted if I should ever come back.
|1        |Two roads diverged in a yellow wood,
|18       |I took the one less traveled by,
|12       |In leaves no step had trodden black.
|7        |And having perhaps the better claim,
|6        |Then took the other, as just as fair,
|17       |Somewhere ages and ages hence:
|9        |Though as for that the passing there
|---------|-------------------------------------------------
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 3
Table sorted
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 1
```

Перемешивание таблицы и её сортировка:

```
Command: 3
Table sorted
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 1
|---------|-------------------------------------------------|
| Key     | Value                                           |
|---------|-------------------------------------------------|
|1        |Two roads diverged in a yellow wood,             |
|2        |And sorry I could not travel both                |
|3        |And be one traveler, long I stood                |
|5        |To where it bent in the undergrowth.             |
|6        |Then took the other, as just as fair,            |
|7        |And having perhaps the better claim,             |
|9        |Though as for that the passing there             |
|10       |Had worn them really about the same.             |
|11       |And both that morning equally lay                |
|12       |In leaves no step had trodden black.             |
|15       |I doubted if I should ever come back.            |
|17       |Somewhere ages and ages hence:                   |
|18       |I took the one less traveled by,                 |
|---------|-------------------------------------------------|
```

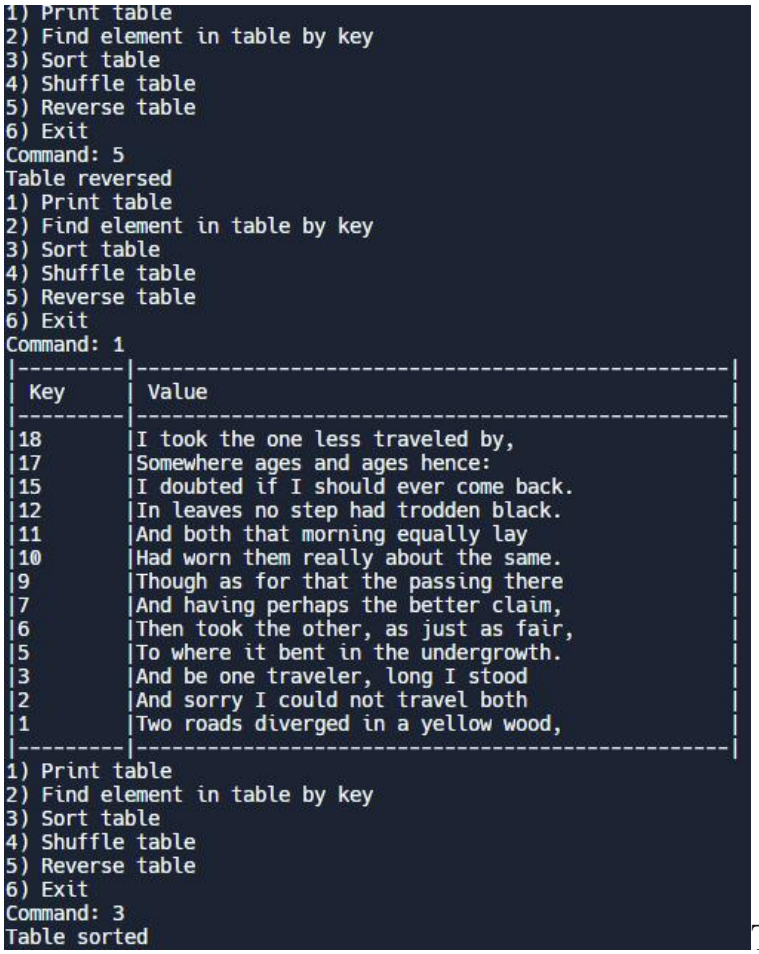


Таблица в обратном порядке

Пример сортировки обратной таблицы и выход из программы

```
Command: 3
Table sorted
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 1
|---------|-----------------------------------------------------|
| Key     | Value                                               |
|---------|-----------------------------------------------------|
|1        |Two roads diverged in a yellow wood,                 |
|2        |And sorry I could not travel both                    |
|3        |And be one traveler, long I stood                    |
|5        |To where it bent in the undergrowth.                 |
|6        |Then took the other, as just as fair,                |
|7        |And having perhaps the better claim,                 |
|9        |Though as for that the passing there                 |
|10       |Had worn them really about the same.                 |
|11       |And both that morning equally lay                    |
|12       |In leaves no step had trodden black.                 |
|15       |I doubted if I should ever come back.                |
|17       |Somewhere ages and ages hence:                       |
|18       |I took the one less traveled by,                     |
|---------|-----------------------------------------------------|
1) Print table
2) Find element in table by key
3) Sort table
4) Shuffle table
5) Reverse table
6) Exit
Command: 6
```

Программный код и текстовые данные.

main.c:

```
#include <stdio.h>
#include "table.h"


void printCommands() {
   printf("1) Print table\n2) Find element in table by key\n3) Sort table\n4) Shuffle
table\n5) Reverse table\n6) Exit\nCommand: ");
}


int main() {
   int i, cnt, rule, key, flag = 1, N=50;
   char ch;
   row arr[N];
   char filename[100];
   printf("Enter file name:\n");
   scanf("%s", filename);
   FILE *file = fopen(filename, "r");
   if (file == NULL) {
      printf("File does not exist!\n");
      return 0;
   }
   i = 0;
   while (i < N && fscanf(file, "%d", &arr[i]._key) == 1)
      { fscanf(file, "%c", &ch);
      getRow(file, arr[i]._str, sizeof(arr[i]._str));
      i++;
   }
   fclose(file);
   cnt = i;
   printCommands();
   while (flag) {
      scanf("%d", &rule);
      switch (rule) {
         case 1:
            printTable(arr, cnt);
            printCommands();
            break;
         case 2:
            if (!isSorted(arr, cnt))
               { printf("Table not
               sorted!\n");
            }
```

```
                else {
                    printf("Enter key\n");
                    scanf("%d", &key);
                    i = binSe
                    if (i > -1) {
                        printf("Element found \"%s\" by key %d\n", arr[i]._str,key);
                    }
                    else
                        printf("Element not found\n");
                    }
                    printCommands();
                    break;
                case 3:
                    mergesort(arr, 0, cnt - 1);
                    printf("Table sorted\n");
                    printCommands(); break;
                case 4:
                    printf("Table shuffled\n");
                    shuffle(arr, cnt);
                    printCommands();
                    break;
                case 5:
                    printf("Table reversed\n");
                    reverse(arr, cnt);
                    printCommands();
                    break;
                case 6:
                    flag = 0;
                    return 0;
            }
        }
    }
```

table.c:

```
#include <time.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "table.h"


void printTable(row *arr, int size)
    {
    printf("|_____|_____
```

```
    ____________________________|\n");
    printf("| %-7s | %-48s |\n",
    "Key", "Value");
    printf("|_______|_____________
                 __|\n");
    arch(arr, cnt, key);
    for (int i = 0; i < size; i++){
        printf("|%-9d|%-50s|\n",
    arr[i]._key, arr[i]._str);
    }
    printf("|_______|_____________
    ____________________________|\n");
}


void getRow(FILE *stream, char
    *str, int size)
    { int cnt = 0;
    char ch;
    while ((ch = getc(stream)) !=
    '\n' && cnt < size - 1) {
        str[cnt++] = ch;
        str[cnt] = '\0';
    }
}


void swapRows(row *r1, row
    *r2)
    { row
    tmp;
    tmp = *r1;
    *r1 = *r2;
    *r2 = tmp;
}


int binSearch(row *arr, int size,
    int key) {
    int start = 0, end = size - 1,
    mid;
    if (size <= 0) {
```

```
    return -1;
    }
    while (start < end) {
    mid = start + (end - start) / 2;
        if (arr[mid]._key == key){
            Return mid;
        }
        else
            if (arr[mid]._key < key)
                start = mid + 1;
        else
            end = mid;
    }
    if (arr[end]._key == key)
        return end;
    return -1;
}


void sort(row *arr, int first, int
    last)
{

    int left_iterator, right_iterator,
    middle;
    row mas[last - first + 1];
    middle = (first + last) / 2;
    left_iterator = first;
    right_iterator = middle + 1;

    for(int j = first; j <= last;
        j++){ if ((left_iterator <=
        middle)
    && ((right_iterator > last) ||
    (arr[left_iterator]._key <
    arr[right_iterator]._key))){
            mas[j - first] =
    arr[left_iterator];
            left_iterator++;
        }
        else{
```

```
        mas[j - first] =
arr[right_iterator];
        right_iterator++;
    }
}
for(int j=first; j <=last;j++)
{ arr[j] = mas[j - first];}
}
```

```
void mergesort(row *arr, int start,
    int end){
    if (start < end){ mergesort(arr,
        start, (start +
    end) / 2);
        mergesort(arr, (start + end)/2
    + 1, end);
        sort(arr, start, end);
    }
}
int randomtwo(const int a, const
    int b) {
    return a + rand() % (b - a + 1);
}

void shuffle(row *arr, const int
    size) {
    int i, j, k;
    srand((unsigned int)time(0));
    for (k = 0; k < size; k++) {
        i = randomtwo(0, size - 1);
        j = randomtwo(0, size - 1);
        swapRows(&arr[i], &arr[j]);
    }
}
void reverse(row *arr, int size)
    { int i, j;
    for (i = 0, j = size - 1; i < j; i++,
    j--)
        swapRows(&arr[i], &arr[j]);
}
int isSorted(row *arr, int size)
    { for (int i = 0; i < size - 1;
    i++){
        if (arr[i]._key > arr[i +
    1]._key){
            return 0;
        }
    }
    return 1;
}
```

table.h:

```
de <stdio.h>
#include "table.h"
i
f
n
d
e
f

T
A
B
L
E
_
H

#
d
e
f
i
n
e

T
A
B
L
E
_
H



#
i
n
c
l
u
```

```
typedef struct _row
{ int _key;
char _str[100];
} row;

void printTable(row *arr, int size);
void getRow(FILE *stream, char *str, int size);
void swapRows(row *r1, row *r2);
int binSearch(row *arr, int size, int key);
void mergesort(row *arr, int first, int last);
void shuffle(row *arr, int size);
void reverse(row *arr, int size);
int randomtwo(int a, int b);
int isSorted(row *arr, int size);
#endif
```

st.txt - файл с стихотворением:

1 Two roads diverged in a yellow wood,
2 And sorry I could not travel both
3 And be one traveler, long I stood
5 To where it bent in the undergrowth.
6 Then took the other, as just as fair,
7 And having perhaps the better claim,
9 Though as for that the passing there
10 Had worn them really about the same.
11 And both that morning equally lay
12 In leaves no step had trodden black.
15 I doubted if I should ever come back.
17 Somewhere ages and ages hence:
18 I took the one less traveled by,

Заключение.

Я научился составлять программы на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблицу. Также я повторил сортировку слиянием.