Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра вычислительной математики и программирования

Курсовая работа

По курсу «Языки и методы программирования»

II семестр

Задание №9 «Сортировка и поиск»

Выполнил студент

1-го курса, 105-ой группы

Махмудов О. С.

_____

(подпись)

Научный руководитель

Доцент кафедры 806

Сластушенский Ю. В.

_____

(подпись)

Работа защищена

«__»_____ 2019

Оценка_____

ОГЛАВЛЕНИЕ

ЦЕЛЬ

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 трех 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 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 (группы 3-5), прозы (группы 1, 2) или изображение ASCII-графики (группы 6-8). Каждый элемент таблицы, содержащий ключ и текст записи, распечатывать в отдельной строке.

ПОСТАНОВКА ЗАДАЧИ

Программа принимает данные из текстового файла и делает следующее:

1) Выводит данные из файла

2) Сортирует таблицу методом сортировки Хоара (рекурсивный вариант)

3) Ищет строку по ключу

4) Располагает элементы в обратном порядку

ПРОГРАММА

Функции

int counter (const char *file)	Считает количество элементов таблицы
int keys_poem(const char *file1, const char *file2, const char *file3)	Разделяет ключи и строк по разным файлам
void string_by_key(const char *file, int re, int im)	Выводит строку по ключу
void print_table(const char *file)	Выводит таблицу
void swap(char **first, char **second)	Выполняет замену элементов массива
void qsortx(double *modul, size_t low, size_t high, char **keys, char **poem)	Выполняет сортировку массивов согласно заданию (Рекурсивная сортировка Хоара)
void reverse(double *value, char **keys, char **poem, int iter)	Выполняет реверс таблицы

Код

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
#include <strings.h>
#include <math.h>

#define MAX_Q 10
#define MAX_S 100
#define MAX_K 6

size_t n = MAX_S;

int counter (const char *file) {
        FILE *in = fopen(file, "r");
        int k = 0;
        char a[100];
```

```
	while (!feof(in)) {
		fgetc(in);
		fgetc(in);
		fgetc(in);
		fgetc(in);
		fgetc(in);
		if (fgets(a, 100, in) !=NULL)
			k++;
	}
	rewind(in);
	fclose(in);
	return k;
}

int keys_poem(const char *file1, const char *file2, const char *file3) {
	FILE *in = fopen(file1, "r");
	FILE *keys = fopen(file2, "w");
	FILE *verse = fopen(file3, "w");
	char poem[100];
	char a, b, c, d, e;
	while (!feof(in)) {
		if ((a = fgetc(in)) != EOF) {
			b = fgetc(in);
			c = fgetc(in);
			d = fgetc(in);
			e = fgetc(in);
			fputc(a, keys);
			fputc(b, keys);
			fputc(c, keys);
			fputc(d, keys);
			fputc(e, keys);
			fputs("\n", keys);
			fgetc(in);
		}
		if (fgets(poem, 100, in) != NULL)
			fputs(poem, verse);
	}
		rewind(in);
		fclose(in);
		rewind(keys);
		fclose(keys);
		rewind(verse);
		fclose(verse);
}

void string_by_key(const char *file, int re, int im) {
	FILE *in = fopen(file, "r");
	int k = 0;
	char h[100];
	while (!feof(in)) {
		if (fgetc(in)-'0' == re) {
			fgetc(in);
			fgetc(in);
			fgetc(in);
			if (fgetc(in) - '0' == im) {
				fgets(h, 100, in);
				printf("%s", h);
```

```
                                k++;
                        }
                }
        }
        if (k == 0)
                printf("Key entered incorrectly!\n");
}

void print_table(const char *file) {
        FILE *in = fopen(file, "r");
        char a[100];
        while (!feof(in)) {
                if (fgets(a, 100, in) != NULL)
                        printf("%s", a);
        }
        rewind(in);
        fclose(in);
}

void swap(char **first, char **second) {
        char *temp = *first;
        *first = *second;
        *second = temp;
}

void qsortx(double *modul, size_t low, size_t high, char **keys, char **poem) {
        size_t i, j;
        double tmp, pivot;

        i = low;
        j = high;

        pivot = modul[(low + (high - low) / 2)];
        do {
                while (modul[i] < pivot) {
                        i++;
                }
                while (modul[j] > pivot) {
                        j--;
                }
                if (i <= j) {
                        if (modul[i] > modul[j]) {
                                tmp = modul[i];
                                modul[i] = modul[j];
                                modul[j] = tmp;
                                swap(&keys[i], &keys[j]);
                                swap(&poem[i], &poem[j]);

                        }
                        i++;
                        if (j > 0) {
                                j--;
                        }
                }
        } while (i <= j);

        if (i < high) {
```

```
                qsortx(modul, i, high, keys, poem);
        }
        if (j > low) {
                qsortx(modul, low, j, keys, poem);
        }
}

void reverse(double *value, char **keys, char **poem, int iter) {
        char **tmp1, **tmp2, *tmp;
        double modul[MAX_Q];
        int i = 0;
        tmp1 = malloc(MAX_Q * sizeof(char*));
        tmp1[i] = malloc(MAX_Q * sizeof(char*));
        tmp2 = malloc(MAX_Q * sizeof(char*));
        tmp2[i] = malloc(MAX_Q * sizeof(char*));
        for (i = 0; i < iter / 2; i++) {
                tmp1[i] = keys[i];
                tmp2[i] = poem[i];
                modul[i] = value[i];
                keys[i] = keys[iter - i - 1];
                poem[i] = poem[iter - i - 1];
                value[i] = value[iter - i - 1];
                keys[iter - i - 1] = tmp1[i];
                poem[iter - i - 1] = tmp2[i];
                value[iter - i - 1] = modul[i];
        }
}

void menu() {
        printf("==============================\n");
        printf("||   1-Print table         ||\n");
        printf("||   2-Print of sort table  ||\n");
        printf("||   3-Print string by key  ||\n");
        printf("||   4-Reverse table        ||\n");
        printf("||   5-Menu                 ||\n");
        printf("||   0-End                  ||\n");
        printf("==============================\n");
        printf("\n");
}

int main() {
        int a;
        char ch = '9';
        char **keys, **poem;
        a = counter("table.txt");
        keys_poem("table.txt", "keys.txt", "verse.txt");
        menu();
        FILE *in = fopen("verse.txt", "r");
        int k = 0, count = 0, imag, real;
        poem = malloc(MAX_Q * sizeof(char*));
        poem[k] = malloc(MAX_Q * sizeof(char*));
        bzero(poem[k], MAX_Q);
        while (getline(&poem[k], &n, in) != -1 && count < a) {
                count++;
                k++;
                poem[k] = malloc(MAX_Q * sizeof(char*));
                bzero(poem[k], MAX_Q);
```

```
}
rewind(in);
fclose(in);

FILE *in2 = fopen("keys.txt", "r");
 k = 0;
char string[MAX_K];
keys = malloc(MAX_Q * sizeof(char *));
keys[k] = malloc(MAX_Q * sizeof(char*));
while (!feof(in2) && k < MAX_Q) {
        fscanf(in2, "%s\n", string);
        keys[k] = strdup(string);
        k++;
}
for (k = 0; k < a; k++)
rewind(in2);
fclose(in2);

FILE *modul = fopen("keys.txt", "r");
int z = 0;
double value[a];
int re, im;
double r, i, sum_sqr, mod;
while (!feof(modul)) {
        re = fgetc(modul);
        fgetc(modul);
        fgetc(modul);
        fgetc(modul);
        im = fgetc(modul);
        fgetc(modul);
        if (re >= '0' && re <= '9' && im >= '0' && im <= '9') {
                r = re - '0';
                i = im - '0';
                sum_sqr = pow(r, 2) + pow(i, 2);
                mod = sqrt(sum_sqr);
                value[z] = mod;
        }
        z++;
}
rewind(modul);
fclose(modul);

while (ch != '0') {
        printf("=> ");
        scanf(" %c", &ch);
        switch (ch) {
        case '1':
                print_table("table.txt");
                break;
        case '2':
                a = counter("table.txt");
                qsortx(value, 0, a - 1, keys, poem);
                for (int t = 0; t < a; t++)
                        printf("%s  %s", keys[t], poem[t]);
                break;
        case '3':
                printf("Enter the real part: ");
```

```
                scanf("%d", &real);
                printf("Enter the imaginary part: ");
                scanf("%d", &imag);
                string_by_key("table.txt", real, imag);
                break;
        case '4':
                a = counter("table.txt");
                reverse(value, keys, poem, a);
                for (int p = 0; p < a; p++)
                        printf("%s  %s", keys[p], poem[p]);
                break;
        case '5':
                menu();
                break;
        }
  }
  return 0;

}
```

Содержимое файла

5+i*6 Two roads diverged in a yellow wood,

7+i*3 And sorry I could not travel both

8+i*2 And be one traveler, long I stood

6+i*5 And looked down one as far as I could

9+i*4 To where it bent in the undergrowth.

7+i*9 Then took the other, as just as fair,

5+i*2 And having perhaps the better claim,

7+i*9 Because it was grassy and wanted wear;

4+i*2 Though as for that the passing there

5+i*3 Had worn them really about the same

Тесты

Admin@LAPTOP-Q5U6S2UH:/mnt/c/Users/Admin/Desktop/Все для вуза/Курсач 2-ой семак (инфа)/Курсач №9$ ./a.out

```
==============================
||  1-Print table             ||
||  2-Print of sort table     ||
||  3-Print string by key     ||
||  4-Reverse table           ||
||  5-Menu                    ||
||  0-End                     ||
==============================
=> 1
5+i*6 Two roads diverged in a yellow wood,
7+i*3 And sorry I could not travel both
8+i*2 And be one traveler, long I stood
6+i*5 And looked down one as far as I could
9+i*4 To where it bent in the undergrowth.
7+i*9 Then took the other, as just as fair,
5+i*2 And having perhaps the better claim,
7*i+9 Because it was grassy and wanted wear;
4+i*2 Though as for that the passing there
5+i*3 Had worn them really about the same
=> 2
4+i*2  Though as for that the passing there
5+i*2  And having perhaps the better claim,
5+i*3  Had worn them really about the same
7+i*3  And sorry I could not travel both
6+i*5  And looked down one as far as I could
5+i*6  Two roads diverged in a yellow wood,
8+i*2  And be one traveler, long I stood
```

9+i*4 To where it bent in the undergrowth.

7+i*9 Then took the other, as just as fair,

7*i+9 Because it was grassy and wanted wear;

=> 4

7*i+9 Because it was grassy and wanted wear;

7+i*9 Then took the other, as just as fair,

9+i*4 To where it bent in the undergrowth.

8+i*2 And be one traveler, long I stood

5+i*6 Two roads diverged in a yellow wood,

6+i*5 And looked down one as far as I could

7+i*3 And sorry I could not travel both

5+i*3 Had worn them really about the same

5+i*2 And having perhaps the better claim,

4+i*2 Though as for that the passing there

=> 2

4+i*2 Though as for that the passing there

5+i*2 And having perhaps the better claim,

5+i*3 Had worn them really about the same

7+i*3 And sorry I could not travel both

5+i*6 Two roads diverged in a yellow wood,

6+i*5 And looked down one as far as I could

8+i*2 And be one traveler, long I stood

9+i*4 To where it bent in the undergrowth.

7+i*9 Then took the other, as just as fair,

7*i+9 Because it was grassy and wanted wear;

=> 5

==========

|| 1-Print table ||

|| 2-Print of sort table ||

|| 3-Print string by key ||

|| 4-Reverse table ||

```
||  5-Menu                          ||
||  0-End                           ||
==============================
=> 3
Enter the real part: 8
Enter the imaginary part: 2
 And be one traveler, long I stood
=> 3
Enter the real part: 5
Enter the imaginary part: 6
 Two roads diverged in a yellow wood,
=> 7
=> 9
=> 3
Enter the real part: 7
Enter the imaginary part: 9
 Then took the other, as just as fair,
 Because it was grassy and wanted wear;
```

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы я улучшил свои навыки работы со строками и файлами, смог реализовать сортировку комплексного типа данных и реализовал быструю сортировку Хоара рекурсивным методом.