**Учреждение образования**

**„Белорусский государственный университет информатики и радиоэлектроники”**

**Кафедра «Вычислительных методов и программирования»**

**ОТЧЕТ**

**По лабораторным работам №11 и №12**

**«Сортировка массивов. Поиск по ключу в одномерном массиве»**

**Выполнил:**

**Студент АСОИ**

**Группа №020601**

**Шумигай В.В.**

**Вариант № 15**

**Проверил:**

**Беспалов С.A.**

**Минск 2020**

**Цель работы:** сформировать умения использования различных сортировок массивов и поиска по ключу. Написать и отладить программы, содержащие сортировки и поиск по ключу.

**Индивидуальное задание:**

**11.** Ключ: объем двигателя автомобиля. Методы сортировки: QuickSort и метод Шелла.

**12.** Вывести на экран марку автомобиля с объемом двигателя 1998 см3. Метод поиска: интерполяционный.

**Текст программы:**

#include <iostream>

#include <cstdio>

#include <io.h>

#include <string.h>

using namespace std;

FILE\* fl;

struct Cars

{

int year;

char marka[10];

int V;

int rashod;

} \*list, car;

char fname[20] = "";

int n = 0, f;

void fadd();

void frd();

void rezc();

void rezf();

int menu();

void qsort();

void mshela();

void poisk();

bool flopen(const char\*);

int main()

{

while (true)

{

switch (menu())

{

case 1: fadd(); break;

case 2: frd(); break;

case 3: rezc(); break;

case 4: rezf(); break;

case 5: qsort(); break;

case 6: mshela(); break;

case 7: poisk(); break;

case 8: return 0;

default:"Error!";

}

system("pause");

system("cls");

}

return 0;

}

int menu()

{

cout << "Select: " << endl;

cout << "1.Enter data and write to file" << endl;

cout << "2.Read data from file" << endl;

cout << "3.Display the result on the screen" << endl;

cout << "4.Write the result to a file" << endl;

cout << "5.QuickSort" << endl;

cout << "6.Metod Shela sort" << endl;

cout << "7.Interpol poisk" << endl;

cout << "8.Exit" << endl;

cin >> f;

return f;

}

bool flopen(const char\* r)

{

if (!strlen(fname))

{

cout << "Enter file name: " << endl;

cin >> fname;

}

if (fopen\_s(&fl, fname, r))

{

cout << "Error" << endl;

return false;

}

else

return true;

}

void fadd()

{

if (!flopen("ab+")) return;

int i, n;

cout << "Enter the number of cars: ";

cin >> n;

cout << "--------------------------------" << endl;

for (i = 0; i < n; i++)

{

cout << "Vvedite god vipuska: "; cin >> car.year; getchar();

cout << "Vvedite marku: "; gets\_s(car.marka, 10);

cout << "Vvedite obiem dvigatela: "; cin >> car.V;

cout << "Vvedite rashod: "; cin >> car.rashod; cout << endl;

fwrite(&car, sizeof(Cars), 1, fl);

}

fclose(fl);

}

void frd()

{

if (!flopen("rb")) return;

n = \_filelength(\_fileno(fl)) / sizeof(Cars);

list = new Cars[n];

fread(list, sizeof(Cars), n, fl);

for (int i = 0; i < n; i++)

cout << endl << list[i].year << " Marka - " << list[i].marka << ", Obiem - " << list[i].V << ", Rashod - " << list[i].rashod;

cout << endl;

delete[]list;

fclose(fl);

}

void rezc()

{

if (!flopen("rb")) return;

n = \_filelength(\_fileno(fl)) / sizeof(Cars);

Cars t;

list = new Cars[n];

fread(list, sizeof(Cars), n, fl);

for (int i = 0; i < n - 1; i++) {

for (int j = n - 1; j > 0; j--) {

if (list[j - 1].rashod > list[j].rashod) {

t = list[j - 1];

list[j - 1] = list[j];

list[j] = t;

}

}

}

for (int i = 0; i < n; i++)

if (list[i].year > 2004)

cout << endl << list[i].year << " Marka - " << list[i].marka << ", Obiem - " << list[i].V << ", Rashod - " << list[i].rashod;

cout << endl;

delete[]list;

fclose(fl);

}

void rezf()

{

char fnamet[20];

cout << "Enter the name of the txt file" << endl;

cin >> fnamet;

FILE\* ft;

if (fopen\_s(&ft, fnamet, "w"))

{

cout << "File not created!";

return;

}

if (!flopen("rb")) return;

n = \_filelength(\_fileno(fl)) / sizeof(Cars);

Cars t;

list = new Cars[n];

fread(list, sizeof(Cars), n, fl);

for (int i = 0; i < n-1; i++) {

for (int j = n - 1; j > 0; j--) {

if (list[j - 1].rashod > list[j].rashod) {

t = list[j - 1];

list[j - 1] = list[j];

list[j] = t;

}

}

}

for (int i = 0; i < n; i++)

if (list[i].year > 2004)

fprintf(ft, "%d %s %d %d\n", list[i].year, list[i].marka, list[i].V, list[i].rashod);

delete[]list;

fclose(ft);

fclose(fl);

}

void qsort()

{

struct

{

int l;

int r;

} stack[20];

int i, j, left, right, x, s = 0;

Cars t;

if (!flopen("rb")) return;

n = \_filelength(\_fileno(fl)) / sizeof(Cars);

list = new Cars[n];

fread(list, sizeof(Cars), n, fl);

stack[s].l = 0; stack[s].r = n - 1;

while (s != -1)

{

left = stack[s].l; right = stack[s].r;

s--;

while (left < right)

{

i = left; j = right; x = list[(left + right) / 2].V;

while (i <= j)

{

while (list[i].V < x)i++;

while (list[j].V > x)j--;

if (i <= j)

{

t = list[i]; list[i] = list[j]; list[j] = t;

i++; j--;

}

}

if ((j - left) < (right - i))

{

if (i < right) { s++; stack[s].l = i; stack[s].r = right;}

right = j;

}

else

{

if (left < j) { s++; stack[s].l = left; stack[s].r = j; }

left = i;

}

}

}

for (int i = 0; i < n; i++)

cout << endl << list[i].year << " Marka - " << list[i].marka << ", Obiem - " << list[i].V << ", Rashod - " << list[i].rashod;

cout << endl;

if (f != 7) delete[]list;

fclose(fl);

}

void mshela()

{

if (!flopen("rb")) return;

n = \_filelength(\_fileno(fl)) / sizeof(Cars);

list = new Cars[n];

fread(list, sizeof(Cars), n, fl);

Cars t;

int d = n / 2;

while (d > 0)

{

for (int i = 0; i < n - d; i++)

{

int j = i;

while (j >= 0 && list[j].V > list[j + d].V)

{

t = list[j];

list[j] = list[j + d];

list[j + d] = t;

j--;

}

}

d = d / 2;

}

for (int i = 0; i < n; i++)

cout << endl << list[i].year << " Marka - " << list[i].marka << ", Obiem - " << list[i].V << ", Rashod - " << list[i].rashod;

cout << endl;

delete[]list;

fclose(fl);

}

void poisk()

{

if (!flopen("rb")) return;

n = \_filelength(\_fileno(fl)) / sizeof(Cars);

qsort();

int i = 0, j = n - 1, m, x;

cout << "Vvedite V: "; cin >> x;

while (i < j)

{

if (list[i].V == list[j].V)

if (list[i].V == x) {

cout << list[i].marka << endl; return;

}

else return;

m = i + (j - i) \* (x - list[i].V) / (list[j].V - list[i].V);

if (list[m].V == x) { cout << list[m].marka << endl; return; }

else if (x > list[m].V) i = m + 1; else j = m - 1;

}

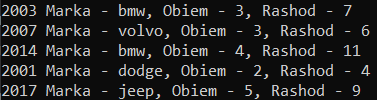
delete[]list;

fclose(fl);

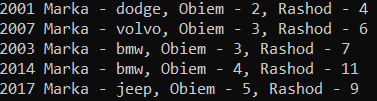
}

**Результат работы программы:**

Исходная:

****

Сортировка по объему:



Поиск:

