#include <iostream>

#include <fstream>

using namespace std;

struct student

{

char FIO[30];

double sb;

double money;

};

int kolvo (); // вычисляем количество штук в файле

student\* zapolnenie(int kol); // считываем из файла

int ZP(); // вычисляем номер перелома

void Qiwi(int left, int right); // квас по зп

void Qiwi\_Z(int left, int right);// квас по баллу

void BbIBOD(); // вывод массива

void menu(); // менюшка

void Lineal(); // линейный сорт по зп

void Lineal\_Z(int z); // линейный сорт по баллу

void Dopolnenie(); // добавляем студента

void binary\_search(); // бинарный поиск

void linely\_search(); // линейный поиск

int kol = kolvo();

student \*spisok = zapolnenie(kol);

int main()

{

while (true)

{

menu();

int V; cin >> V;

system("cls");

switch (V)

{

case 1: //квас

Qiwi(0, (kol-1));

Qiwi\_Z(ZP(), kol);

break;

case 2: //линейная сортировка

Lineal();

Lineal\_Z(ZP());

break;

case 3: //просто вывод списка студентов

BbIBOD();

break;

case 4: //добавление студентов

Dopolnenie();

break;

case 5: //поиск по зарплате?

linely\_search();

break;

case 0:

return 0;

}

}

}

student\* zapolnenie(int kol)

{

student \*spisok = new student[30];

char SB[5], MONEY[30];

ifstream fin("spisok.txt");

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

{

fin >> spisok[i].FIO;

fin >> SB; spisok[i].sb = atof(SB);

fin >> MONEY; spisok[i].money = atof(MONEY);

}

fin.close();

return spisok;

}

void menu()

{

system("cls");

cout << "Количество студентов: " << kol << endl << endl;

cout << "Ваши действия?" << endl;

cout << " 1. Q-сортировка.\n 2. L-сортировка.\n 3. Вывести список студентов.\n 4. Добавить студента.\n 5. Поиск по зарплате?\n 0. Выход." << endl;

cout << "-->";

}

int kolvo()

{

setlocale(LC\_ALL, "rus");

int k=0;

ifstream io("spisok.txt");

char sth[30];

while (!io.eof())

{

io >> sth;

k++;

}

k = k / 3;

if (k > 30)

k = 30;

io.close();

return k;

}

void BbIBOD()

{

for (int r = 0; r < kol; r++)

{

cout << "№ [" << r + 1 << "]" << endl;

cout << "Имя студента:\t" << spisok[r].FIO << endl;

cout << "Средний балл студента:\t" << spisok[r].sb << endl;

cout << "Заработок семьи:\t" << spisok[r].money << endl << endl;

}

system("pause");

}

void Dopolnenie()

{

char SB[5], MONEY[5];

kol++;

ofstream fout("spisok.txt", ios\_base::app);

system("cls");

cout << "Введите имя студента:\t";

cin >> spisok[kol - 1].FIO; fout << endl << spisok[kol - 1].FIO;

cout << endl << "Введите средний балл студента:\t";

cin >> SB; fout << endl << SB; spisok[kol - 1].sb = atof(SB);

cout << endl << "Введите заработок семьи студента:\t";

cin >> MONEY; fout << endl << MONEY; spisok[kol - 1].money = atof(MONEY);

fout.close();

}

void binary\_search()

{

system("cls");

double zap;

int i = 0, j=kol, n;

cout << "Введите искомую зарплату:\t";

cin >> zap;

cout << endl << endl;

while (i < j)

{

n = (i + j) / 2;

if (zap > spisok[n].money)

i = n + 1;

else j = n;

}

if (spisok[i].money == zap)

{

//cout << "Искомый студент:" << endl << endl;

cout << "Имя студента:\t" << spisok[i].FIO << endl;

cout << "Средний балл студента:\t" << spisok[i].sb << endl;

cout << "Заработок семьи:\t" << spisok[i].money << endl << endl;

}

else

{

cout << "Такого студента нет!";

}

system("pause");

}

void linely\_search()

{

system("cls");

double zap;

int nu=0;

cout << "Введите искомую зарплату:\t";

cin >> zap;

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

{

if (spisok[i].money == zap)

{

nu++;

cout << endl << endl;

//cout << "Искомый студент:" << endl << endl;

cout << "Имя студента:\t" << spisok[i].FIO << endl;

cout << "Средний балл студента:\t" << spisok[i].sb << endl;

cout << "Заработок семьи:\t" << spisok[i].money << endl << endl;

}

}

if (nu == 0)

{

cout << endl << "Такого студента нет!" << endl << endl;

}

system("pause");

}

int ZP()

{

int e = 0;

while (spisok[e].money < 20)

e++;

return e;

}

void Qiwi(int left, int right)

{

int i = left, j = right;

student t, x;

x = spisok[(i + j) / 2];

do {

while (spisok[i].money < x.money && i < right) i++;

while (spisok[j].money > x.money && j > left) j--;

if (i <= j) {

t = spisok[i];

spisok[i] = spisok[j];

spisok[j] = t;

i++; j--;

}

} while (i <= j);

if (left < j) Qiwi(left, j);

if (i < right) Qiwi(i, right);

}

void Qiwi\_Z(int left, int right)

{

int i = left, j = right;

student t, x;

x = spisok[(i + j) / 2];

do {

while (spisok[i].sb > x.sb && i < right) i++;

while (spisok[j].sb < x.sb && j > left) j--;

if (i <= j) {

t = spisok[i];

spisok[i] = spisok[j];

spisok[j] = t;

i++; j--;

}

} while (i <= j);

if (left < j) Qiwi\_Z(left, j);

if (i < right) Qiwi\_Z(i, right);

}

void Lineal()

{

student Buff;

for (int y = 0; y < kol; y++)

{

for (int l = kol-1; l > y; l--)

{

if (spisok[l-1].money > spisok[l].money)

{

Buff = spisok[l-1];

spisok[l-1] = spisok[l];

spisok[l] = Buff;

}

}

}

}

void Lineal\_Z(int z)

{

student Buff;

for (int y = z; y < kol; y++)

{

for (int l = kol - 1; l > y; l--)

{

if (spisok[l - 1].sb < spisok[l].sb)

{

Buff = spisok[l - 1];

spisok[l - 1] = spisok[l];

spisok[l] = Buff;

}

}

}

}