#include<iostream>

using namespace std;

struct Datum {

int\* dan, \* mjesec, godina;

void Unos(int d, int m, int g)

{

dan = new int(d);

mjesec = new int(m);

godina = g;

}

void Ispis()

{

cout << \*dan << "." << \*mjesec << "." << godina << endl;

}

void Dealociraj()

{

delete dan;

dan = nullptr;

delete mjesec;

mjesec = nullptr;

}

};

struct Predmet {

char\* Naziv;

int ECTS;

Datum\* DatumOdgovaranja;

int\* Ocjena;

void Unos(const char\* n, int ects, Datum d, int o)

{

Naziv = new char[strlen(n) + 1];

strcpy\_s(Naziv, strlen(n) + 1, n);

ECTS = ects;

DatumOdgovaranja = new Datum(d);

DatumOdgovaranja->Unos(\*d.dan, \*d.mjesec, d.godina);

Ocjena = new int(o);

}

void Ispis()

{

cout << "Naziv: " << Naziv << endl;

cout << "ECTS: " << ECTS << endl;

DatumOdgovaranja->Ispis();

cout << "Ocjena: " << \*Ocjena << endl;

}

void Dealociraj()

{

delete Naziv;

Naziv = nullptr;

DatumOdgovaranja->Dealociraj();

delete DatumOdgovaranja;

DatumOdgovaranja = nullptr;

delete Ocjena;

Ocjena = nullptr;

}

};

struct Ucenik {

char\* Ime;

char Prezime[15];

char Index[10];

Datum\* DatumRodjenja;

Predmet \_predmeti[100];

int\* trenutnoPredmeta;

void Unos(const char\* i, const char\* p, const char\* index, Datum d)

{

Ime = new char[strlen(i) + 1];

strcpy\_s(Ime, strlen(i) + 1, i);

strcpy\_s(Prezime, strlen(p) + 1, p);

strcpy\_s(Index, strlen(index) + 1, index);

DatumRodjenja = new Datum(d);

DatumRodjenja->Unos(\*d.dan, \*d.mjesec, d.godina);

trenutnoPredmeta = new int(0);

}

void Ispis()

{

cout << "Ime: " << Ime << endl;

cout << "Prezime: " << Prezime << endl;

cout << "Index: " << Index << endl;

DatumRodjenja->Ispis();

for (int i = 0; i < \*trenutnoPredmeta; i++)

{

\_predmeti[i].Ispis();

}

}

void Dealociraj()

{

delete Ime;

Ime = nullptr;

DatumRodjenja->Dealociraj();

delete DatumRodjenja;

DatumRodjenja = nullptr;

delete trenutnoPredmeta;

trenutnoPredmeta = nullptr;

}

bool dodajPredmet(Predmet p)

{

if (\*trenutnoPredmeta == 100)

return false;

\_predmeti[\*trenutnoPredmeta].Unos(p.Naziv, p.ECTS, \*p.DatumOdgovaranja, \*p.Ocjena);

(\*trenutnoPredmeta)++;

return true;

}

float Prosjek()// prosjek ucenika

{

int suma = 0, prosjek = 0;

for (int i = 0; i < \*trenutnoPredmeta; i++)

{

suma += \*\_predmeti[i].Ocjena;

}

prosjek = (float)suma / (\*trenutnoPredmeta);

return (float)prosjek;

}

};

//Prebrojiti koliko je maloljetnih studenata proslo vrlo dobrim uspjehom

int Prebroji(Ucenik\* niz, int vel)

{

int brojac = 0;

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

{

for (int j = 0; j < \*niz[i].trenutnoPredmeta; j++)

{

if (niz[i].DatumRodjenja->godina > 2005 &&

niz[i].Prosjek() >= 3.5 && niz[i].Prosjek() <= 4.4)

{

brojac++;

}

}

}

return brojac;

}

//Provjeriti da li je bar 50 % ucenika odlicno.

//Funkcija treba da ispise indekse svih odlicnih ucenika.

bool PolaOdlicnih(Ucenik\* niz, int vel)

{

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

{

int brojOdlicnih = 0, brojUcenika = 0;

for (int j = 0; j < \*niz[i].trenutnoPredmeta; j++)

{

brojUcenika++;

if (niz[i].Prosjek() >= 4.5 && niz[i].Prosjek() <= 5.0)

{

brojOdlicnih++;

}

if ((brojUcenika - brojOdlicnih) % 2 == 0)

return true;

}

return false;

}

}

void main() {

int vel = 3;

Ucenik\* niz = new Ucenik[vel];

Datum datumRodjenja1, datumRodjenja2, datumRodjenja3,

datumPolaganja1, datumPolaganja2, datumPolaganja3;

datumRodjenja1.Unos(1, 12, 1999);

datumRodjenja2.Unos(2, 10, 2000);

datumRodjenja3.Unos(3, 5, 2005);

datumPolaganja1.Unos(1, 12, 2021);

datumPolaganja2.Unos(1, 10, 2020);

datumPolaganja3.Unos(1, 12, 2021);

Predmet predmet1, predmet2, predmet3;

predmet1.Unos("Matematika", 6, datumPolaganja1, 5);

predmet2.Unos("Matematika", 6, datumPolaganja2, 1);

predmet3.Unos("Programiranje", 6, datumPolaganja3, 2);

niz[0].Unos("Dalila", "Kolicic", "528528", datumRodjenja1);

niz[1].Unos("Ajla", "Kolasinac", "486532", datumRodjenja2);

niz[2].Unos("Adel", "Grebovic", "151321", datumRodjenja3);

niz[0].dodajPredmet(predmet1);

niz[0].dodajPredmet(predmet2);

niz[1].dodajPredmet(predmet3);

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

{

niz[i].Ispis();

}

//pozvati i testirati globalne funkcije

cout << "Maloljetni vrlodobri: " << Prebroji(niz, vel) << endl;

if (PolaOdlicnih(niz, vel) == true)

cout << "TRUE" << endl;

else

cout << "FALSE" << endl;

}