1.

#include<iostream>

using namespace std;

struct Rezervacija {

char \*imePrezime;

int brojOsoba;

double uplata;

void Unos() {

char temp[100];

cout << "Unesite ime i prezime: ";

cin.getline(temp, 100);

imePrezime = new char[strlen(temp) + 1];

strcpy\_s(imePrezime, strlen(temp) + 1, temp);

cout << "Unesite iznos novca koji uplaćujete: ";

cin >> uplata;

cout << "Unesite broj osoba: ";

cin >> brojOsoba;

cin.ignore();

}

void Ispis() {

cout << "Rezervacija na ime " << imePrezime << ". Broj osoba: " << brojOsoba << ". Uplaceno: " << uplata << " KM.\n";

}

void Dealociraj() {

delete[]imePrezime;

imePrezime = nullptr;

}

};

struct Proslava {

char \* naziv;

Rezervacija \* gosti[50];

int brojRezervacija;

void Unos() {

char temp[100];

cout << "Unesite naziv proslave: ";

cin.getline(temp, 100);

naziv = new char[strlen(temp) + 1];

strcpy\_s(naziv, strlen(temp) + 1, temp);

brojRezervacija = 0;

}

void Ispis() {

cout << "Proslava: " << naziv << endl;

cout << ":::::::::::::::::: REZERVACIJE ::::::::::::::::::\n";

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

{

gosti[i]->Ispis();

cout << "====================================\n";

}

}

void AddRezervacija() {

if (brojRezervacija == 50)

cout << "Nemoguće dodati više od 50 rezervacija." << endl;

else {

gosti[brojRezervacija] = new Rezervacija;

gosti[brojRezervacija]->Unos();

brojRezervacija++;

}

}

void Dealociraj() {

delete[]naziv;

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

{

delete gosti[i];

gosti[i] = nullptr;

}

}

};

Rezervacija \* GetRezervacija(Proslava \* p) {

char temp[30];

cout << "Unesite ime i prezime za pretragu:";

cin.getline(temp, 30);

for (int i = 0; i < p->brojRezervacija; i++)

{

if (strcmp(p->gosti[i]->imePrezime, temp) == 0)

return p->gosti[i];

}

return NULL;

}

void PrikaziDuznike(Proslava \* p) {

int dug, duznici = 0;

cout << "Unesite cifru prema kojoj zelite pretraziti dugovanja::";

cin >> dug;

for (int i = 0; i < p->brojRezervacija; i++)

{

double a=0;

if (p->gosti[i]->uplata < p->gosti[i]->brojOsoba \* 50) {

a = p->gosti[i]->brojOsoba \* 50 - p->gosti[i]->uplata;

if (a > dug) {

p->gosti[i]->Ispis();

duznici++;

}

}

}

int izbor, j = 0;

Rezervacija \*niz = new Rezervacija[duznici];

for (int i = 0; i < p->brojRezervacija; i++) {

double a=0;

if (p->gosti[i]->uplata < p->gosti[i]->brojOsoba \* 50) {

a = p->gosti[i]->brojOsoba \* 50 - p->gosti[i]->uplata;

if (a > dug) {

niz[j] = \*p->gosti[i];

j++;

}

}

}

}

void main() {

Proslava \*p = new Proslava;

p->Unos();

cin.ignore();

p->AddRezervacija();

p->AddRezervacija();

Rezervacija \*r = GetRezervacija(p);

if (r != NULL)

r->Ispis();

else

cout << "Rezervacija sa trazenim imenom ne postoji!" << endl;

PrikaziDuznike(p);

delete p;

p = nullptr;

system("pause>0");

}

2.

#include<iostream>

using namespace std;

struct Rezervacija {

char \*imePrezime;

int brojOsoba;

double uplata;

void Unos() {

char temp[100];

cout << "Unesite ime i prezime: ";

cin.getline(temp, 100);

imePrezime = new char[strlen(temp) + 1];

strcpy\_s(imePrezime, strlen(temp) + 1, temp);

cout << "Unesite iznos novca koji uplaćujete: ";

cin >> uplata;

cout << "Unesite broj osoba: ";

cin >> brojOsoba;

cin.ignore();

}

void Ispis() {

cout << "Rezervacija na ime " << imePrezime << ". Broj osoba: " << brojOsoba << ". Uplaceno: " << uplata << " KM.\n";

}

void Dealociraj() {

delete[]imePrezime;

imePrezime = nullptr;

}

};

struct Proslava {

char \* naziv;

Rezervacija gosti[50];

int brojRezervacija;

void Unos() {

char temp[100];

cout << "Unesite naziv proslave: ";

cin.getline(temp, 100);

naziv = new char[strlen(temp) + 1];

strcpy\_s(naziv, strlen(temp) + 1, temp);

brojRezervacija = 0;

}

void Ispis() {

cout << "Proslava: " << naziv << endl;

cout << ":::::::::::::::::: REZERVACIJE ::::::::::::::::::\n";

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

{

gosti[i].Ispis();

cout << "====================================\n";

}

}

void AddRezervacija() {

if (brojRezervacija == 50)

cout << "Nemoguće dodati više od 50 rezervacija." << endl;

else {

gosti[brojRezervacija].Unos();

brojRezervacija++;

}

}

void Dealociraj() {

delete[]naziv;

}

};

Rezervacija \* GetRezervacija(Proslava \* p) {

char temp[30];

cout << "Unesite ime i prezime za pretragu:";

cin.getline(temp, 30);

for (int i = 0; i < p->brojRezervacija; i++)

{

if (strcmp(p->gosti[i].imePrezime, temp) == 0)

return &p->gosti[i];

}

return NULL;

}

void PrikaziDuznike(Proslava \* p) {

int dug, duznici = 0;

cout << "Unesite cifru prema kojoj zelite pretraziti dugovanja::";

cin >> dug;

for (int i = 0; i < p->brojRezervacija; i++)

{

double a=0;

if (p->gosti[i].uplata < p->gosti[i].brojOsoba \* 50) {

a = p->gosti[i].brojOsoba \* 50 - p->gosti[i].uplata;

if (a > dug) {

p->gosti[i].Ispis();

duznici++;

}

}

}

int izbor, j = 0;

Rezervacija \*niz = new Rezervacija[duznici];

for (int i = 0; i < p->brojRezervacija; i++) {

double a=0;

if (p->gosti[i].uplata < p->gosti[i].brojOsoba \* 50) {

a = p->gosti[i].brojOsoba \* 50 - p->gosti[i].uplata;

if (a > dug) {

niz[j] = p->gosti[i];

j++;

}

}

}

}

void main() {

Proslava \*p = new Proslava;

p->Unos();

cin.ignore();

p->AddRezervacija();

p->AddRezervacija();

Rezervacija \*r = GetRezervacija(p);

if (r != NULL)

r->Ispis();

else

cout << "Rezervacija sa trazenim imenom ne postoji!" << endl;

PrikaziDuznike(p);

delete p;

p = nullptr;

system("pause>0");

}

3.

#include<iostream>

using namespace std;

struct Rezervacija {

char \*imePrezime;

int brojOsoba;

double uplata;

void Unos() {

char temp[100];

cout << "Unesite ime i prezime: ";

cin.getline(temp, 100);

imePrezime = new char[strlen(temp) + 1];

strcpy\_s(imePrezime, strlen(temp) + 1, temp);

cout << "Unesite iznos novca koji uplaćujete: ";

cin >> uplata;

cout << "Unesite broj osoba: ";

cin >> brojOsoba;

cin.ignore();

}

void Ispis() {

cout << "Rezervacija na ime " << imePrezime << ". Broj osoba: " << brojOsoba << ". Uplaceno: " << uplata << " KM.\n";

}

void Dealociraj() {

delete[]imePrezime;

imePrezime = nullptr;

}

};

struct Proslava {

char \* naziv;

Rezervacija \*gosti;

int maxRezervacija;

int brojRezervacija;

void Unos() {

char temp[100];

cout << "Unesite naziv proslave: ";

cin.getline(temp, 100);

naziv = new char[strlen(temp) + 1];

strcpy\_s(naziv, strlen(temp) + 1, temp);

cout << "Unesite maksimalan broj rezervacija: ";

cin >> maxRezervacija;

cin.ignore();

gosti = new Rezervacija[maxRezervacija];

brojRezervacija = 0;

}

void Ispis() {

cout << "Proslava: " << naziv << endl;

cout << ":::::::::::::::::: REZERVACIJE ::::::::::::::::::\n";

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

{

gosti[i].Ispis();

cout << "====================================\n";

}

}

void AddRezervacija() {

if (brojRezervacija == maxRezervacija)

cout << "Nemoguće dodati više od 50 rezervacija." << endl;

else {

gosti[brojRezervacija].Unos();

brojRezervacija++;

}

}

void Dealociraj() {

delete[]naziv;

delete[]gosti;

gosti = nullptr;

}

};

Rezervacija \* GetRezervacija(Proslava \* p) {

char temp[30];

cout << "Unesite ime i prezime za pretragu:";

cin.getline(temp, 30);

for (int i = 0; i < p->brojRezervacija; i++)

{

if (strcmp(p->gosti[i].imePrezime, temp) == 0)

return &p->gosti[i];

}

return NULL;

}

void PrikaziDuznike(Proslava \* p) {

int dug, duznici = 0;

cout << "Unesite cifru prema kojoj zelite pretraziti dugovanja::";

cin >> dug;

for (int i = 0; i < p->brojRezervacija; i++)

{

double a=0;

if (p->gosti[i].uplata < p->gosti[i].brojOsoba \* 50) {

a = p->gosti[i].brojOsoba \* 50 - p->gosti[i].uplata;

if (a > dug) {

p->gosti[i].Ispis();

duznici++;

}

}

}

int izbor, j = 0;

Rezervacija \*niz = new Rezervacija[duznici];

for (int i = 0; i < p->brojRezervacija; i++) {

double a=0;

if (p->gosti[i].uplata < p->gosti[i].brojOsoba \* 50) {

a = p->gosti[i].brojOsoba \* 50 - p->gosti[i].uplata;

if (a > dug) {

niz[j] = p->gosti[i];

j++;

}

}

}

}

void main() {

Proslava \*p = new Proslava;

p->Unos();

cin.ignore();

p->AddRezervacija();

p->AddRezervacija();

Rezervacija \*r = GetRezervacija(p);

if (r != NULL)

r->Ispis();

else

cout << "Rezervacija sa trazenim imenom ne postoji!" << endl;

PrikaziDuznike(p);

delete p;

p = nullptr;

system("pause>0");

}