

PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 4

2023-2024.

Zadatak 1

Definirajte strukturu Osoba sa članovima ime, prezime, OIB i godina rođenja. Sami odredite tipove podataka. Napravite dvije instance strukture. U prvu instancu učitate vrijednosti od korisnika, zatim sve vrijednosti iskopirajte u drugu instancu te na kraju ispišite sve vrijednosti iz druge instance na ekran.

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

struct Osoba {
    string ime;
    string prezime;
    string oib;
    int godina_rodjenja;
};

int main() {
    Osoba osoba1;
    cout << "Ime: ";
    getline(cin, osoba1.ime);
    cout << "Prezime: ";
    getline(cin, osoba1.prezime);
    cout << "OIB: ";
    getline(cin, osoba1.oib);
    cout << "Godina rođenja: ";
    cin >> osoba1.godina_rodjenja;

    Osoba osoba2 = osoba1;

    cout << osoba2.ime << " " << osoba2.prezime << " (" << osoba2.oib << ", " <<
osoba2.godina_rodjenja << ")" << endl;

    return 0;
}
```

Zadatak 2

Definirajte strukturu koja čuva informacije o automobilima koji su koristili garažu (pogledajte donju tablicu). Kreirajte polje automobila sa svim automobilima iz tablice. Na kraju, omogućite korisniku da unese sat i minutu pa mu ispišite sve automobile koji su ušli u garažu nakon tog trenutka. Za svaki automobil ispišite i je li još u garaži ili ju je napustio. Primjerice, ako korisnik upiše 11 sati i 45 minuta, potrebno je ispisati:

ST 1112-AB (napustio garazu)

PU 632-IN (jos je u garazi)

Automobil	Sat ulaska	Minuta ulaska	Napustio garažu
ZG 2566-PF	11	16	NE
RI 356-HF	11	37	DA
ST 1112-AB	11	55	DA
PU 632-IN	12	14	NE

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

struct KorisnikGaraze {
    string automobil;
    int sat;
    int minuta;
    bool napustio_garazu;
};

int main() {
    const int n = 4;

    KorisnikGaraze korisnici[n];
    korisnici[0].automobil = "ZG 2566-PF";
    korisnici[0].sat = 11;
    korisnici[0].minuta = 16;
    korisnici[0].napustio_garazu = false;
    korisnici[1].automobil = "RI 356-HF";
    korisnici[1].sat = 11;
    korisnici[1].minuta = 37;
    korisnici[1].napustio_garazu = true;
    korisnici[2].automobil = "ST 1112-AB";
    korisnici[2].sat = 11;
    korisnici[2].minuta = 55;
    korisnici[2].napustio_garazu = true;
    korisnici[3].automobil = "PU 632-IN";
    korisnici[3].sat = 12;
    korisnici[3].minuta = 14;
    korisnici[3].napustio_garazu = false;

    int min_sat;
    cout << "Upisite sat ulaska: ";
    cin >> min_sat;

    int min_minuta;
    cout << "Upisite minutu ulaska: ";
```

```
    cin >> min_minuta;

    for (int i = 0; i < n; i++) {
        if (korisnici[i].sat > min_sat ||
            (korisnici[i].sat == min_sat && korisnici[i].minuta >
min_minuta)) {
            cout << korisnici[i].automobil << " (";
            if (korisnici[i].napustio_garazu) {
                cout << "napustio garazu";
            }
            else {
                cout << "jos je u garazi";
            }
            cout << ")" << endl;
        }
    }

    return 0;
}
```

Zadatak 3

Definirajte strukturu Drzava koja sadrži naziv države, broj stanovnika u državi (u milijunima) te popis proizvoljnog broja gradova (0 ili više). Nakon toga, napravite polje od tri države te svakoj izmislite broj stanovnika i dodajte proizvoljni broj gradova. Nakon toga, ispišite sve države, a gradove svake države odvojite zarezom, uz točku na kraju popisa. Primjer ispisa:

Belgija ima 11.5 milijuna stanovnika i gradove: Antwerpen, Bruxelles, Brugge.
Finska ima 5.5 milijuna stanovnika i gradove: Helsinki, Turku, Tampere, Rovaniemi.
Juzna Koreja ima 51 milijun stanovnika i gradove: Seul, Incheon.

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Drzava {
    string naziv;
    double broj_stanovnika;
    vector<string> gradovi;
};

int main() {
    const int n = 3;

    Drzava drzave[n];
    drzave[0].naziv = "Belgija";
    drzave[0].broj_stanovnika = 11.5;
    drzave[0].gradovi.push_back("Antwerpen");
    drzave[0].gradovi.push_back("Bruxelles");
    drzave[0].gradovi.push_back("Brugge");

    drzave[1].naziv = "Finska";
    drzave[1].broj_stanovnika = 5.5;
    drzave[1].gradovi.push_back("Helsinki");
    drzave[1].gradovi.push_back("Turku");
    drzave[1].gradovi.push_back("Tampere");
    drzave[1].gradovi.push_back("Rovaniemi");

    drzave[2].naziv = "Juzna Koreja";
    drzave[2].broj_stanovnika = 51;
    drzave[2].gradovi.push_back("Seul");
    drzave[2].gradovi.push_back("Incheon");

    for (int i = 0; i < n; i++) {
        cout << drzave[i].naziv << " ima " << drzave[i].broj_stanovnika << "
milijuna stanovnika i gradove: ";

        for (int j = 0; j < drzave[i].gradovi.size(); j++) {
            cout << drzave[i].gradovi[j];
            if (j < drzave[i].gradovi.size() - 1) {
                cout << ", ";
            }
            else {
                cout << ".";
            }
        }
    }
}
```

```
        }  
    }  
    cout << endl;  
}  
return 0;  
}
```

Zadatak 4

Dovršite program tako da bude ispravan, ako je zadano:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

int main() {
    Racun racun;
    racun.broj_racuna = "358/1/43";
    racun.nacin_placanja = "Visa";

    Stavka stavka1;
    stavka1.redni_broj = 1;
    stavka1.proizvod = "Kava s mlijekom";
    stavka1.cijena = 1.20;

    Stavka stavka2;
    stavka2.redni_broj = 2;
    stavka2.proizvod = "Mineralna voda";
    stavka2.cijena = 1.50;

    racun.stavke.push_back(stavka1);
    racun.stavke.push_back(stavka2);

    return 0;
}
```

Moguće rješenje:

```
#include<iostream>
#include<string>
#include<vector>
using namespace std;

struct Stavka {
    int redni_broj;
    string proizvod;
    double cijena;
};

struct Racun {
    string broj_racuna;
    string nacin_placanja;
    vector<Stavka> stavke;
};

int main() {
    Racun racun;
    racun.broj_racuna = "358/1/43";
    racun.nacin_placanja = "Visa";

    Stavka stavka1;
    stavka1.redni_broj = 1;
    stavka1.proizvod = "Kava s mlijekom";
    stavka1.cijena = 1.20;
```

```
    Stavka stavka2;
    stavka2.redni_broj = 2;
    stavka2.proizvod = "Mineralna voda";
    stavka2.cijena = 1.50;

    racun.stavke.push_back(stavka1);
    racun.stavke.push_back(stavka2);

    return 0;
}
```


Zadatak 5

Napišite program koji definira strukturu Datum sa članovima godina, mjesec i dan. Napišite funkciju koja prima dva objekta tipa Datum i vraća manjeg od njih. Pripremite dva datuma pa pomoću funkcije pronađite i ispišite manjeg u obliku „dan.mjesec.godina.“, npr. „8.12.2023.“

Moguće rješenje:

```
#include <iostream>
using namespace std;

struct Datum {
    int dan;
    int mjesec;
    int godina;
};

Datum manji(Datum d1, Datum d2) {
    if (d1.godina < d2.godina) {
        return d1;
    }
    else if (d1.godina > d2.godina) {
        return d2;
    }
    else { // godine su jednake
        if (d1.mjesec < d2.mjesec) {
            return d1;
        }
        else if (d1.mjesec > d2.mjesec) {
            return d2;
        }
        else { // godine i mjeseci su jednaki
            if (d1.dan < d2.dan) {
                return d1;
            }
            else {
                return d2;
            }
        }
    }
}

// Ova funkcija radi istu stvar
Datum manji2(Datum d1, Datum d2) {
    int ticks_d1 = d1.godina * 365 + d1.mjesec * 12 + d1.dan;
    int ticks_d2 = d2.godina * 365 + d2.mjesec * 12 + d2.dan;

    if (ticks_d1 < ticks_d2) {
        return d1;
    }
    else {
        return d2;
    }
}
```

```
int main() {
    Datum d1;
    d1.dan = 5;
    d1.mjesec = 9;
    d1.godina = 2023;

    Datum d2;
    d2.dan = 9;
    d2.mjesec = 9;
    d2.godina = 2023;

    Datum m = manji2(d1, d2);
    cout << m.dan << "." << m.mjesec << "." << m.godina << "." << endl;

    return 0;
}
```

Zadatak 6

Napišite program koji priprema vektor objekata iz donje tablice. Omogućite korisniku da upiše duljinu u kilometrima, a vi mu ispišite sve rijeke koje su dulje od toga. Primjerice, ako korisnik upiše 6500, onda trebate ispisati:

Nil (Afrika)

Amazona (Juzna Amerika)

Rijeka	Kontinent	Duljina [km]
Nil	Afrika	6695
Amazona	Južna Amerika	6570
Jangce	Azija	6380

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Rijeka {
    string naziv;
    string kontinent;
    int duljina;
};

int main() {
    vector<Rijeka> rijeke;

    Rijeka r1;
    r1.naziv = "Nil";
    r1.kontinent = "Afrika";
    r1.duljina = 6695;

    Rijeka r2;
    r2.naziv = "Amazona";
    r2.kontinent = "Juzna Amerika";
    r2.duljina = 6570;

    Rijeka r3;
    r3.naziv = "Jangce";
    r3.kontinent = "Azija";
    r3.duljina = 6380;

    rijeke.push_back(r1);
    rijeke.push_back(r2);
    rijeke.push_back(r3);

    int min_duljina;
    cout << "Upisite minimalnu duljinu rijeke: ";
    cin >> min_duljina;

    for (int i = 0; i < rijeke.size(); i++) {
        if (rijeke[i].duljina > min_duljina) {
            cout << rijeke[i].naziv << " (" << rijeke[i].kontinent << ")" <<
endl;
```

```
    }  
  }  
  return 0;  
}
```

Zadatak 7

Napišite program koji definira strukturu IpAdresa koja čuva podatke o jednoj IP adresi (dobro razmislite kako ćete čuvati te podatke). Napišite funkciju koja prima jednu IpAdresu i vraća koje je klase, A, B ili C, prema sljedećim kriterijima:

- Klasu A čine adrese od 0.0.0.0 do 127.255.255.255
- Klasu B čine adrese od 128.0.0.0 do 191.255.255.255
- Klasu C čine adrese od 192.0.0.0 do 223.255.255.255

Pozovite funkciju iz main()-a i ispišite kojoj klasi pripada ta IP adresa kao u primjeru:

IP adresa 13.107.246.60 pripada klasi A

Moguće rješenje:

```
#include <iostream>
using namespace std;

struct IpAdresa {
    int prvi_bajt;
    int drugi_bajt;
    int treci_bajt;
    int cetvrti_bajt;
};

char klasa(IpAdresa ip) {
    if (ip.prvi_bajt <= 127) {
        return 'A';
    }
    else if (ip.prvi_bajt <= 191) {
        return 'B';
    }
    else {
        return 'C';
    }
}

int main() {
    // ping openai.com
    // Pinging openai.com [13.107.246.60] with 32 bytes of data :
    IpAdresa ip;
    ip.prvi_bajt = 13;
    ip.drugi_bajt = 107;
    ip.treci_bajt = 246;
    ip.cetvrti_bajt = 60;

    cout << "IP adresa " << ip.prvi_bajt << "." << ip.drugi_bajt << "." <<
ip.treci_bajt << "." << ip.cetvrti_bajt << " pripada klasi " << klasa(ip) << endl;

    return 0;
}
```

Zadatak 8

Napišite program koji definira strukture Zvijezda, Planet i Mjesec. Za sva tri svemirska tijela omogućite čuvanje naziva. Dodatno, za zvijezde omogućite čuvanje proizvoljnog broja planeta, a za planete proizvoljnog broja mjesece. Primjerice, za zvijezdu Sunce možemo upisati planete Merkur, Veneru i Mars, a za planet Mars mjesece Deimos i Phobos. Pripremite vektor s nekoliko zvijezda. Na kraju, ispišite sadržaj tako da zvijezde ispišete bez uvlačenja, planete jednom uvučeno, a mjesece dva puta, kao u prikazu:

```
Sunce
    Merkur
    Mars
        Deimos
        Phobos
    Jupiter
        Io
        Ganymede
        Callisto
Kosjenka
```

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Mjesec {
    string naziv;
};

struct Planet {
    string naziv;
    vector<Mjesec> mjesece;
};

struct Zvijezda {
    string naziv;
    vector<Planet> planeti;
};

int main() {
    Mjesec deimos;
    deimos.naziv = "Deimos";

    Mjesec phobos;
    phobos.naziv = "Phobos";

    Mjesec io;
    io.naziv = "Io";

    Mjesec ganymede;
    ganymede.naziv = "Ganymede";

    Mjesec callisto;
    callisto.naziv = "Callisto";
```

```

Planet merkur;
merkur.naziv = "Merkur";

Planet mars;
mars.naziv = "Mars";
mars.mjeseci.push_back(deimos);
mars.mjeseci.push_back(phobos);

Planet jupiter;
jupiter.naziv = "Jupiter";
jupiter.mjeseci.push_back(io);
jupiter.mjeseci.push_back(ganymede);
jupiter.mjeseci.push_back(callisto);

Zvijezda sunce;
sunce.naziv = "Sunce";
sunce.planeti.push_back(merkur);
sunce.planeti.push_back(mars);
sunce.planeti.push_back(jupiter);

Zvijezda kosjenka;
kosjenka.naziv = "Kosjenka";

vector<Zvijezda> zvijezde;
zvijezde.push_back(sunce);
zvijezde.push_back(kosjenka);

for (int i = 0; i < zvijezde.size(); i++) {
    cout << zvijezde[i].naziv << endl;

    for (int j = 0; j < zvijezde[i].planeti.size(); j++) {
        cout << "\t" << zvijezde[i].planeti[j].naziv << endl;

        for (int k = 0; k < zvijezde[i].planeti[j].mjeseci.size(); k++)
            cout << "\t\t" << zvijezde[i].planeti[j].mjeseci[k].naziv
<< endl;
    }
}

return 0;
}

```

Zadatak 9

Napišite program koji definira strukturu Udžbenik koja sadrži članove naziv i broj stranica. Učitavajte od korisnika udžbenike dok to korisnik želi. Nakon učitavanja, ispišite naziv udžbenika koji ima najviše stranica (ispišite i broj stranica) te ispišite ukupan zbroj stranica svih udžbenika. Primjer rada:

```
Upisite naziv: Udžbenik 1
Upisite broj stranica: 42
Dalje (1=da, 0=ne): 1
Upisite naziv: Udžbenik 2
Upisite broj stranica: 195
Dalje (1=da, 0=ne): 1
Upisite naziv: Udžbenik 3
Upisite broj stranica: 75
Dalje (1=da, 0=ne): 0
Najviše stranica (195) ima Udžbenik 2
Ukupno svi udžbenici imaju: 312 stranica
```

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Udžbenik {
    string naziv;
    int broj_stranica;
};

int main() {
    vector<Udžbenik> udžbenici;
    bool dalje;

    do {
        Udžbenik novi;
        cout << "Upisite naziv: ";
        getline(cin, novi.naziv);
        cout << "Upisite broj stranica: ";
        cin >> novi.broj_stranica;
        udžbenici.push_back(novi);

        cout << "Dalje (1=da, 0=ne): ";
        cin >> dalje;
        cin.ignore();
    } while (dalje);

    Udžbenik najviše_stranica = udžbenici[0];
    int ukupno_stranica = 0;

    for (int i = 0; i < udžbenici.size(); i++) {
        if (udžbenici[i].broj_stranica > najviše_stranica.broj_stranica) {
            najviše_stranica = udžbenici[i];
        }
        ukupno_stranica += udžbenici[i].broj_stranica;
    }
}
```



```
        cout << "Najviše stranica (" << najviše_stranica.broj_stranica << ") ima " <<
najviše_stranica.naziv << endl;
        cout << "Ukupno svi udzbenici imaju: " << ukupno_stranica << " stranica" <<
endl;

        return 0;
}
```

Zadatak 10

Napišite program koji definira strukturu Prognoza sa članovima datum, najveća temperatura i opis vremena (primjerice, za datum „9.12.2023.“ je najveća temperatura 2 stupnja, a opis „oblacno“). Omogućite korisniku da unese tri prognoze. Nakon unosa svih prognoza, ispišite ih u sljedećem formatu:

Upisite datum: 9.12.2023.

Upisite najveću temperaturu: 2

Upisite opis: oblacno

Upisite datum: 10.12.2023.

Upisite najveću temperaturu: 7

Upisite opis: kisa

Upisite datum: 11.12.2023.

Upisite najveću temperaturu: 8

Upisite opis: oblacno

Na datum 9.12.2023. ce najveca temperatura biti 2 (oblacno)

Na datum 10.12.2023. ce najveca temperatura biti 7 (kisa)

Na datum 11.12.2023. ce najveca temperatura biti 8 (oblacno)

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

struct Prognoza {
    string datum;
    int max_temperatura;
    string opis;
};

int main() {
    const int n = 3;
    Prognoza prognoze[n];

    for (int i = 0; i < n; i++) {
        cout << "Upisite datum: ";
        getline(cin, prognoze[i].datum);
        cout << "Upisite najveću temperaturu: ";
        cin >> prognoze[i].max_temperatura;
        cin.ignore();
        cout << "Upisite opis: ";
        getline(cin, prognoze[i].opis);
    }

    for (int i = 0; i < n; i++) {
        cout << "Na datum " << prognoze[i].datum << " ce najveca temperatura
biti " << prognoze[i].max_temperatura << " (" << prognoze[i].opis << ")" << endl;
    }

    return 0;
}
```

Zadatak 11

Napišite program koji definira strukturu TekuciRacun sa članovima broj i stanje te u beskonačnoj petlji korisniku omogućite sljedeće:

- Otvaranje novog tekućeg računa
- Dodavanje sredstava na tekući račun. Pretpostavite da će korisnik uvijek unijeti postojeći broj računa.
- Ispis svih tekućih računa s iznosima.

Primjer rada programa:

```
1 = otvaranje novog racuna
2 = dodavanje sredstava na racun
3 = ispis svega
> 1
Upisite broj racuna: HR1723600001101234565
> 2
Upisite broj racuna: HR1723600001101234565
Koliko EUR zelite dodati: 1520
> 1
Upisite broj racuna: AT483200000012345864
> 2
Upisite broj racuna: AT483200000012345864
Koliko EUR zelite dodati: 4000
> 3
HR1723600001101234565: 1520 EUR
AT483200000012345864: 4000 EUR
> 2
Upisite broj racuna: HR1723600001101234565
Koliko EUR zelite dodati: 485
> 3
HR1723600001101234565: 2005 EUR
AT483200000012345864: 4000 EUR
...
```

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct TekuciRacun {
    string broj;
    double iznos;
};

int main() {
    vector<TekuciRacun> racuni;
    char op;
    string broj;
    double iznos;
    TekuciRacun racun;

    cout << "1 = otvaranje novog racuna" << endl;
    cout << "2 = dodavanje sredstava na racun" << endl;
```

```

cout << "3 = ispis svega" << endl;

while (true) {
    cout << "> ";
    cin >> op;
    cin.ignore();

    switch (op) {
        case '1':
            racun.iznos = 0.0;
            cout << "Upisite broj racuna: ";
            getline(cin, racun.broj);
            racuni.push_back(racun);
            break;
        case '2':
            cout << "Upisite broj racuna: ";
            getline(cin, broj);
            cout << "Koliko EUR zelite dodati: ";
            cin >> iznos;
            for (int i = 0; i < racuni.size(); i++) {
                if (racuni[i].broj == broj) {
                    racuni[i].iznos += iznos;
                    break;
                }
            }
            break;
        case '3':
            for (int i = 0; i < racuni.size(); i++) {
                cout << racuni[i].broj << ": " << racuni[i].iznos << "
EUR" << endl;
            }
            break;
        default:
            cout << "Nepoznata operacija" << endl;
            break;
    }
}

return 0;
}

```

Zadatak 12

Napišite program koji priprema polje objekata iz donje tablice. Nakon toga, omogućite korisniku da upiše s koliko novca raspolaže, a vi mu ispišite do kojih sve odredišta može stići. Primjer rada programa:

Unesite iznos kojim raspolazete: 8.00

Mozete posjetiti neko od sljedećih odredista:

Ivanic Grad

Kutina

Novska

Odredište	Iznos cestarine
Ivanić Grad	1.50 EUR
Kutina	4.20 EUR
Novska	5.50 EUR
Sl. Brod zapad	10.80 EUR

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

struct Odrediste {
    string naziv;
    double iznos;
};

int main() {
    const int n = 4;
    Odrediste odredista[n];
    odredista[0].naziv = "Ivanic Grad";
    odredista[0].iznos = 1.50;
    odredista[1].naziv = "Kutina";
    odredista[1].iznos = 4.20;
    odredista[2].naziv = "Novska";
    odredista[2].iznos = 5.50;
    odredista[3].naziv = "Sl. Brod zapad";
    odredista[3].iznos = 10.80;

    int iznos_na_raspolaganju;
    cout << "Unesite iznos kojim raspolazete: ";
    cin >> iznos_na_raspolaganju;

    cout << "Mozete posjetiti neko od sljedećih odredista:" << endl;
    for (int i = 0; i < n; i++) {
        if (odredista[i].iznos <= iznos_na_raspolaganju) {
            cout << odredista[i].naziv << endl;
        }
    }

    return 0;
}
```

Zadatak 13

Napišite program koji priprema vektor objekata iz donje tablice. Nakon toga, učitajte od korisnika polazni i odredišni aerodrom pa mu ispišite sve raspoložive letove. Ako nema niti jednog raspoloživog leta, ispišite to. Primjer rada programa:

Upisite oznaku polaznog aerodroma: ZAG

Upisite oznaku odredišnog aerodroma: CDG

Raspoloživi letovi:

OU220

AF5112

Polazište	Odredište	Broj leta
ZAG	CDG	OU220
CDG	VIE	LH1951
MUC	FRA	LH9562
ZAG	CDG	AF5112
ZAG	DBV	OU440

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Let {
    string polaziste;
    string odrediste;
    string broj;
};

int main() {
    vector<Let> letovi;

    Let let1;
    let1.polaziste = "ZAG";
    let1.odrediste = "CDG";
    let1.broj = "OU220";
    letovi.push_back(let1);

    Let let2;
    let2.polaziste = "CDG";
    let2.odrediste = "VIE";
    let2.broj = "LH1951";
    letovi.push_back(let2);

    Let let3;
    let3.polaziste = "MUC";
    let3.odrediste = "FRA";
    let3.broj = "LH9562";
    letovi.push_back(let3);

    Let let4;
    let4.polaziste = "ZAG";
    let4.odrediste = "CDG";
    let4.broj = "AF5112";
```

```

letovi.push_back(let4);

Let let5;
let5.polaziste = "ZAG";
let5.odrediste = "DBV";
let5.broj = "OU440";
letovi.push_back(let5);

string polaziste;
cout << "Upisite oznaku polazniskog aerodroma: ";
getline(cin, polaziste);

string odrediste;
cout << "Upisite oznaku odrediskog aerodroma: ";
getline(cin, odrediste);

bool pronadjen_ijedan_let = false;
cout << "Raspoloživi letovi:" << endl;
for (int i = 0; i < letovi.size(); i++) {
    if (letovi[i].polaziste == polaziste &&
        letovi[i].odrediste == odrediste) {
        cout << letovi[i].broj << endl;
        pronadjen_ijedan_let = true;
    }
}

if (!pronadjen_ijedan_let) {
    cout << "Na žalost, nema niti jednog raspoloživog leta" << endl;
}

return 0;
}

```

Zadatak 14

Napišite program koji definira strukturu `Korisnik` koja može čuvati podatke o imenu, prezimenu, korisničkom imenu, lozinci te je li korisnik administrator ili nije. Napišite funkciju koja od korisnika učitava jedan objekt tipa `Korisnik` i vraća ga pozivatelju. Napišite i funkciju koja prima objekt tipa `Korisnik` i ispisuje ga na ekran. U `main()`-u, iskoristite funkcije da učitajte i ispišete jednog korisnika.

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

struct Korisnik {
    string ime;
    string prezime;
    string korisnicko_ime;
    string lozinka;
    bool is_admin;
};

Korisnik ucitaj() {
    Korisnik k;

    cout << "ime: ";
    getline(cin, k.ime);
    cout << "prezime: ";
    getline(cin, k.prezime);
    cout << "korisnicko ime: ";
    getline(cin, k.korisnicko_ime);
    cout << "lozinka: ";
    getline(cin, k.lozinka);
    cout << "Je li korisnik administrator (1=da, 0=ne): ";
    cin >> k.is_admin;

    return k;
}

void ispisi(Korisnik k) {
    cout << k.ime << " " << k.prezime << endl;
    cout << k.korisnicko_ime << " / " << k.lozinka << endl;
    if (k.is_admin) {
        cout << "Administrator" << endl;
    }
    else {
        cout << "Obicni korisnik" << endl;
    }
}

int main() {
    ispisi(ucitaj());

    return 0;
}
```


Zadatak 15

Napišite program koji definira strukturu `ZnanstveniRad` koja može čuvati podatke o jednom ili više autora, naslovu te godini izdanja. U `main()`, učitajte od korisnika podatke o jednom znanstvenom radu. Nakon završenog učitavanja, pošaljite objekt u funkciju koja ga treba ispisati u sljedećem formatu:

- Autori su međusobno odvojeni zarezima, iza zadnjeg autora nema zareza.
- Godina se piše u okruglim zagradama
- Naslov se navodi zadnji s točkom na kraju.

Primjer rada programa:

Autor: R. E. Johnson

Unijeti jos jednog autora (1=da, 0=ne): 1

Autor: P. J. Huggins

Unijeti jos jednog autora (1=da, 0=ne): 0

Naslov: Toroidal Atmospheres around Extrasolar Planets

Godina izdanja: 2006

R. E. Johnson, P. J. Huggins (2006) Toroidal Atmospheres around Extrasolar Planets.

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct ZnanstveniRad {
    vector<string> autori;
    string naslov;
    int godina_izdanja;
};

void ispisi(ZnanstveniRad zr) {
    for (int i = 0; i < zr.autori.size(); i++) {
        cout << zr.autori[i];
        if (i < zr.autori.size() - 1) {
            cout << ", ";
        }
    }

    cout << " (" << zr.godina_izdanja << ") " << zr.naslov << "." << endl;
}

int main() {
    ZnanstveniRad zr;
    bool dalje;
    string autor;

    do {
        cout << "Autor: ";
        getline(cin, autor);
        zr.autori.push_back(autor);

        cout << "Unijeti jos jednog autora (1=da, 0=ne): ";
        cin >> dalje;
        cin.ignore();
    } while (dalje);
}
```

```
    } while (dalje);  
  
    cout << "Naslov: ";  
    getline(cin, zr.naslov);  
  
    cout << "Godina izdanja: ";  
    cin >> zr.godina_izdanja;  
  
    ispisi(zr);  
  
    return 0;  
}
```

Zadatak 16

Napišite program koji simulira rad DNS poslužitelja. Pripremite polje objekata prema donjoj tablici i u beskonačnoj petlji omogućite korisniku dvije opcije:

- Ako unese domensko ime, vi mu ispišite IP adresu.
- Ako unese IP adresu, vi mu ispišite sva domenska imena na toj IP adresi.

Primjer rada programa:

```
1 = domensko ime => ip adresa
2 = ip adresa => domenska imena
> 1
Upit: mail.google.com
142.250.180.133
> 2
Upit: 142.251.209.3
www.google.hr
> 2
Upit: 178.218.162.131
algebra.hr
www.algebra.hr
> ...
```

Domensko ime	IP adresa
www.google.com	142.250.180.132
www.google.hr	142.251.209.3
mail.google.com	142.250.180.133
algebra.hr	178.218.162.131
www.algebra.hr	178.218.162.131

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

struct DnsZapis {
    string domensko_ime;
    string ip_adresa;
};

int main() {
    const int n = 5;
    DnsZapis dns[n];
    dns[0].domensko_ime = "www.google.com";
    dns[0].ip_adresa = "142.250.180.132";
    dns[1].domensko_ime = "www.google.hr";
    dns[1].ip_adresa = "142.251.209.3";
    dns[2].domensko_ime = "mail.google.com";
    dns[2].ip_adresa = "142.250.180.133";
    dns[3].domensko_ime = "algebra.hr";
    dns[3].ip_adresa = "178.218.162.131";
    dns[4].domensko_ime = "www.algebra.hr";
    dns[4].ip_adresa = "178.218.162.131";
```

```

cout << "1 = domensko ime => ip adresa" << endl;
cout << "2 = ip adresa => domenska imena" << endl;

char opcija;
string upit;

while (true) {
    cout << "> ";
    cin >> opcija;
    cin.ignore();

    cout << "Upit: ";
    getline(cin, upit);

    switch (opcija) {
        case '1':
            for (int i = 0; i < n; i++) {
                if (dns[i].domensko_ime == upit) {
                    cout << dns[i].ip_adresa << endl;
                    break;
                }
            }
            break;
        case '2':
            for (int i = 0; i < n; i++) {
                if (dns[i].ip_adresa == upit) {
                    cout << dns[i].domensko_ime << endl;
                }
            }
            break;
        default:
            cout << "Nepoznata opcija" << endl;
            break;
    }
}

return 0;
}

```

Zadatak 17

Napišite program koji definira strukturu Pločica sa članovima x i y koordinata lijevog gornjeg kuta. Neka program od korisnika učitava širinu i visinu ekrana u pikselima (pretpostavite da će korisnik uvijek unijeti širinu i visinu djeljivu s brojem 10). Neka program zatim generira potrebni broj objekata tipa Pločica širine i visine 10 piksela tako da se pokrije cijela širina i visina ekrana, ako pretpostavimo da je prva generirana pločica na koordinatama (0, 0). Ispišite sve generirane pločice. Primjer rada programa:

Upisite širinu ekrana: 80

Upisite visinu ekrana: 50

```
(0,0) (10,0) (20,0) (30,0) (40,0) (50,0) (60,0) (70,0)
(0,10) (10,10) (20,10) (30,10) (40,10) (50,10) (60,10) (70,10)
(0,20) (10,20) (20,20) (30,20) (40,20) (50,20) (60,20) (70,20)
(0,30) (10,30) (20,30) (30,30) (40,30) (50,30) (60,30) (70,30)
(0,40) (10,40) (20,40) (30,40) (40,40) (50,40) (60,40) (70,40)
```

Moguće rješenje:

```
#include <iostream>
#include <vector>
using namespace std;

struct Plocica {
    int x;
    int y;
};

int main() {
    int sirina;
    cout << "Upisite širinu ekrana: ";
    cin >> sirina;
    int visina;
    cout << "Upisite visinu ekrana: ";
    cin >> visina;

    vector<Plocica> plocice;
    for (int i = 0; i < visina; i += 10) {
        for (int j = 0; j < sirina; j += 10) {
            Plocica pl;
            pl.x = j;
            pl.y = i;
            plocice.push_back(pl);
        }
    }

    for (int i = 0; i < plocice.size(); i++) {
        if (plocice[i].x == 0) {
            cout << endl;
        }
        cout << "(" << plocice[i].x << ", " << plocice[i].y << ")\t";
    }

    return 0;
}
```

Zadatak 18

Napišite program koji definira strukturu `KolicinaKise` koja može čuvati količinu pale kiše i redni broj mjeseca (primjerice, u 6. mjesecu je palo 47 mm kiše). Napravite polje sa svih 12 mjeseci i za svaki mjesec generirajte slučajnu količinu pale kiše između 25 i 100 mm. Nakon toga, pronađite u kojem mjesecu je palo najviše kiše i ispišite njegov broj i količinu kiše.

Moguće rješenje:

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;

struct KolicinaKise {
    int mjesec;
    int kolicina;
};

int main() {
    srand(time(nullptr));

    const int n = 12;
    KolicinaKise padaline[n];

    for (int i = 0; i < n; i++) {
        padaline[i].mjesec = i + 1;
        padaline[i].kolicina = 25 + rand() % (100 - 25 + 1);
    }

    KolicinaKise max = padaline[0];
    for (int i = 1; i < n; i++) {
        if (padaline[i].kolicina > max.kolicina) {
            max = padaline[i];
        }
    }

    cout << "Najviše kiše je palo u mjesecu broj " << max.mjesec << " (" <<
max.kolicina << " mm)" << endl;

    return 0;
}
```

Zadatak 19

Napišite program koji definira strukturu Osoba sa članovima ime, prezime i godina rođenja. Napravite jedan objekt i u njega učitate podatke od korisnika. Zatim iskopirajte objekt 1.000.0000 puta.

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Osoba {
    string ime;
    string prezime;
    int godina_rodjenja;
};

int main() {
    Osoba predlozak;

    cout << "Ime: ";
    getline(cin, predlozak.ime);
    cout << "Prezime: ";
    getline(cin, predlozak.prezime);
    cout << "Godina rođenja: ";
    cin >> predlozak.godina_rodjenja;

    vector<Osoba> klonovi;
    for (int i = 0; i < 1000000; i++) {
        klonovi.push_back(predlozak);
    }

    return 0;
}
```

Zadatak 20

Napišite program koji definira strukture Točka i Pravokutnik. Neka Točka čuva podatke o x i y koordinati, a Pravokutnik čuva podatke o koordinatama gornjeg lijevog kuta te širini i visini. Napišite funkciju koja prima točku i pravokutnik i vraća nalazi li se točka unutar pravokutnika. Pozovite funkciju s točkom i pravokutnikom i ispišite rezultat. Pretpostavite da se (0,0) nalazi u gornjem lijevom kutu, a da x i y rastu u desno i prema dolje.

Moguće rješenje:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Tocka {
    int x;
    int y;
};

struct Pravokutnik {
    Tocka gore_lijevo;
    int sirina;
    int visina;
};

bool kolizija(Tocka t, Pravokutnik p) {
    if (t.x >= p.gore_lijevo.x &&
        t.x <= p.gore_lijevo.x + p.sirina &&
        t.y >= p.gore_lijevo.y &&
        t.y <= p.gore_lijevo.y + p.visina) {
        return true;
    }
    return false;
}

int main() {
    Tocka t;
    t.x = 5;
    t.y = 4;

    Pravokutnik p;
    p.gore_lijevo.x = 5;
    p.gore_lijevo.y = 5;
    p.sirina = 10;
    p.visina = 10;

    if (kolizija(t, p)) {
        cout << "Tocka je unutar pravokutnika" << endl;
    }
    else {
        cout << "Tocka je izvan pravokutnika" << endl;
    }

    return 0;
}
```