

# PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 2

2023-2024.

## Zadatak 1

Napišite program koji učitava pet imena od korisnika u polje, a zatim učitava i broj N. Neka program ispiše sva imena iz polja čiji je broj znakova manji od N. Na primjer, ako korisnik unese "Anna", "Mary", "Peter", "Bart" i "Anastasia" i N=5, program bi trebao prikazati:

Anna  
Mary  
Bart

*Moguće rješenje:*

---

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

int main() {
    string imena[5];

    for (int i = 0; i < 5; i++) {
        cout << "Upisite ime: ";
        getline(cin, imena[i]);
    }

    int n;
    cout << "Upisite n: ";
    cin >> n;

    for (int i = 0; i < 5; i++) {
        if (imena[i].length() < n) {
            cout << imena[i] << endl;
        }
    }

    return 0;
}
```

## Zadatak 2

Napišite program koji generira deset slučajnih brojeva između 1 i 100 i smješta ih u polje. Nakon toga izračunajte prosjek tih brojeva. Nakon toga kopirajte sve brojeve veće ili jednake prosjeku u vektor. Na kraju ispišite sve brojeve iz polja, zatim prosjek, a zatim sve brojeve iz vektora.

*Moguće rješenje:*

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

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

    int brojevi[10];

    // Prvo: napunimo polje slučajnim brojevima
    for (int i = 0; i < 10; i++) {
        brojevi[i] = 1 + rand() % (100 - 1 + 1);
    }

    // Drugo: izračunamo prosjek
    double sum = 0.0;
    for (int i = 0; i < 10; i++) {
        sum += brojevi[i];
    }

    double average = sum / 10;

    // Treće: kopiramo u vektor
    vector<int> veci;

    for (int i = 0; i < 10; i++) {
        if (brojevi[i] >= average) {
            veci.push_back(brojevi[i]);
        }
    }

    // Četvrto: ispisemo
    cout << "Brojevi iz polja: ";
    for (int i = 0; i < 10; i++) {
        cout << brojevi[i] << " ";
    }
    cout << endl << "Prosjek je: " << average << endl;
    cout << "Brojevi iz vektora: ";
    for (int i = 0; i < veci.size(); i++) {
        cout << veci[i] << " ";
    }

    return 0;
}
```

### Zadatak 3

Napišite program koji priprema vektor s tri naziva filma. Ispišite sve filmove od kraja vektora do početka vektora, ali tako da naziv svakog filma ispišete od kraja do početka. Na primjer, ako su filmovi u vektoru "Matrix", "Star Wars", "Finding Nemo", tada bi ispisi trebali biti:

```
omeN gnidniF  
sraW ratS  
xirtaM
```

*Moguće rješenje:*

---

```
#include <iostream>  
#include <string>  
#include <vector>  
using namespace std;  
  
int main() {  
    vector<string> filmovi = { "Matrix", "Star Wars", "Finding Nemo" };  
  
    for (int i = filmovi.size() - 1; i >= 0; i--) {  
        string trenutni_film = filmovi[i];  
  
        for (int j = trenutni_film.length() - 1; j >= 0; j--) {  
            cout << trenutni_film[j];  
        }  
        cout << endl;  
    }  
  
    return 0;  
}
```

## Zadatak 4

Napišite program koji sadrži dva polja. Neka jedno polje sadrži sljedeće pridjeve: Amusing, Awesome, Beautiful, Good i Happy. Neka drugo polje sadrži sljedeće imenice: dog, cat, bird i cloud. Neka vaš program ispiše jedan slučajni pridjev s jednom slučajnom imenicom. Ponavljajte onoliko dugo koliko korisnik želi. Primjer ispisa programa može biti:

```
Good dog
Zelite li ispisati jos jednu frazu (1=da, 0=ne): 1
Beautiful cat
Zelite li ispisati jos jednu frazu (1=da, 0=ne): 1
Good cloud
Zelite li ispisati jos jednu frazu (1=da, 0=ne): 1
Good dog
Zelite li ispisati jos jednu frazu (1=da, 0=ne): 1
Amusing dog
Zelite li ispisati jos jednu frazu (1=da, 0=ne): 0
```

*Moguće rješenje:*

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

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

    string pridjevi[5] = { "Amusing", "Awesome", "Beautiful", "Good", "Happy" };
    string imenice[4] = { "dog", "cat", "bird", "cloud" };

    bool dalje;

    do {
        // Posto ima 5 pridjeva, ispravni indeksi su 0, 1, 2, 3 i 4
        int slucajni_indeks_pridjevi = 0 + rand() % (4 - 0 + 1);

        // Posto ima 4 imenice, ispravni indeksi su 0, 1, 2 i 3
        int slucajni_indeksi_imenice = 0 + rand() % (3 - 0 + 1);

        // Print
        cout << pridjevi[slucajni_indeks_pridjevi] << " " <<
        imenice[slucajni_indeksi_imenice] << endl;

        // Pitamo korisnika zeli li dalje
        cout << "Zelite li ispisati jos jednu frazu (1=da, 0=ne): ";
        cin >> dalje;
    } while (dalje);

    return 0;
}
```

## Zadatak 5

Napišite program koji korisniku nudi sljedeće opcije, sve dok korisnik to želi:

- Generiranje pet slučajnih godina od 1995. do 2032.
- Prikaz svih generiranih godina.

Primjer prikaza programa:

```
A = Generiranje slučajnih godina
B = Ispis svih generiranih godina
X = Izlaz
> A
> A
> B
2006 2025 2024 2024 2001 2027 2020 2009 1997 1998
> A
> B
2006 2025 2024 2024 2001 2027 2020 2009 1997 1998 2021 2005 1996 2015 2005
> X
```

*Moguće rješenje:*

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

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

    cout << "A = Generiranje slučajnih godina" << endl;
    cout << "B = Ispis svih generiranih godina" << endl;
    cout << "X = Izlaz" << endl;

    char izbor;
    int slučajna_godina;
    vector<int> generirane_godine;

    do {
        cout << "> ";
        cin >> izbor;

        switch (izbor) {
            case 'A':
                for (int i = 0; i < 5; i++) {
                    slučajna_godina = 1995 + rand() % (2032 - 1995 + 1);
                    generirane_godine.push_back(slučajna_godina);
                }
                break;
            case 'B':
                for (int i = 0; i < generirane_godine.size(); i++) {
                    cout << generirane_godine[i] << " ";
                }
                cout << endl;
                break;
        }
    }
}
```

```
    } while (izbor != 'X');  
    return 0;  
}
```

## Zadatak 6

Napišite program koji simulira miješanje pjesama. Prvo stvorite polje s 5 imena pjesama u njemu. Zatim zamijenite dvije slučajne pjesme u polju i nastavite tako mijenjati slučajne pjesme 100 puta. Na kraju, prikažite sve pjesme iz polja.

*Moguće rješenje:*

---

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

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

    string pjesme[5] = { "Fade to black", "A moment of clarity", "One", "Enter
sandman", "Story to tell" };

    int ri1;
    int ri2;
    string temp;
    int preostalo = 100;

    do {
        ri1 = 0 + rand() % (4 - 0 + 1);
        ri2 = 0 + rand() % (4 - 0 + 1);

        if (ri1 != ri2) {
            // Zamijenimo ih
            temp = pjesme[ri1];
            pjesme[ri1] = pjesme[ri2];
            pjesme[ri2] = temp;

            preostalo--;
        }
    } while (preostalo > 0);

    for (int i = 0; i < 5; i++) {
        cout << pjesme[i] << endl;
    }
    cout << endl;

    return 0;
}
```



## Zadatak 7

Napišite program koji simulira mehanizam pretvaranja RNA sekvence u proteine. Neka program traži od korisnika da unese RNA sekvencu (koju čine niz slova U, C, G i A). Zatim ispišite niz proteina koji će biti kreiran, na temelju sljedećih pravila prevođenja:

- AUG: početak – ignorirajte sva prethodna slova.
- UUU: će kreirati fenilalanin
- CAA: će kreirati glutamin
- GCA: će kreirati alanin
- ACC: će kreirati treonin
- UAA, UAG, UGA: kraj – kraj procesiranja, ignorirajte sva sljedeća slova

Ako u sekvenci postoji početak, možete biti sigurni da će kraj također biti prisutan, te da ću između biti ispravne kombinacije slova. Primjer valjanih ulaznih RNA sekvenci su:

AUGACCUAA – prevodi se kao početak, treonin, kraj

AUGGCAACCUGA – prevodi se kao početak, alanin, treonin, kraj

MEOWAUGACCUAAWOOF – MEOW i WOOF se zanemaruju pa se prevodi kao početak, treonin, kraj

MEOWMOEWWOOF – prevodi se kao "nije pronađen AUG".

*Moguće rješenje:*

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

int main() {
    string rna;
    cout << "Unesite RNA sekvencu: ";
    getline(cin, rna);

    // Trazimo AUG, moze biti bilo gdje u stringu (ili cak nigdje)
    int pocetni_indeks = -1;
    for (int i = 0; i < rna.length() - 2; i++) {
        if (rna[i] == 'A' && rna[i + 1] == 'U' && rna[i + 2] == 'G') {
            pocetni_indeks = i;
            break;
        }
    }

    // Nastavljamo samo ako smo nasli AUG
    if (pocetni_indeks > -1) {
        cout << "pocetak" << endl;

        for (int i = pocetni_indeks; i < rna.length() - 2; i += 3) {
            if (rna[i] == 'U' && rna[i + 1] == 'U' && rna[i + 2] == 'U') {
                cout << "fenilalanin" << endl;
            }
            else if (rna[i] == 'C' && rna[i + 1] == 'A' && rna[i + 2] ==
'A') {
                cout << "glutamin" << endl;
            }
            else if (rna[i] == 'G' && rna[i + 1] == 'C' && rna[i + 2] ==
'A') {
                cout << "alanin" << endl;
            }
        }
    }
}
```

```

    }
    else if (rna[i] == 'A' && rna[i + 1] == 'C' && rna[i + 2] ==
'C') {
        cout << "treonin" << endl;
    }
    else if (rna[i] == 'U' && rna[i + 1] == 'A' && rna[i + 2] == 'A'
||
        rna[i] == 'U' && rna[i + 1] == 'A' && rna[i + 2] == 'G' ||
        rna[i] == 'U' && rna[i + 1] == 'G' && rna[i + 2] == 'A') {
        cout << "kraj" << endl;
        break;
    }
}
}
else {
    cout << "nije pronadjen AUG" << endl;
}
return 0;
}

```

## Zadatak 8

Napišite program koji koristi jednostavnu tehniku šifriranja. Napravite dva polja: prvo treba sadržavati sva mala slova od a-z, a drugo također, ali obrnutim redoslijedom, od z-a. Zatim učitajte riječ od korisnika, pronađite svaki znak u prvom polju i ispišite znak na istom indeksu u drugom polju. Ako naidete na nepoznati znak, samo ga ispišite. Primjer rada programa:

Upisite recenicu koju zelite sifrirati: danas je lijep i sunčan dan  
wzmzh qv orqvk r hfmxzm wzm

*Moguće rješenje:*

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

int main() {
    // Pripremimo prvo polje
    char first[26];
    char c = 'a';
    for (int i = 0; i < 26; i++) {
        first[i] = c;
        c++;
    }

    // Pripremimo drugo polje
    char second[26];
    c = 'z';
    for (int i = 0; i < 26; i++) {
        second[i] = c;
        c--;
    }

    // Ucitamo recenicu
    string recenica;
    cout << "Upisite recenicu koju zelite sifrirati: ";
    getline(cin, recenica);

    // Sifriranje
    for (int i = 0; i < recenica.length(); i++) {
        char trenutno_slovo = recenica[i];

        // Pronadjemo u prvom polju
        bool pronadjeno = false;
        for (int j = 0; j < 26; j++) {
            if (first[j] == trenutno_slovo) {
                // Pronadjeno na indeksu j, ispisemo iz drugog polja
                cout << second[j];
                pronadjeno = true;
                break;
            }
        }

        // Nije pronadjeno u prvom polju, samo ga ispisemo
        if (!pronadjeno) {
            cout << trenutno_slovo;
        }
    }
}
```

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

## Zadatak 9

Napišite program koji stvara dva polja: jedan sadrži pet decimalnih brojeva, a drugi pet boolova. Za svaki indeks u drugom polju na kojem je true, uzmite broj iz prvog polja na istom indeksu i kopirajte ga u vektor. Na kraju ispišite sadržaj vektora na sljedeći način: neka svaki element bude uvučen za sljedeći broj razmaka: indeksu u vektoru \* 4. Na primjer:

```
12.8
   4.5
      12.001
```

*Moguće rješenje:*

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

int main() {
    double brojevi[5] = { 12.8, 4.5, -3.4, 12.001, 7.6 };
    bool treba_kopirati[5] = { true, true, false, true, false };
    vector<double> v;

    // Kopiramo sto treba
    for (int i = 0; i < 5; i++) {
        if (treba_kopirati[i]) {
            v.push_back(brojevi[i]);
        }
    }

    for (int i = 0; i < v.size(); i++) {
        // Uvlacenje
        for (int j = 0; j < 4 * i; j++) {
            cout << ' ';
        }

        cout << v[i] << endl;
    }

    return 0;
}
```

## Zadatak 10

Napišite program koji stvara slučajne prodajne vrijednosti za svaki mjesec u godini. Neka svaka prodajna vrijednost bude u rasponu od 1.000 dolara do 50.000 dolara. Nakon toga, prikažite trakasti graf gdje u svakom retku prikazujete broj mjeseca i jedan znak '#' za svakih punih 1.000 dolara iz tog mjeseca. Također, na kraju svakog retka prikažite prodajnu vrijednost. Na primjer:

```
Mjesec 1: ##### (21154 USD)
Mjesec 2: ##### (31945 USD)
Mjesec 3: ##### (27940 USD)
Mjesec 4: ##### (10703 USD)
Mjesec 5: ##### (31751 USD)
Mjesec 6: ##### (6249 USD)
Mjesec 7: ##### (18194 USD)
Mjesec 8: ##### (14551 USD)
Mjesec 9: ### (3772 USD)
Mjesec 10: ##### (24096 USD)
Mjesec 11: ##### (17583 USD)
Mjesec 12: ##### (31486 USD)
```

*Moguće rješenje:*

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

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

    int prodano[12];

    // Kreiranje slučajnih prodajnih vrijednosti
    for (int i = 0; i < 12; i++) {
        prodano[i] = 1000 + rand() % (50000 - 1000 + 1);
    }

    // Prikaz
    for (int i = 0; i < 12; i++) {
        cout << "Mjesec " << (i + 1) << ":\t";

        int koliko_hasheva = prodano[i] / 1000;

        for (int j = 0; j < koliko_hasheva; j++) {
            cout << '#';
        }
        cout << " (" << prodano[i] << " USD)" << endl;
    }

    return 0;
}
```

## Zadatak 11

Napišite program koji učitava 1D točke (to jest, cijele brojeve) dok to korisnik želi. Kada je unos gotov, ispišite taj 1D koordinatni sustav od 0 do 50, koristeći 'x' za označavanje korisnički definirane točke i '-' za označavanje praznog prostora. Primjer:

```
Upisite tocku: 50
Dodati jos jednu tocku (1=da/0=ne): 1
Upisite tocku: 4
Dodati jos jednu tocku (1=da/0=ne): 1
Upisite tocku: 40
Dodati jos jednu tocku (1=da/0=ne): 0
----x-----x-----x
```

*Moguće rješenje:*

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

int main() {
    vector<double> tocke;
    double temp;
    bool dalje;

    do {
        cout << "Upisite tocku: ";
        cin >> temp;
        tocke.push_back(temp);

        cout << "Dodati jos jednu tocku (1=da/0=ne): ";
        cin >> dalje;
    } while (dalje);

    for (int x = 0; x <= 50; x++) {
        bool postoji = false;

        for (int i = 0; i < tocke.size(); i++) {
            if (tocke[i] == x) {
                cout << 'x';
                postoji = true;
                break;
            }
        }

        if (!postoji) {
            cout << '-';
        }

        cout << endl;
    }

    return 0;
}
```

## Zadatak 12

Napišite program koji kreira tri polja cijelih brojeva: prvo polje treba sadržavati 5 cijelih brojeva, drugo također treba sadržavati 5 cijelih brojeva, a treće treba imati prostora za 10 cijelih brojeva i biti inicijalno prazno. Zatim kopirajte elemente iz prvog u treće polje, a potom i iz drugog u treće polje. Na taj način treće polje treba sadržavati sve vrijednosti iz prva dva polja. Na kraju, prikažite sadržaj trećeg polja.

*Moguće rješenje:*

---

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

int main() {
    int prvo[5] = { 10, 20, 30, 40, 50 };
    int drugo[5] = { 60, 70, 80, 90, 100 };
    int trece[10];

    for (int i = 0; i < 5; i++) {
        trece[i] = prvo[i];
    }

    for (int i = 0; i < 5; i++) {
        trece[i + 5] = drugo[i];
    }

    for (int i = 0; i < 10; i++) {
        cout << trece[i] << " ";
    }

    return 0;
}
```



## Zadatak 13

Napišite program koji učitava kategoričke vrijednosti (kao što su "vrlo loše", "loše", "dobro", "vrlo dobro" itd.) koliko to korisnik želi. Nakon toga, učitate od korisnika jednu kategoričku vrijednost X i zatim prikažite sve prethodno učitane vrijednosti na sljedeći način: ako je vrijednost jednaka X, prikažite 1; u suprotnom, prikažite 0. Primjer programa:

```
Upisite kategoricku vrijednost: dobro
Dodati jos jednu (1=da, 0=ne): 1
Upisite kategoricku vrijednost: lose
Dodati jos jednu (1=da, 0=ne): 1
Upisite kategoricku vrijednost: dobro
Dodati jos jednu (1=da, 0=ne): 1
Upisite kategoricku vrijednost: vrlo dobro
Dodati jos jednu (1=da, 0=ne): 0
Upisite kategoricku vrijednost X: dobro
1010
```

*Moguće rješenje:*

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

int main() {
    vector<string> vrijednosti;
    string temp;
    bool dalje;

    do {
        cout << "Upisite kategoricku vrijednost: ";
        getline(cin, temp);
        vrijednosti.push_back(temp);

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

    string trazi;
    cout << "Upisite kategoricku vrijednost X: ";
    getline(cin, trazi);

    for (int i = 0; i < vrijednosti.size(); i++) {
        if (vrijednosti[i] == trazi) {
            cout << "1";
        }
        else {
            cout << "0";
        }
    }
    cout << endl;

    return 0;
}
```

## Zadatak 14

Napišite program za rad s jednostavnim upitnim jezikom (BSQL). Jezik se koristi za zadavanje upita nad bazom podataka: vektorom znakova. Neka ovaj jezik sadrži tri naredbe: SELECT, INSERT i UPDATE:

- SELECT – dohvaća sve vrijednosti iz pohrane podataka i prikazuje ih.
- INSERT <vrijednost> – dodati vrijednost u bazu podataka.
- UPDATE <vrijednost> TO <nova-vrijednost> – promijeniti sve vrijednosti u bazi podataka na novu.

Iskoristite činjenicu da je vrijednost uvijek jedan znak i da će korisnik uvijek ispravno unijeti jednu od tri moguće naredbe. U beskonačnoj petlji dopustite korisniku da manipulira bazom podataka. Na primjer:

Koristite sljedeće naredbe:

```
SELECT
INSERT <vrijednost>
UPDATE <vrijednost> TO <nova-vrijednost>
> INSERT a
> INSERT b
> SELECT
a b
> INSERT c
> INSERT a
> SELECT
a b c a
> UPDATE a TO z
> SELECT
z b c z
...
```

*Moguće rješenje:*

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

int main() {
    vector<char> baza_podataka;
    string naredba;
    char stara_vrijednost;
    char nova_vrijednost;

    cout << "Koristite sljedeće naredbe:" << endl;
    cout << "SELECT" << endl;
    cout << "INSERT <vrijednost>" << endl;
    cout << "UPDATE <vrijednost> TO <nova-vrijednost>" << endl;

    while (true) {
        cout << "> ";
        getline(cin, naredba);

        if (naredba == "SELECT") { // select
            for (int i = 0; i < baza_podataka.size(); i++) {
                cout << baza_podataka[i] << " ";
            }
            cout << endl;
        }
    }
}
```

```
    }
    else if (naredba[0] == 'I') { // insert
        nova_vrijednost = naredba[7];
        baza_podataka.push_back(nova_vrijednost);
    }
    else { // update
        stara_vrijednost = naredba[7];
        nova_vrijednost = naredba[12];

        for (int i = 0; i < baza_podataka.size(); i++) {
            if (baza_podataka[i] == stara_vrijednost) {
                baza_podataka[i] = nova_vrijednost;
            }
        }
    }
}

return 0;
}
```

## Zadatak 15

Napišite program koji simulira igru pogađanja riječi. Definirajte popis riječi napisanih malim slovima, a zatim odaberite slučajnu riječ. Dopustite korisniku da pogađa znakove dok ne pogodi sve. Nakon svakog unosa znaka, prikažite riječ tako da ispišete sve pogođene znakove, a nepogođene prikažite kao '\_'. Ispišite broj pokušaja koji je bio potreban da se pogodi cijela riječ. Na primjer:

```
Pogodite slovo: a
_a_a_a_
Pogodite slovo: c
_a_a_ac
Pogodite slovo: f
_a_a_ac
Pogodite slovo: m
ma_a_ac
Pogodite slovo: g
maga_ac
Pogodite slovo: r
magarac
Pogodili ste u 6 pokušaja
```

*Moguće rješenje:*

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

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

    string rijeci[5] = { "integral", "brokula", "magarac", "prevjes", "studeni"
};

    // Odaberemo slučajnu riječ
    int ri = 0 + rand() % (4 - 0 + 1);
    string odabrana_rijec = rijeci[ri];
    cout << odabrana_rijec;
    vector<char> pokusaji;
    char temp;
    int broj_pokusaja = 0;

    while (true) {
        // Novi pokušaj
        cout << "Pogodite slovo: ";
        cin >> temp;
        pokusaji.push_back(temp);
        broj_pokusaja++;

        // Ispis rijeci
        int pogodjeno_slova = 0;
        for (int i = 0; i < odabrana_rijec.length(); i++) {
            char trenutno_slovo = odabrana_rijec[i];
            bool pronadjeno = false;
```

```

        for (int j = 0; j < pokusaji.size(); j++) {
            if (pokusaji[j] == trenutno_slovo) {
                pronadjeno = true;
                pogodjeno_slova++;
                break;
            }
        }

        if (pronadjeno) {
            cout << trenutno_slovo;
        }
        else {
            cout << '_';
        }
    }
    cout << endl;

    // Sve pogodjeno?
    if (pogodjeno_slova == odabrana_rijec.length()) {
        cout << "Pogodili ste u " << broj_pokusaja << " pokusaja" <<
endl;
        break;
    }
}

return 0;
}

```

## Zadatak 16

Napišite program koji od korisnika traži da unese telefonski broj kao niz od 10 znamenki. Zatim ispišite formatiranu verziju istog broja. Nakon prve tri znamenke ispišite '/', a nakon sljedeće 4 znamenke ispišite '-', kao u primjeru:

Upisite telefonski broj: 0998902314

Formatirani telefonski broj: 099/8902-314

*Moguće rješenje:*

---

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

int main() {
    string broj;
    cout << "Upisite telefonski broj: ";
    getline(cin, broj);

    cout << "Formatirani telefonski broj: ";
    for (int i = 0; i < broj.length(); i++) {
        cout << broj[i];

        if (i == 2) {
            cout << '/';
        }
        else if (i == 6) {
            cout << '-';
        }
    }

    return 0;
}
```

## Zadatak 17

Napišite program koji od korisnika učitava rečenicu i slovo koje treba pronaći. Zatim pronađite sva pojavljivanja tog slova u rečenici i ispišite njihove indekse. Ako nema nijednog pojavljivanja, ispišite odgovarajuću poruku. Na primjer:

```
Upisite recenicu: ovo je primjer jednostavne recenice
Upisite slovo koje zelite pronaci: e
Pronadjeno na indeksu: 5
Pronadjeno na indeksu: 12
Pronadjeno na indeksu: 16
Pronadjeno na indeksu: 25
Pronadjeno na indeksu: 28
Pronadjeno na indeksu: 30
Pronadjeno na indeksu: 34
```

*Moguće rješenje:*

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

int main() {
    string recenica;
    cout << "Upisite recenicu: ";
    getline(cin, recenica);

    char trazi;
    cout << "Upisite slovo koje zelite pronaci: ";
    cin >> trazi;

    int pojavljivanja = 0;

    for (int i = 0; i < recenica.length(); i++) {
        if (recenica[i] == trazi) {
            cout << "Pronadjeno na indeksu: " << i << endl;
            pojavljivanja++;
        }
    }

    if (pojavljivanja == 0) {
        cout << "Nije pronadjeno" << endl;
    }

    return 0;
}
```

## Zadatak 18

Napišite program koji komprimira niz znakova. Kompresija se izvodi zamjenom svakog niza nula s količinom nula u nizu, navedenim u zagradama. Sve ostale znakove treba samo ispisati. Neka program traži od korisnika da unese niz znakova i onda mu ispiše komprimiranu verziju. Na primjer:

Upisite niz: 2000006300250000302090000000

Komprimirano: 2(5)63(2)25(4)3(1)2(1)9(7)

*Moguće rješenje:*

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

int main() {
    string niz;
    cout << "Upisite niz: ";
    getline(cin, niz);

    bool brojim_nule = false;
    int pojavljivanja = 0;

    cout << "Komprimirano: ";
    for (int i = 0; i < niz.length(); i++) {
        if (niz[i] == '0') {
            if (!brojim_nule) {
                // Pronasli smo prvu nulu u sekvenci
                brojim_nule = true;
            }
            pojavljivanja++;
        }
        else {
            if (brojim_nule) {
                // Upravo smo prestali brojati nule
                brojim_nule = false;
                cout << '(' << pojavljivanja << ')';
                pojavljivanja = 0;
            }

            cout << niz[i];
        }
    }

    if (pojavljivanja > 0) {
        cout << '(' << pojavljivanja << ')';
    }

    return 0;
}
```



## Zadatak 19

Napišite program koji kreira vektor elemenata i provjerava je li taj niz elemenata strogo rastući (strogo rastući znači da je svaki element veći od prethodnika). Na primjer, ovaj niz je strogo rastući: 2 3 5 8 9 10 15 47, a ovaj nije: 2 3 5 4 9 10 15 47.

*Moguće rješenje:*

---

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

int main() {
    vector<int> niz = { 2, 4, 6, 8, 10, 12, 11, 16 };

    bool strogo_rastuci = true;

    for (int i = 0; i < niz.size() - 1; i++) {
        if (niz[i] >= niz[i + 1]) {
            strogo_rastuci = false;
            break;
        }
    }

    if (strogo_rastuci) {
        cout << "Niz je strogo rastuci" << endl;
    }
    else {
        cout << "Niz nije strogo rastuci" << endl;
    }

    return 0;
}
```

## Zadatak 20

Napišite program koji stvara dva polja, jedno sadrži 8 imena, a drugo 8 cijelih brojeva. Zatim prikažite imena iz prvog polja uvučena onoliko razmaka koliko piše na istom indeksu u drugom polju.

*Moguće rješenje:*

---

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

int main() {
    string imena[8] = {"Ian", "Zack", "Lana", "Ivo", "Leo", "Iva", "Ana", "Mia"};
    int uvuci[8] = { 0, 2, 4, 5, 4, 6, 2, 0 };

    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < uvuci[i]; j++) {
            cout << ' ';
        }
        cout << imena[i] << endl;
    }

    return 0;
}
```