

# PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 6

2023-2024.

## Zadatak 1

Ručno napravite datoteku „imena.txt“ i u nju upišite proizvoljni broj imena. Napišite program koji iz te datoteke učitava sva imena i izračunava prosjek znakova u njima. Nakon toga, imena s brojem znakova manjim od prosjeka upišite u datoteku „kratka\_imena.txt“, a ostala u datoteku „duga\_imena.txt“.

*Moguće rješenje:*

---

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

int main() {
    vector<string> sva_imena;

    // prvo sva imena ubacimo u vektor i usput izračunamo prosjek duljina
    ifstream dat_in("imena.txt");
    if (!dat_in) {
        cout << "Greska pri otvaranju datoteke imena.txt" << endl;
        return 1;
    }

    double suma = 0.0;
    string ime;

    while (getline(dat_in, ime)) {
        sva_imena.push_back(ime);
        suma += ime.length();
    }

    // zatvorimo datoteku jer smo iz nje pročitali što nam je trebalo
    dat_in.close();

    double prosjek = suma / sva_imena.size();
    cout << "Prosjek je: " << prosjek << endl;

    // sad imena upišemo u datoteke
    ofstream dat_duga("duga_imena.txt");
    if (!dat_duga) {
        cout << "Greska pri otvaranju datoteke duga_imena.txt" << endl;
        return 2;
    }

    ofstream dat_kratka("kratka_imena.txt");
    if (!dat_kratka) {
        cout << "Greska pri otvaranju datoteke kratka_imena.txt" << endl;
        dat_duga.close();
        return 3;
    }

    for (int i = 0; i < sva_imena.size(); i++) {
        if (sva_imena[i].length() < prosjek) {
            dat_kratka << sva_imena[i] << endl;
        }
    }
}
```

```
        else {
            dat_duga << sva_imena[i] << endl;
        }
    }

    dat_kratka.close();
    dat_duga.close();
    return 0;
}
```

## Zadatak 2

Ručno napravite tekstualnu datoteku „pravokutnici.txt“ i u svaki redak upišite po 2 cijela broja odvojena razmakom (redaka neka bude proizvoljna količina). U svakom retku prvi broj označava širinu, a drugi visinu pravokutnika. Pripremite strukturu Pravokutnik te učitajte iz datoteke podatke i napravite onoliko objekata koliko ima redaka u datoteci (svaki redak je jedan objekt). Nakon što su svi pravokutnici učitani i datoteka zatvorena, ispišite površine svih pravokutnika.

*Moguće rješenje:*

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

struct Pravokutnik {
    int sirina;
    int visina;
};

int main() {
    vector<Pravokutnik> pravokutnici;

    ifstream dat_in("pravokutnici.txt");
    if (!dat_in) {
        cout << "Greska pri otvaranju datoteke pravokutnici.txt" << endl;
        return 1;
    }

    Pravokutnik p;
    while (dat_in >> p.sirina) {
        dat_in >> p.visina;
        pravokutnici.push_back(p);
    }

    dat_in.close();

    for (int i = 0; i < pravokutnici.size(); i++) {
        cout << pravokutnici[i].sirina * pravokutnici[i].visina << endl;
    }

    return 0;
}
```

### Zadatak 3

Napišite program koji u vektor stavlja 10.000 slučajnih brojeva između 1 i 10.000. Nakon toga sortirajte brojeve funkcijom sort (<https://cplusplus.com/reference/algorithm/sort/>) pa ih zapišite u datoteku „brojevi.txt“ odvojene razmacima.

*Moguće rješenje:*

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

int main() {
    srand(time(nullptr));
    vector<int> brojevi;

    for (int i = 0; i < 10000; i++) {
        brojevi.push_back(1 + rand() % (10000 - 1 + 1));
    }

    sort(brojevi.begin(), brojevi.end());

    ofstream dat("brojevi.txt");
    if (!dat) {
        cout << "Greska pri otvaranju datoteke brojevi.txt" << endl;
        return 1;
    }

    for (int i = 0; i < brojevi.size(); i++) {
        dat << brojevi[i] << endl;
    }

    dat.close();
    return 0;
}
```

## Zadatak 4

Ručno kreirajte datoteku „imena.txt“ i u svaki red upišite po jedno ime. Napišite program koji sva imena iz te datoteke prepisuje u drugu datoteku, ali u drukčijem formatu: umjesto da svako ime bude u svom retku, neka sva imena budu u istom retku, ali ispred svakog imena prvo upišite broj znakova. Primjerice, ako prva datoteka sadrži:

Ana  
Pero

Onda druga datoteka treba sadržavati:

3Ana4Pero

*Moguće rješenje:*

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

int main() {
    ifstream dat_in("imena.txt");
    if (!dat_in) {
        cout << "Greska pri otvaranju datoteke imena.txt" << endl;
        return 1;
    }

    ofstream dat_out("imena_drugi_format.txt");
    if (!dat_out) {
        cout << "Greska pri otvaranju datoteke imena_drugi_format.txt" << endl;
        dat_in.close();
        return 2;
    }

    string ime;
    while (getline(dat_in, ime)) {
        dat_out << ime.length() << ime;
    }

    dat_in.close();
    dat_out.close();
    return 0;
}
```

## Zadatak 5

Napišite program koji od korisnika učitava širinu i visinu ekrana i u datoteku upisuje znakove ('G', 'R', 'W') prema pravilima:

- Prvi znak u prvom retku je uvijek 'G'.
- Kod kreiranja svakog sljedećeg znaka neka je 50% šansa da će ostati jednak znak kao i prethodni, a neka je 25% šansa za preostala dva znaka.

Znakove spremite u vektor, a zatim ih zapišite u datoteku. Primjerice, ako je širina 50, a visina 16, u datoteku treba zapisati:

```
GGWWWGGGRRRRRRRWRGGWWRWWWGWWWWGGRRRRWWWGRRRRGGGW
GRRGRWRWWWGWWWWWRRRRGGRRRRWRRWWWGRRRRWWWGGGGGW
RRRRRRWGWWGGGGGGRRWGWWWWWRRRRRRRRGGGGWRRGGWRRGGR
WWWWWGWWGGGWWRWWWGWWGGGGWRRGWWRWWWRRWWWWWWWRWGWWG
RRRWGWRWRWWWGGGGWGGGWRRWGGRWGWRRRRWRRWWWRGWRRG
WWWRWGGRRGGGGGGRRRRRRGGRRRRRRWRRRRRRRRWWWWWW
WRRRRGGGGRRGGRWWRRRRRRRGGRRWGWWGGGGWRRGGRRWGGRG
WRWGGRRGGWGRGGGGRRRRGGWRWWWGRRGGGWRRWRWWWGRG
GRRWGRWWWGGGGGGWGGWGGGGWRRWWWGWWGGGGWRRRRWRGR
WRRRWGGGWGWWGWWRRWWWGWWRRWGGGRGRWGWRRWGWRRWWW
RRWGWGRWRWRRGRGWWWGGRRRRGGGGWWWRRGGRRWGGRWWW
WGWWWGRGRWRWGWWGGGGWWWGRRGGGGRRRRRRRRRRGGGG
GRWRRRRRGWWRGWWWWWGGWWRRRRRWRRWWWGGRRGGGGRRGGGG
GRRGWWWRRRRRRGRRRWGWRRWRRGGWGWGGGGGGWGGRRRR
GRGGRRRRRRWWWRRGGGGGGWRRRRRRWRRWRRGGRRGWRRWWWGGGG
GGWRRRRRRRRWGGRRRRGWRRRRWWWWWWRRRRRRRRWWWRRRRWWW
```

*Moguće rješenje:*

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

char daj_sljedeci(char trenutni_znak) {
    const int n = 3;
    char znakovi[n] = { 'G', 'R', 'W' };

    // generiram slučajni broj između 1 i 100:
    // ako je između 1 i 50, vraćam trenutni znak
    // ako je između 51 i 75, vraćam jedan od preostala dva
    // ako je između 75 i 100, vraćam preostali
    int slucaj = 1 + rand() % (100 - 1 + 1);

    if (slucaj >= 1 && slucaj <= 50) {
        return trenutni_znak;
    }
    else if (slucaj >= 51 && slucaj <= 75) {
        // prvi preostali
        for (int i = 0; i < n; i++) {
            if (znakovi[i] != trenutni_znak) {
                return znakovi[i];
            }
        }
    }
}
```

```

        }
    }
    else {
        // drugi preostali
        for (int i = n - 1; i >= 0; i--) {
            if (znakovi[i] != trenutni_znak) {
                return znakovi[i];
            }
        }
    }
}

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

    int sirina;
    cout << "Sirina: ";
    cin >> sirina;

    int visina;
    cout << "Visina: ";
    cin >> visina;

    vector<char> znakovi;
    char trenutni_znak = 'G';
    znakovi.push_back(trenutni_znak);

    // prvo generiramo znakove u vektor
    for (int i = 0; i < sirina * visina - 1; i++) {
        trenutni_znak = daj_sljedeci(trenutni_znak);
        znakovi.push_back(trenutni_znak);
    }

    // pa ih zapišemo u datoteku
    ofstream dat("mapa.txt");
    if (!dat) {
        cout << "Greska pri otvaranju datoteke mapa.txt" << endl;
        return 1;
    }

    for (int i = 0; i < znakovi.size(); i++) {
        dat << znakovi[i];

        // ako je kraj retka, idemo u novi
        if ((i + 1) % sirina == 0) {
            dat << endl;
        }
    }

    dat.close();
    return 0;
}

```



## Zadatak 6

Kreirajte ručno dvije datoteke i u obje upišite decimalnih brojeva prema želji odvojene razmakom (ne mora biti jednaka količina brojeva u obje). Napišite program koji učitava brojeve iz obje datoteke u jedan vektor. Nakon toga, pronađite i ispišite najmanji i najveći broj iz vektora. Kreirajte dodatne funkcije prema želji.

*Moguće rješenje:*

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

void ucitaj_u_vektor(ifstream& dat, vector<double>& v) {
    double broj;
    while (dat >> broj) {
        v.push_back(broj);
    }
}

int main() {
    vector<double> brojevi;

    // učitamo prvu
    ifstream dat1("prva.txt");
    if (!dat1) {
        cout << "Greska pri otvaranju datoteke prva.txt" << endl;
        return 1;
    }

    ucitaj_u_vektor(dat1, brojevi);

    dat1.close();

    // učitamo drugu
    ifstream dat2("druga.txt");
    if (!dat2) {
        cout << "Greska pri otvaranju datoteke druga.txt" << endl;
        return 2;
    }

    ucitaj_u_vektor(dat2, brojevi);

    dat2.close();

    // odradimo posao
    double najmanji = brojevi[0];
    double najveći = brojevi[0];

    for (int i = 0; i < brojevi.size(); i++) {
        if (brojevi[i] < najmanji) {
            najmanji = brojevi[i];
        }

        if (brojevi[i] > najveći) {
```

```
        najveci = brojevi[i];
    }
}

cout << "Najmanji je: " << najmanji << endl;
cout << "Najveci je: " << najveci << endl;

return 0;
}
```

## Zadatak 7

Napišite program koji od korisnika učitava tekst na engleskom jeziku sastavljen samo od malih slova, a zatim u datoteku upisuje koliko se puta koje malo slovo engleske abecede pojavljuje u tekstu, primjerice:

Slovo	Broj pojavljivanja
a	24
b	6
c	0
...	
y	3
z	1

*Moguće rješenje:*

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

struct Pojava {
    char znak;
    int kolicina;
};

void azuriraj_kolicinu(char c, vector<Pojava>& v) {
    for (int i = 0; i < v.size(); i++) {
        if (v[i].znak == c) {
            v[i].kolicina++;
            return;
        }
    }
}

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

    // u vektor dodamo sve znakove s količinom 0
    vector<Pojava> v;
    Pojava p;
    for (char c = 'a'; c <= 'z'; c++) {
        p.znak = c;
        p.kolicina = 0;
        v.push_back(p);
    }

    // sad ažuriramo znakove u vektoru
    for (int i = 0; i < tekst.length(); i++) {
        azuriraj_kolicinu(tekst[i], v);
    }

    // upišemo u datoteku
    ofstream dat("pojavljivanja.txt");
```

```
if (!dat) {
    cout << "Greska pri otvaranju datoteke pojavljivanja.txt" << endl;
    return 1;
}

dat << "Slovo\tBroj pojavljivanja" << endl;
for (int i = 0; i < v.size(); i++) {
    dat << v[i].znak << "\t" << v[i].kolicina << endl;
}

dat.close();
return 0;
}
```

## Zadatak 8

Ručno napravite datoteku i u nju upišite nekoliko redaka. U svaki redak prvo upišite jedan bool (0 ili 1), a zatim odvojen razmakom jedan broj, primjerice:

```
0 12
1 7
1 21
0 85
1 3
```

Nakon toga, napišite program koji će ispisati zbroj onih brojeva iz datoteke kod kojih je bool jednak 1 (u slučaju gornje datoteke, treba ispisati 31).

*Moguće rješenje:*

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

int main() {
    ifstream dat("podaci.txt");
    if (!dat) {
        cout << "Greska pri otvaranju datoteke podaci.txt" << endl;
        return 1;
    }

    int suma = 0;
    bool podatak1;
    int podatak2;

    while (dat >> podatak1) {
        dat >> podatak2;

        if (podatak1) {
            suma += podatak2;
        }
    }

    cout << "Suma je: " << suma << endl;

    dat.close();
    return 0;
}
```

## Zadatak 9

Ručno kreirajte datoteku i u nju upišite jedan broj koji predstavlja *seed* za generator slučajnih brojeva. Napišite program koji inicijalizira generator slučajnih brojeva na *seed* iz datoteke. Kreirajte 10 slučajnih brojeva i spremite ih u drugu datoteku.

*Moguće rješenje:*

---

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

int main() {
    // učitamo seed
    ifstream dat1("seed.txt");
    if (!dat1) {
        cout << "Greska pri otvaranju datoteke seed.txt" << endl;
        return 1;
    }

    int seed;
    dat1 >> seed;

    dat1.close();

    // postavimo seed
    srand(seed);

    // spremimo u datoteku 10 brojeva
    ofstream dat2("slucajni_brojevi.txt");
    if (!dat2) {
        cout << "Greska pri otvaranju datoteke slucajni_brojevi.txt" << endl;
        return 1;
    }

    for (int i = 0; i < 10; i++) {
        dat2 << rand() << endl;
    }

    dat2.close();
    return 0;
}
```

## Zadatak 10

Kreirajte ručno datoteku i u svaki red upišite neki tekst. Napišite program koji u drugu datoteku prepisuje sve retke iz prve, ali od zadnjeg prema prvom. Primjerice, ako u prvoj datoteci piše:

Vectors are sequence containers representing arrays that can change in size.  
Just like arrays, vectors use contiguous storage locations for their elements.  
Internally, vectors use a dynamically allocated array to store their elements.

Onda u drugoj treba pisati:

Internally, vectors use a dynamically allocated array to store their elements.  
Just like arrays, vectors use contiguous storage locations for their elements.  
Vectors are sequence containers representing arrays that can change in size.

*Moguće rješenje:*

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

int main() {
    vector<string> recenice;

    ifstream dat1("recenice.txt");
    if (!dat1) {
        cout << "Greska pri otvaranju datoteke recenice.txt" << endl;
        return 1;
    }

    string recenica;
    while (getline(dat1, recenica)) {
        recenice.push_back(recenica);
    }

    dat1.close();

    ofstream dat2("recenice_obrnuto.txt");
    if (!dat2) {
        cout << "Greska pri otvaranju datoteke recenice_obrnuto.txt" << endl;
        return 1;
    }

    for (int i = recenice.size() - 1; i >= 0; i--) {
        dat2 << recenice[i] << endl;
    }

    dat2.close();
    return 0;
}
```