



Zadaća 1

Tehnike programiranja

Student: **Daris Mujkić**

Ocjena: **3.59**

Grupa: **RI1-2b**

Broj indeksa: **19413**

Potpis:

Potpis tutora:

Zadatak . 1.cpp

```
//TP 2022/2023: Zadaća 1, Zadatak 1
#include <cstdlib>
#include <iostream>
#include <cmath>
#include <limits>
#include <stdexcept>
#include <array>
#include <vector>
enum class TretmanNegativnih {IgnorirajZnak, Odbaci, PrijaviGresku};
typedef std::array<std::vector<long long int>, 10> matrica;

long long int MultiplikativniDigitalniKorijen(long long int broj, int B)
{
    long long int MDK=1;
    if (broj==std::numeric_limits<long long int>::min())
    {
        while (broj!=0)
        {
            MDK=MDK*(broj%B);
            broj=broj/B;
        }
        broj=std::abs(MDK);
    }
    if (broj==0) return 0;
    if (broj<0) broj=std::abs(broj);
    if (B<2) throw std::domain_error("Neispravna baza");

    for(;;) //679 u bazi 10 je 6
    {
        while (broj!=0)
        {
            MDK=MDK*(broj%B); // 9 // 9*7 // 9*7*6=378
            broj=broj/B;      // 67 // 6 // 0
        }
        if (MDK>=B)
        {
            broj=MDK;
            MDK=1;
        }
        else if (MDK<B) return MDK;
    }
}

matrica RazvrstajBrojeve(std::vector<long long int>v, TretmanNegativnih IOP)
{
    matrica m;
    for(int i=0;i<v.size();i++)
    {
        if (IOP==TretmanNegativnih::IgnorirajZnak)
        {
            if (v.at(i)<0) v.at(i)=std::abs(v.at(i));
        }
        if (IOP==TretmanNegativnih::Odbaci)
        {

```

```
        if (v.at(i)<0) i++;
    }
    if (IOP==TretmanNegativnih::PrijavaGresku)
    {
        if (v.at(i)<0) throw std::domain_error(
"Nije predvidjeno razvrstavanje negativnih brojeva");
    }
    long long int MDK=MultiplikativniDigitalniKorijen(v.at(i), 10); //baza 10
    for (int j=0;j<10;j++)
    {
        if (MDK==j)
        {
            m.at(j).push_back(v.at(i));
        }
    }
}
return m;
}

int main ()
{
    try
    {
        TretmanNegativnih IOP=TretmanNegativnih::IgnorirajZnak;
        std::cout<<"Unesite brojeve (bilo koji ne-broj oznacava kraj): ";
        std::vector<long long int>vec;
        for(;;)
        {
            long long int element;
            std::cin>>element;
            if (std::cin) vec.push_back(element);
            if (element<0) throw std::domain_error(
"Nije podrzano razvrstavanje negativnih brojeva!");
            if (!std::cin)
            {
                std::cin.clear(); //Clear, IIgnore, CIG, CIGARA
                std::cin.ignore(10000, '\n'); //izvinjavanje toku CIGARA metoda
                break;
            }
        }

        matrica m=RazvrstajBrojeve(vec, IOP);
        std::cout<<
"Rezultati razvrstavanja po multiplikativnom digitalnom korijenu:"<<std::endl;
        for (int i=0;i<10;i++)
        {
            if (m.at(i).size()!=0)
            {
                std::cout<<i<<" ";
                for (int j=0;j<m.at(i).size();j++)
                {
                    std::cout<<m.at(i).at(j)<<" ";
                }
                std::cout<<std::endl;
            }
        }
    }
}
```

```
        else continue;
    }
}

catch(std::domain_error e)
{
    std::cout<<e.what();
}

return 0;
}
```

Zadatak . 2.cpp

```
// TP 2022/2023: Zadatak 1, Zadatak 2
#include <cmath> //oko svakog piksela prozor (2n +1)(2n +1) N JE RED FILTERA PRIRODAN
BROJ
#include <iomanip>
#include <iostream>
#include <stdexcept>
#include <vector>

double aritmetikaprozora(std::vector<std::vector<double>> p) //PREKO VEKTORA PROBATI
{
    int brel = 0;
    double suma = 0;
    for (int i=0;i<p.size();i++) brel=brel+p.at(i).size();
    for (int i=0;i<p.size();i++)
    {
        for (int j=0;j<p.at(i).size();j++) //NECE RADITI ZADATAK ZA GRBAVE, PROBATI
        {
            suma=suma+p.at(i).at(j);
        }
    }
    double arsr = suma / brel;
    return arsr;
}

std::vector<std::vector<double>> UsrednjavajuciFilter(std::vector<std::vector<double>> m, int N)
{
    if (N < 0) throw std::domain_error("Neispravan red filtriranja");
    int brojredova = m.size();
    std::vector<std::vector<double>> novamatrica;
    novamatrica.resize(brojredova);
    for (int i=0;i<brojredova;i++) novamatrica.at(i).resize(m.at(i).size());

    // od i-N do i+N, od j-N do j+N, pa na to mjesto aritmeticku sred obuhvacenih

    for (int i = 0; i < brojredova; i++)
    {
        for (int j = 0; j < m.at(i).size(); j++)
        {
            std::vector<std::vector<double>> prozor;
            prozor.resize(2*N+1);
            int a=0;
            int pocetakreda = i - N;
            int krajreda = i + N;
            int pocetakkolone = j - N;
            int krajkolone = j + N;
            if (pocetakreda < 0) pocetakreda = 0;
            if (krajreda >= brojredova) krajreda = brojredova-1;
            if (pocetakkolone < 0) pocetakkolone = 0;
            if (krajkolone >= m.at(i).size()) krajkolone = m.at(i).size()-1;
            for (int k = pocetakreda; k <= krajreda; k++)
            {
                for (int l = pocetakkolone; l <= krajkolone; l++)
```

```
        prozor.at(a).push_back(m.at(k).at(l));
    }
    a++;
}
double arsr;
arsr = aritmetikaprozora(prozor);
novamatrica.at(i).at(j) = arsr;
}
}
return novamatrica;
}

int main() {
    try {
        std::cout << "Unesite broj redova i kolona matrice: " << std::endl;
        int brojredova, brojkolona;
        std::cin >> brojredova >> brojkolona;
        std::cout << "Unesite elemente matrice: " << std::endl;
        std::vector<std::vector<double>> m;
        m.resize(brojredova);
        for (int i = 0; i < brojredova; i++) m.at(i).resize(brojkolona);
        for (int i = 0; i < brojredova; i++)
        {
            for (int j = 0; j < brojkolona; j++)
            {
                int element;
                std::cin >> element;
                m.at(i).at(j) = element;
            }
        }
        std::cout << "Unesite red filtriranja: " << std::endl;
        int N;
        std::cin >> N;
        std::vector<std::vector<double>> nova = UsrednjavajuciFilter(m, N);
        std::cout << "Matrica nakon filtriranja: " << std::endl;
        for (int i = 0; i < brojredova; i++) {
            for (int j = 0; j < brojkolona; j++) {
                std::cout << std::fixed << std::setprecision(2) << std::setw(7) << nova.at(i)
.at(j);
            }
            std::cout << std::endl;
        }
    }

    catch (std::domain_error e) {
        std::cout << "GRESKA: " << e.what() << "!\n";
    }

    return 0;
}
```

Zadatak . 3.cpp

```
//TP 2022/2023: Zadaća 1, Zadatak 3
#include <iostream>
#include <cmath>
#include <vector>
#include <limits>
typedef std::vector<std::vector<int>> matrica;

bool DaLiJeSpiralnaMatrica (matrica m)
{
    int brojac=1;
    for (int i=0;i<m.size()-1;i++)
    {
        if (m.at(i).size()!=m.at(i+1).size()) return false;
        brojac++;
    }
    if (brojac==m.size()) return true;

    int brojredova;
    brojredova=m.size();
    int brojkolona;
    brojkolona=m.at(0).size();
    int k;
    k=m.at(0).at(0);
    bool smjer;
    if ((m.at(0).at(1))+1 == m.at(0).at(0)) smjer=true;
    if ((m.at(0).at(0))+1 == m.at(0).at(0)) smjer=false;

    if (smjer==true) //desna
    {
        int gornjagr=0;
        int donjagr=brojredova-1;
        int desnagr=brojkolona-1;
        int lijevagr=0;
        int i=0;
        int j=0;
        while (gornjagr<=donjagr && lijevagr<=desnagr)
        {
            while(j<desnagr-1)
            {
                m.at(gornjagr).at(j)=k;
                if (m.at(gornjagr).at(j+1) != (m.at(gornjagr).at(j))+1) return false;
                k++;
                j++;
            }
            gornjagr++;

            while(i<donjagr-1)
            {
                m.at(i).at(desnagr)=k;
                if (m.at(i+1).at(desnagr) != (m.at(i).at(desnagr))+1) return false;
                k++;
                i++;
            }
        }
    }
}
```

```
    desnagr--;
    //donjagr--;
    //gornjagr++;

    while(j>lijevagr-1)
    {
        m.at(donjagr).at(j)=k;
        if (m.at(donjagr).at(j-1) != (m.at(donjagr).at(j))+1) return false;
        k++;
        j--;
    }
    donjagr--;
    //desnagr--;
    //lijevagr++;

    while(i>gornjagr-1)
    {
        m.at(i).at(lijevagr)=k;
        if (m.at(i-1).at(lijevagr) != (m.at(i).at(lijevagr))+1) return false;
        k++;
        i--;
    }
    lijevagr++;
}
m.at(i).at(j)=k;
if (m.at(i).at(j) < m.at(0).at(0)) return false;
return true;
}
else if (smjer==false) //lijeva
{
    int gornjagr=0;
    int donjagr=brojredova-1;
    int desnagr=brojkolona-1;
    int lijevagr=0;
    int i=0;
    int j=0;
    while (gornjagr<=donjagr && lijevagr<=desnagr)
    {
        while(i<donjagr-1)
        {
            m.at(i).at(lijevagr)=k;
            if (m.at(i+1).at(lijevagr) != (m.at(i).at(lijevagr))+1) return false;
            k++;
            i++;
        }
        lijevagr++;

        while(j<desnagr-1)
        {
            m.at(donjagr).at(j)=k;
            if (m.at(donjagr).at(j+1) != (m.at(donjagr).at(j))+1) return false;
            k++;
            j++;
        }
        donjagr--;
    }
}
```



```
while(i>gornjagr-1)
{
    m.at(i).at(desnagr)=k;
    if (m.at(i-1).at(desnagr) != (m.at(i).at(desnagr))+1) return false;
    k++;
    i--;
}
desnagr--;

while(j>lijevagr-1)
{
    m.at(gornjagr).at(j)=k;
    if (m.at(gornjagr).at(j-1) != (m.at(gornjagr).at(j))+1) return false;
    k++;
    j--;
}
gornjagr++;
}
m.at(i).at(j)=k;
if (m.at(i).at(j) < m.at(0).at(0)) return false;
return true;
}
}
```

```
matrica KreirajSpiralnuMatricu (int brojredova, int brojkolona, int k, bool smjer)
{
    matrica prazna;
    if (brojredova<=0 || brojkolona<=0) return prazna; //0x0 matrica
    matrica m;
    m.resize(brojredova);
    for (int i=0;i<brojredova;i++) m.at(i).resize(brojkolona);

    if (smjer==true) //desna
    {
        int gornjagr=0;
        int donjagr=brojredova-1;
        int desnagr=brojkolona-1;
        int lijevagr=0;
        int i=0;
        int j=0;
        while (gornjagr<=donjagr && lijevagr<=desnagr)
        {
            while(j<desnagr)
            {
                m.at(gornjagr).at(j)=k;
                k++;
                j++;
            }
            gornjagr++;

            while(i<donjagr)
            {
                m.at(i).at(desnagr)=k;
                k++;
                i++;
            }
            donjagr--;
            desnagr--;
            lijevagr++;
        }
    }
}
```

```
}
desnagr--;
//donjagr--;
//gornjagr++;

while(j>lijevagr)
{
    m.at(donjagr).at(j)=k;
    k++;
    j--;
}
donjagr--;
//desnagr--;
//lijevagr++;

while(i>gornjagr)
{
    m.at(i).at(lijevagr)=k;
    k++;
    i--;
}
lijevagr++;
}
m.at(i).at(j)=k;
}
else if (smjer==false) //lijeva
{
    int gornjagr=0;
    int donjagr=brojredova-1;
    int desnagr=brojkolona-1;
    int lijevagr=0;
    int i=0;
    int j=0;
    while (gornjagr<=donjagr && lijevagr<=desnagr)
    {
        while(i<donjagr)
        {
            m.at(i).at(lijevagr)=k;
            k++;
            i++;
        }
        lijevagr++;

        while(j<desnagr)
        {
            m.at(donjagr).at(j)=k;
            k++;
            j++;
        }
        donjagr--;

        while(i>gornjagr)
        {
            m.at(i).at(desnagr)=k;
            k++;
            i--;
```

```
    }
    desnagr--;

    while(j>lijevagr)
    {
        m.at(gornjagr).at(j)=k;
        k++;
        j--;
    }
    gornjagr++;
}
m.at(i).at(j)=k;
}
return m;
}

int main ()
{
    std::cout<<"Unesite broj redova i kolona matrice: "<<std::endl;
    int brojredova;
    int brojkolona;
    std::cin>>brojredova>>brojkolona;
    std::cout<<"Unesite pocetnu vrijednost: "<<std::endl;
    int k;
    std::cin>>k;
    std::cout<<"Unesite L za lijevu, a D za desnu spiralnu matricu: "<<std::endl;
    char smjer;
    std::cin>>smjer;

    if (smjer=='D')
    {
        bool pravac=true;
        matrica m = KreirajSpiralnuMatricu(brojredova,brojkolona,k,pravac);
        std::cout<<"Kreirana desna spiralna matrica: "<<std::endl;
        for (int i=0;i<brojredova;i++)
        {
            for (int j=0;j<brojkolona;j++)
            {
                std::cout<<m.at(i).at(j)<<" ";
            }
            std::cout<<std::endl;
        }
    }
    else if (smjer=='L')
    {
        bool pravac=false;
        matrica m = KreirajSpiralnuMatricu(brojredova,brojkolona,k,pravac);
        std::cout<<"Kreirana lijeva spiralna matrica: "<<std::endl;
        for (int i=0;i<brojredova;i++)
        {
            for (int j=0;j<brojkolona;j++)
            {
                std::cout<<m.at(i).at(j)<<" ";
            }
            std::cout<<std::endl;
        }
    }
}
```



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

Zadatak . 4.cpp

```
//TP 2022/2023: Zadaća 1, Zadatak 4
#include <iostream>
#include <cmath>
#include <stdexcept>
#include <string>
#include <vector>
#include <cctype>

bool isti_su (std::string rijec, std::string zabranjena)
{
    for (int i=0;i<rijec.length();i++)
    {
        //if (std::tolower(rijec.at(i))==std::tolower(zabranjena.at(i))) isti = true;
        if (std::tolower(rijec.at(i))!=std::tolower(zabranjena.at(i))) return false;
    }
    return true;
}

std::string Cenzura(std::string recenica, std::vector<std::string>spisakrijeci)
{
    for (int i=0;i<spisakrijeci.size();i++)
    {
        for (int j=0;j<spisakrijeci.at(i).length();j++)
        {
            //if (((spisakrijeci.at(i).at(j)>='a' && spisakrijeci.at(i).at(j)<='z')
            //|| (spisakrijeci.at(i).at(j)>='A' && spisakrijeci.at(i).at(j)<='Z')
            //|| (spisakrijeci.at(i).at(j)>='0' && spisakrijeci.at(i).at(j)<='9'))))
            if (!isalnum(spisakrijeci.at(i).at(j)))throw std::logic_error(
"Neispravna rije?");
        }
    }

    std::string rijec;
    for (int i=0;i<recenica.length();i++) //DODATI EXCEPTIO ZA NEISPRAVNA RIJEC
    {
        if (isalnum(recenica.at(i)))
        {
            int pocetakrijeci = i;

//while (i<recenica.length() && recenica.at(i)!=' ' && recenica.at(i)!=',' && recenica
.at(i)!='!' && recenica.at(i)!='.' && recenica.at(i)!='?' && recenica.at(i)!='"')
            while (i<recenica.length() && isalnum(recenica.at(i)))
            {
                rijec=rijec+recenica.at(i);
                i++;
            }

            for (int j=0;j<spisakrijeci.size();j++)
            {
                if (rijec.length()==spisakrijeci.at(j).length())
//TOLOWER MOZE AKO JE at.i at.j ali onda kako porediti slovo i rijec cijelu
                {
                    if (isti_su(rijec, spisakrijeci.at(j)))
```

```
{
    for (int k=pocetakrijeci;k<i;k++)
    {
        recenica.at(k)='*';
    }
}

/*for (int x=pocetakrijeci;x<pocetakrijeci+rijec.length();x++)
{
    if (std::tolower(rijec.at(x)) == std::tolower(spisakrijeci.at(
j).at(x)))
        recenica.at(x)='*';
}*/
//for (int k=pocetakrijeci;k<i;k++)
//{
//    recenica.at(k)='*';
//}
}
rijec="";
}

}
return recenica;
}

int main ()
{
    try
    {
        std::cout<<"Unesite recenicu: "<<std::endl;
        std::string recenica;
        std::getline(std::cin, recenica);
        std::cout<<"Unesite zabranjene rijeci (. za kraj): "<<std::endl;
        std::vector<std::string> spisak;
        std::string zabranjene;
        int i=0;
        std::getline(std::cin, zabranjene);
        while (zabranjene.at(i)!='.')
        {
            std::string rijec;
            while (zabranjene.at(i)!=' ')
            {
                rijec=rijec+zabranjene.at(i);
                i++;
            }
            spisak.push_back(rijec);
            i++;
        }
        std::string cenzurisanarecenica = Cenzura(recenica,spisak);
        std::cout<<"Cenzurisana recenica: "<<cenzurisanarecenica;
    }

    catch(std::logic_error e)
    {

```

```
//std::cout<<e.what()<<std::endl;  
std::cout<<"GRESKA: Nelegalne zabranjene rijeci!";  
return 0;  
}  
  
return 0;  
}
```