

Datum 4.3.2015.

STATIČKI NIZOVI

O statičkim nizovima, jednodimenzionalnim i dvodimenzionalnim, je već bilo riječi na predmetu Programiranje I. Na predmetu Programiranje II bi ste se samo trebali podsjetiti već naučenog.

Niz je kontinuirani blok memorijskih lokacija koje se nazivaju jednim imenom i sve mogu pohraniti podatke istog tipa. Kako bi se jednoznačno odredilo o kojoj je memorijskoj lokaciji riječ, odnosno o kojem elementu niza, koriste se indeksi.

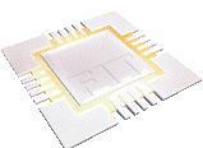
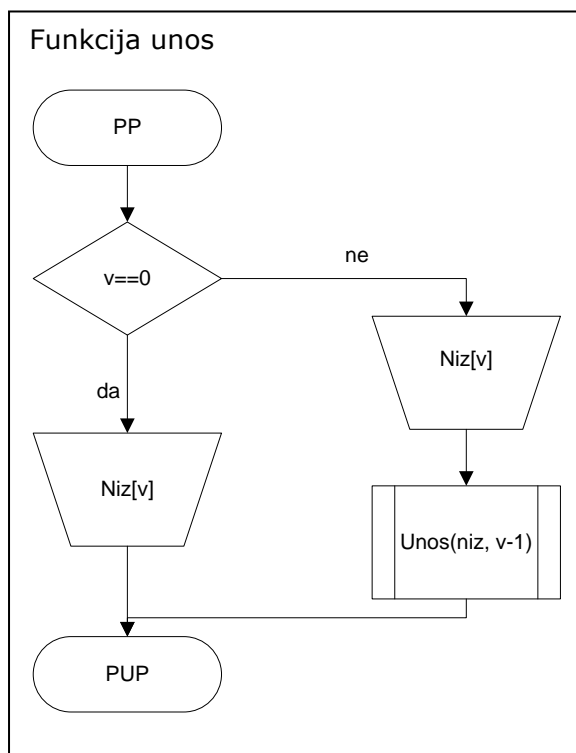
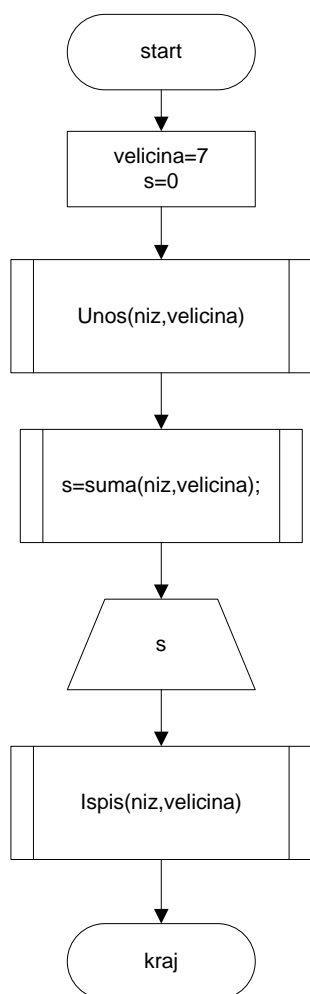
Veličina statičkog niza mora biti konstantna vrijednost. Prvi element niza ima indeks 0.

Kako vam je gradivo o statičkim nizovima već poznato, u zadacima koji slijede kombinirat ćemo ga s novonaučenim sadržajem - rekurzivnim funkcijama.

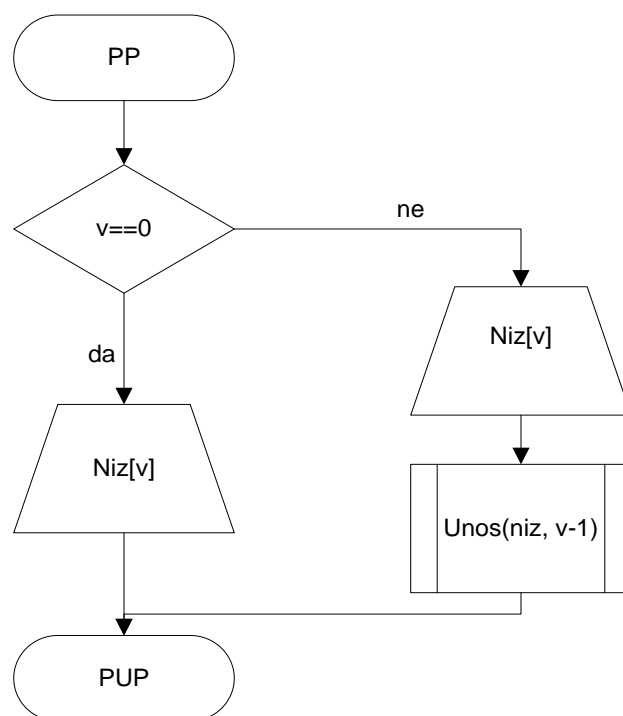
Zadatak

Napišite program koji će omogućiti kreiranje niza od 7 cijelih brojeva, te uz pomoć rekurzivnih funkcija omogućiti unos elemenata, ispis elemenata i izračunati sumu elemenata niza.

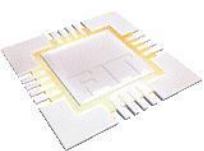
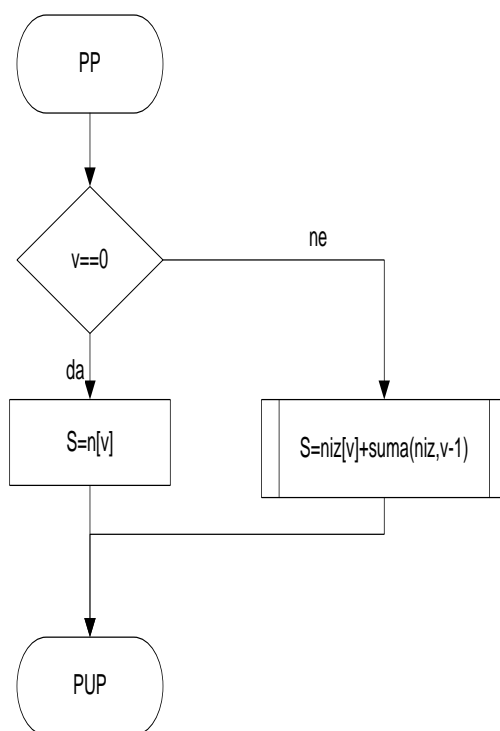
Funkcija main



Funkcija ispis



Funkcija suma



```
#include<iostream>
using namespace std;
void unos (int n[], int v)
{
    if(v==0)
        cin>>n[v];
    else
    {
        cin>>n[v];
        unos(n,v-1);
    }
}

void ispis (int n[], int v)
{
    if(v==0)
        cout<<n[v]<<"\t";
    else
    {
        cout<<n[v]<<"\t";
        ispis(n,v-1);
    }
}

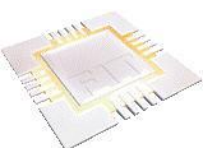
int suma(int n[], int v)
{
    if(v==0)
        return n[v];
    else
        return n[v]+suma(n, v-1);
}
```

```
int main()
{
    const int velicina=7;
    int niz[velicina];
    unos(niz, velicina-1); //velicina-1 jer ne
    postoji element s indeksom 7
    cout<<endl;
    cout<<"suma niza iznosi "<<suma(niz, velicina-
    1)<<endl; //velicina-1 jer ne postoji element s
    indeksom 7
    cout<<endl;
    ispis(niz, velicina-1); //velicina-1 jer ne
    postoji element s indeksom 7
    system("pause");
    return 0;
}
```



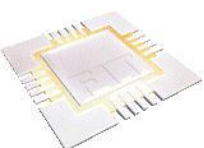
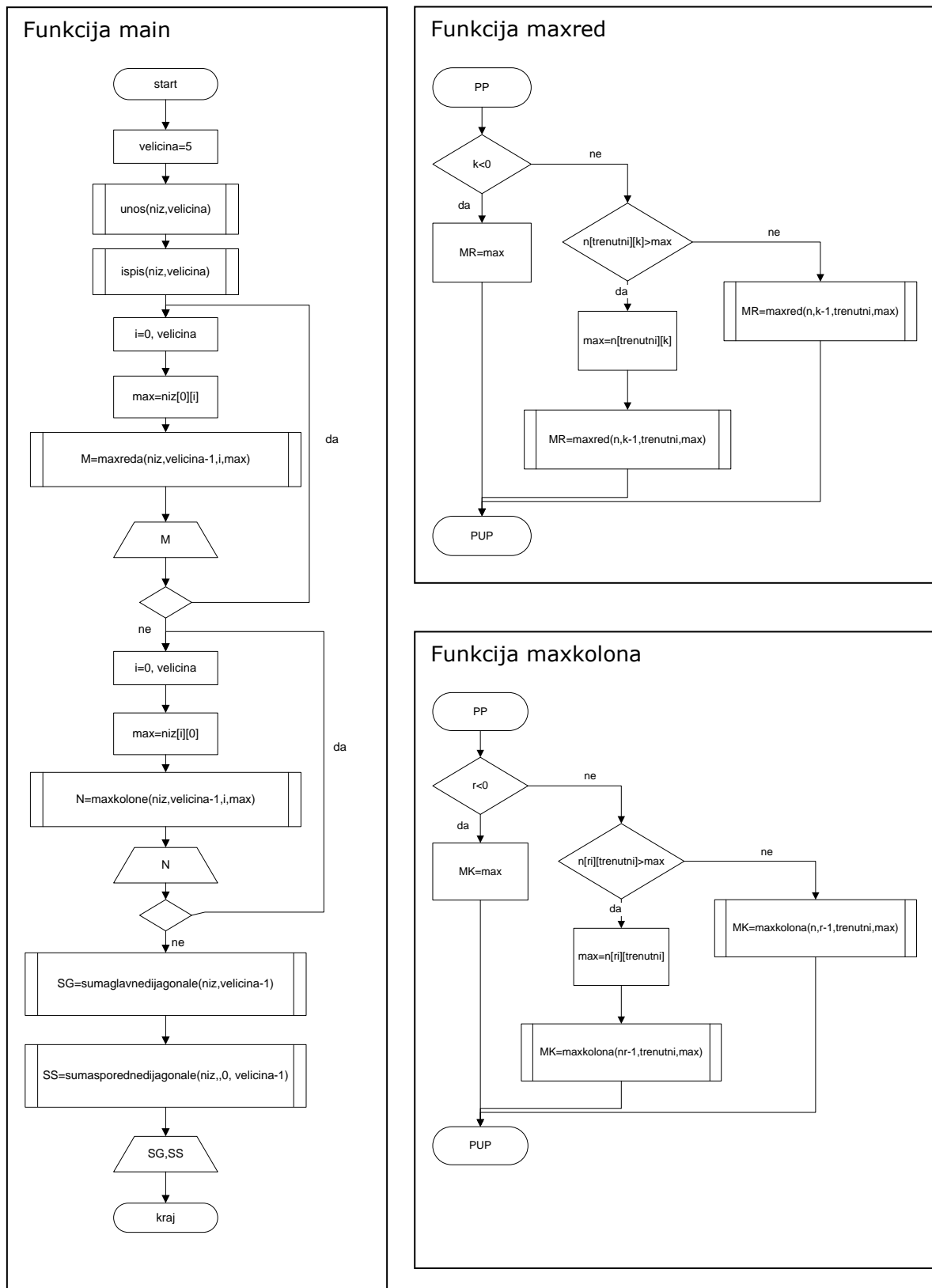
```
C:\Windows\system32\cmd.exe
Unsejte 7 cijelih brojeva
1
3
5
7
9
11
13

suma niza iznosi 49
1      3      5      7      9      11      13
```

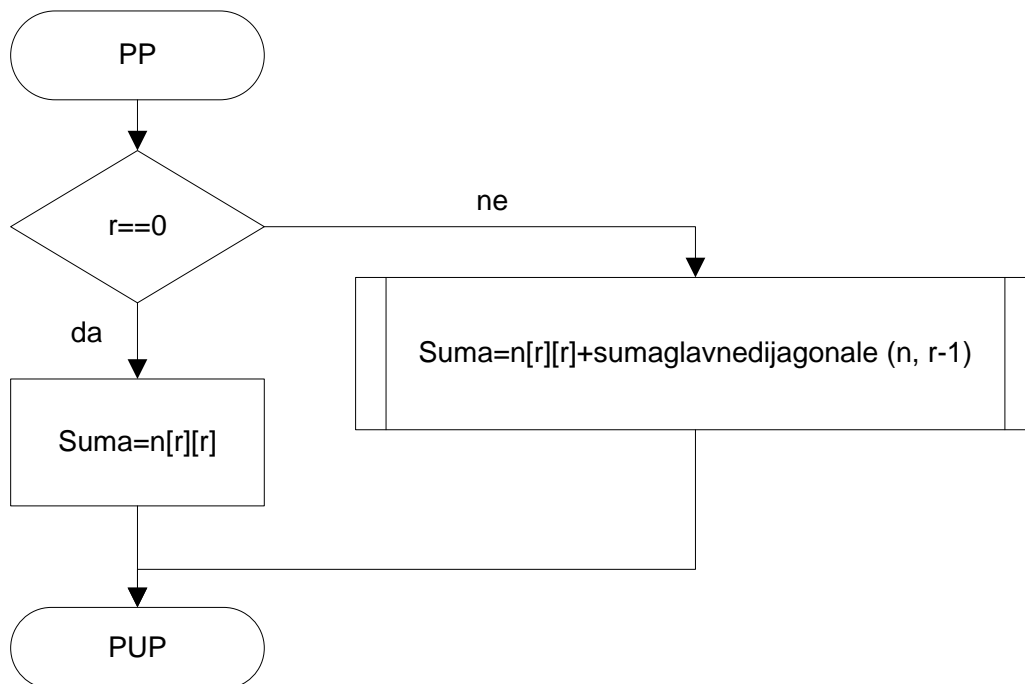


Zadatak

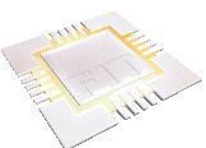
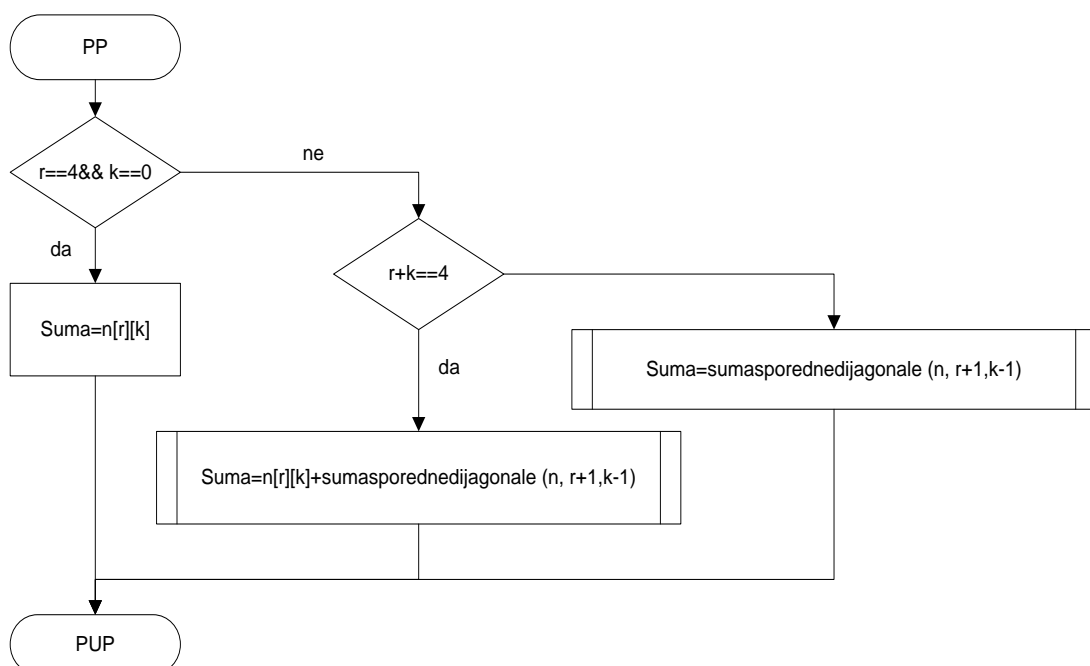
Napišite program koji će omogućiti kreiranje matrice 5x5. Uz pomoć rekurzivnih funkcija unijeti elemente u niz, ispisati elemente niza, pronaći najveću vrijednost u svakom od redaka, svakoj od kolona, te sumu elemenata na glavnoj i sporednoj dijagonali.



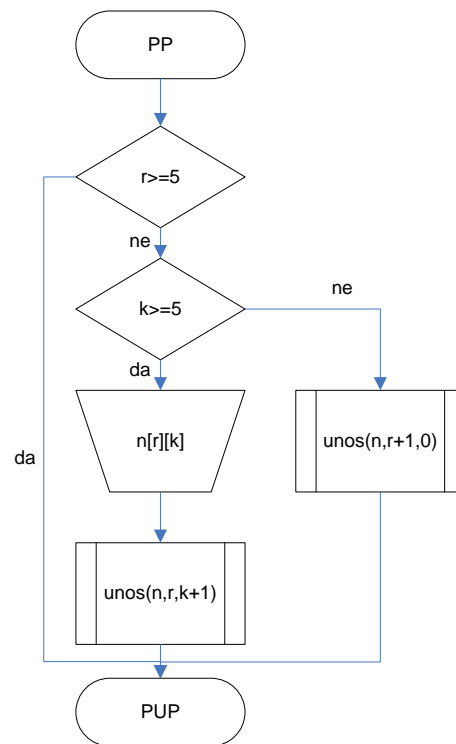
Funkcija sumaglavnedijagonale



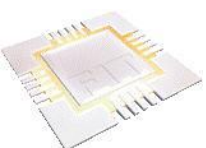
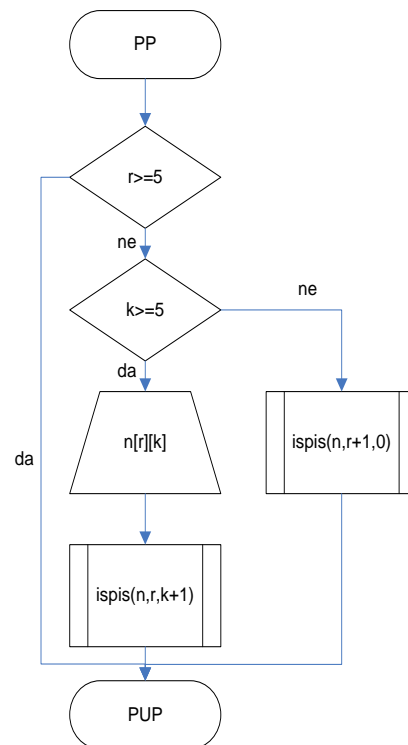
Funkcija sumasporednedijagonale



Funkcija unos



Funkcija ispis



```

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

void unos (int [][][5], int, int);
void ispis (int [][][5], int, int );
int maxred(int [][][5], int , int , int );
int maxkolona(int [][][5], int , int , int );
int sumaglavnedijagonale (int [][][5], int );
int sumasporednedijagonale (int [][][5], int , int );

int main()
{
    const int velicina=5;
    int niz[velicina][velicina], max;
    cout<<"Unesite elemente kvadratne matrice "<<velicina<<" x
    "<<velicina<<endl;
    unos(niz,0,0);
    cout<<endl;
    cout<<"u matricu su unesene vrijednosti"<<endl;
    ispis(niz, 0,0);
    cout<<endl;
        for(int i=0; i<velicina; i++)
        {
            max=niz[i][0];
            cout<<"najveca vrijednost u retku "<<i+1<<" iznosi
    "<<setw(3)<<maxred(niz, velicina-1, i,max)<<endl;
        }
    cout<<endl;
        for(int i=0; i<velicina; i++)
        {
            max=niz[0][i];
            cout<<"najveca vrijednost u koloni "<<i+1<<" iznosi
    "<<setw(3)<<maxkolona(niz, velicina-1, i,max)<<endl;
        }
    cout<<endl;
    cout<<"Suma elemenata na glavnoj dijagonali iznosi
    "<<sumaglavnedijagonale(niz, velicina-1)<<endl<<endl;
    cout<<"Suma elemenata na sporednoj dijagonali iznosi
    "<<sumasporednedijagonale(niz, 0,4);

    system("pause>0");
    return 0;

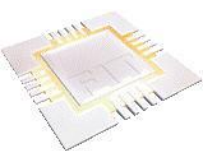
}

void unos(int n[][][5], int r, int k)
{
    if (r>=5)
        return;
    if(k<5)
    {
        cout<<"Unesite element ["<<r<<"] ["<<k<<"]:";
        cin>>n[r][k];
        unos(n,r,k+1);}
    else
        unos(n,r+1,0);

}

void ispis(int n[][][5], int r, int k)
{
    if (r>=5)
        return;

```



```
if(k<5)
{
    cout<<setw(3)<<n[r][k];
    ispis(n,r,k+1);
}
else
{
    cout<<endl;
    ispis(n,r+1,0);
}

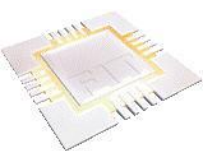
int maxred(int n[][5], int k, int trenutni, int max)
{
    if(k<0)
        return max;
    else
        if(n[trenutni][k]>max)
        {
            max=n[trenutni][k];
            return maxred(n,k-1,trenutni, max);
        }
        else
            return maxred(n, k-1, trenutni,max);
}

int maxkolona(int n[][5], int r, int trenutna, int max)
{
    if(r<0)
        return max;

    if(n[r][trenutna]>max)
    {
        max=n[r][trenutna];
        return maxkolona(n, r-1, trenutna,max);
    }
    else
        return maxkolona(n, r-1, trenutna,max);
}

int sumaglavnedijagonale (int n[][5], int r)
{
    if (r==0)
        return n[r][r];
    else
        return n[r][r] + sumaglavnedijagonale(n, r-1);
}

int sumasporednedijagonale (int n[][5], int r, int k)
{
    if ((r==4)&&(k==0))
        return n[r][k];
    if (r+k==4)
        return n[r][k]+ sumasporednedijagonale(n, r+1,k-1);
    else
        return sumasporednedijagonale(n, r+1,k-1);
}
```




```
C:\Windows\system32\cmd.exe

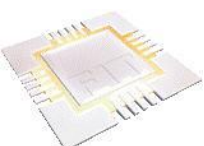
Unesite elemente kvadratne matrice 5 x 5
Unesite element [0] [0]:1
Unesite element [0] [1]:2
Unesite element [0] [2]:3
Unesite element [0] [3]:4
Unesite element [0] [4]:5
Unesite element [1] [0]:6
Unesite element [1] [1]:7
Unesite element [1] [2]:8
Unesite element [1] [3]:9
Unesite element [1] [4]:10
Unesite element [2] [0]:11
Unesite element [2] [1]:12
Unesite element [2] [2]:13
Unesite element [2] [3]:14
Unesite element [2] [4]:15
Unesite element [3] [0]:16
Unesite element [3] [1]:17
Unesite element [3] [2]:18
Unesite element [3] [3]:19
Unesite element [3] [4]:20
Unesite element [4] [0]:21
Unesite element [4] [1]:22
Unesite element [4] [2]:23
Unesite element [4] [3]:24
Unesite element [4] [4]:25

u matricu su unesene vrijednosti
 1  2  3  4  5
 6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

najveca vrijednost u retku 1 iznosi 5
najveca vrijednost u retku 2 iznosi 10
najveca vrijednost u retku 3 iznosi 15
najveca vrijednost u retku 4 iznosi 20
najveca vrijednost u retku 5 iznosi 25

najveca vrijednost u koloni 1 iznosi 21
najveca vrijednost u koloni 2 iznosi 22
najveca vrijednost u koloni 3 iznosi 23
najveca vrijednost u koloni 4 iznosi 24
najveca vrijednost u koloni 5 iznosi 25

Suma elemenata na glavnoj dijagonali iznosi 65
Suma elemenata na sporednoj dijagonali iznosi 65
```



Napomena: Predstavljena rješenja nisu jedina moguća rekurzivna rješenja (ima ih još) niti su osobito efikasna - služe samo za uvježbavanje rekurzije.

Na primjer, definicije nekih od funkcija iz prethodnog zadatka su mogle izgledati i ovako:

```
const int red = 5, kolona = 5; //globalne konstante

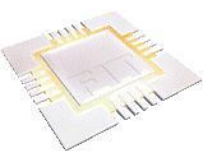
void unos(int matrica[][kolona], int r, int k)
{
    if(r<0)
        return;
    if(k==0)
        unos(matrica, r-1, kolona-1);
    else
        unos(matrica, r, k-1);
    cout <<"Unesi element na mjestu ["<<r<<"]["<<k<<"]: ";
    cin >> matrica[r][k];
}

void ispis(int matrica[][kolona], int r, int k)
{
    if(r<0)
        return;
    if(k==0)
    {
        ispis(matrica, r-1, kolona-1);
        cout<<endl;
    }
    else
        ispis(matrica, r, k-1);

    cout<< matrica[r][k]<<"\t";
}

int najveca_u_redu(int matrica[][kolona], int i, int j,int najveci)
{
    if(j<0)
        return najveci;
    if(matrica[i][j] > najveci)
        najveci = matrica[i][j];
    return najveca_u_redu(matrica, i, j-1, najveci);
}

int najveca_u_koloni(int matrica[][kolona], int i, int j, int najveci)
{
    if(j<0)
        return najveci;
    if(matrica[i][j] > najveci)
        najveci = matrica[i][j];
    return najveca_u_koloni(matrica, i-1, j, najveci);
}
```



A pozivi ovih funkcija bi izgledali ovako:

```
unos(matrica, red-1, kolona-1);

ispis(matrica, red-1, kolona-1);

for(int i=0; i<red; i++)
{
    najveci = matrica[i][0];
    cout <<"Najveca vrijednost u redku br."<<i<<" ==> "<<najveca_u_redu(matrica, i,kolona-1,najveci)<<endl;
}

for(int j=0; j<kolona; j++)
{
    najveci = matrica[0][j];
    cout <<"Najveca vrijednost u koloni br."<<j<<" ==> "<<najveca_u_koloni(matrica, red-1, j, najveci)<<endl;
}
```

