

01001111 01110011 01101110 01101111
01110110 01101001 00100000 01110000
01110010 01101111 01100111 01110010
01100001 01101101 01101001 01110010
01100001 01101110 01101010 01100001

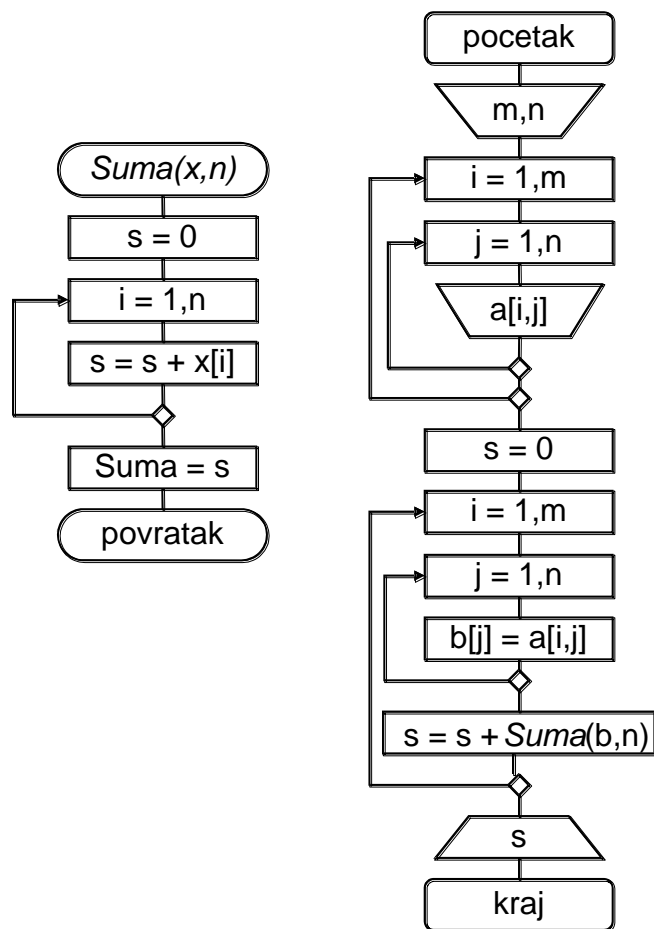


OSNOVI PROGRAMIRANJA

VEŽBE 10

Marina Svičević, Đorđe Nedić, Mladen Marić,
Danica Prodanović, Jovan Janićijević

- Definisati funkciju **Suma** koja za niz celih brojeva dužine n izračunava njegovu sumu. Napisati program koji, koristeći funkciju **Suma**, određuje sumu svih elemenata matrice celih brojeva dimenzije $m \times n$.



```

#include <stdio.h>
int Suma(int a[], int n)
{
    int i, s=0;
    for(i=0; i<n; i++)
        s +=a[i];
    return s;
}

main()
{
    int a[20][20], b[20];
    int i, j, n, m, s;
    scanf("%d%d", &m, &n);
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            scanf("%d", &a[i][j]);

    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++) b[j]=a[i][j];

        s +=Suma(b, n);
    }

    printf("%d ", s);
}
  
```

- Definirati funkciju **Max** koja za niz od n celih brojeva određuje maksimum niza. Definirati funkciju **MaxPozicija** koja za niz od n celih brojeva vraća poziciju maksimuma niza. Napisati program koji za matricu celih brojeva dimenzije $m \times n$ formira dva niza, gde prvi sadrži pronađene maksimalne elemente za odgovarajuću vrstu, a drugi poziciju maksimalnog elementa u vrsti. Koristeći dobijene nizove i proceduru **MaxPozicija** odrediti vrednost i poziciju najvećeg elementa u matrici.

```
#include <stdio.h>
```

```
int Max(int niz[], int n)
```

```
{
    int i;
    int maksimum = niz[0];

    for(i = 1; i < n; i++)
    {
        if(niz[i] > maksimum)
            maksimum = niz[i];
    }

    return maksimum;
}
```

```
int MaxPozicija(int niz[], int n)
```

```
{
    int i;
    int maksimum = niz[0];
    int pozicijaMaksimuma = 0;

    for(i = 1; i < n; i++)
    {
        if(niz[i] > maksimum)
        {
            pozicijaMaksimuma = i;
            maksimum = niz[i];
        }
    }

    return pozicijaMaksimuma;
}
```

```
main()
```

```
{
    int i, j, m, n, p;
    int matrica[50][50];
    int niz[50];
    int maksimumiPoVrstama[50];
    int pozicijeMaksimuma[50];
    int t[50];

    scanf("%d%d", &m, &n);

    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            scanf("%d", &matrica[i][j]);

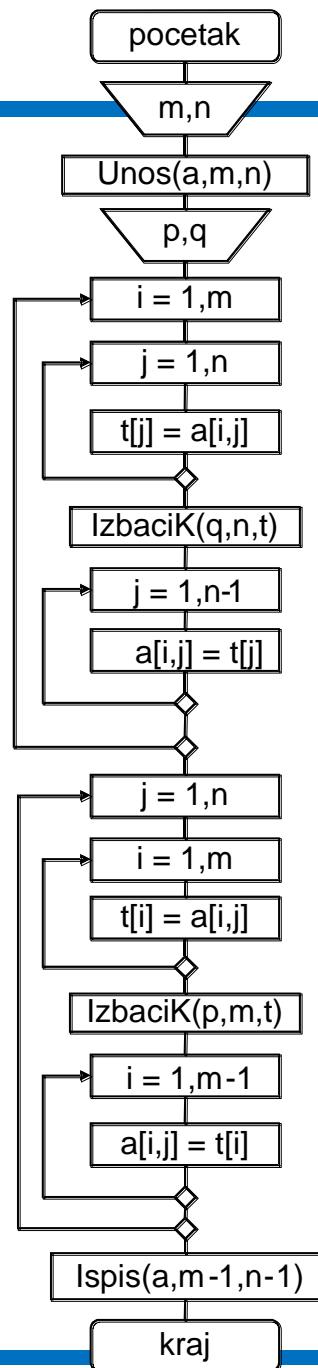
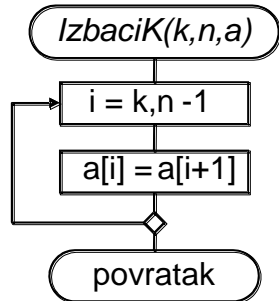
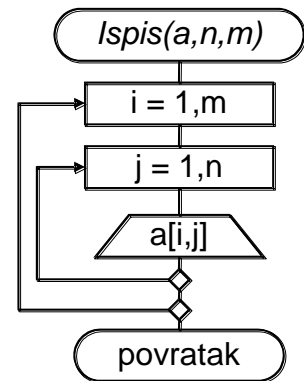
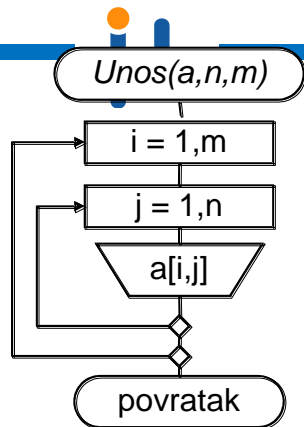
    for(i = 0; i < m; i++)
    {
        for(j = 0; j < n; j++)
            t[j] = matrica[i][j];

        maksimumiPoVrstama[i] = Max(t, n);
        pozicijeMaksimuma[i] = MaxPozicija(t, n);
    }

    p = MaxPozicija(maksimumiPoVrstama, m);

    printf("[%d, %d] = %d", p, pozicijeMaksimuma[p],
            maksimumiPoVrstama[p]);
}
```

- Definisati funkciju **IzbaciK** koja iz niza celih brojeva, dužine n , izbacuje k -ti ($1 \leq k \leq n$) element. Napisati program koji iz matrice celih brojeva dimenzije $m \times n$ izbacuje p -tu vrstu ($1 \leq p \leq m$) i q -tu kolonu ($1 \leq q \leq n$).



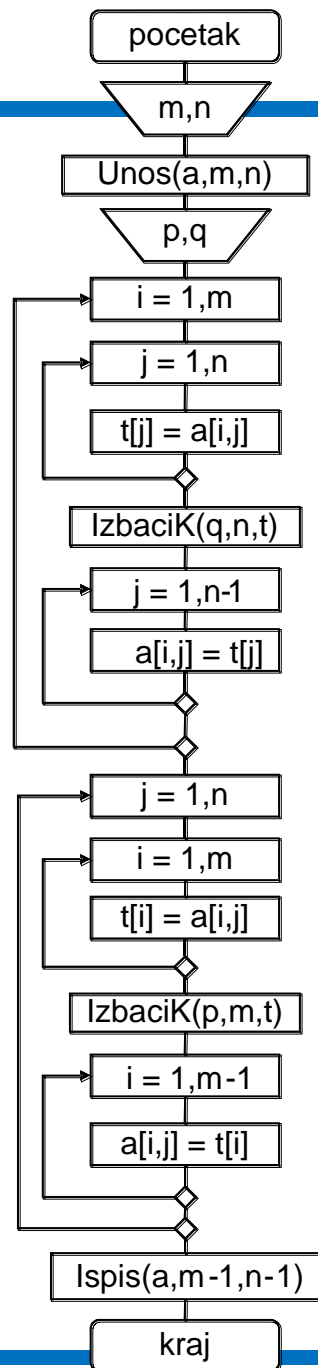
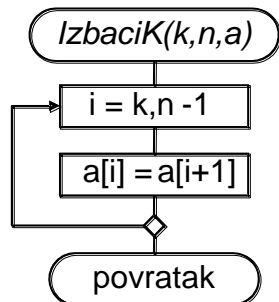
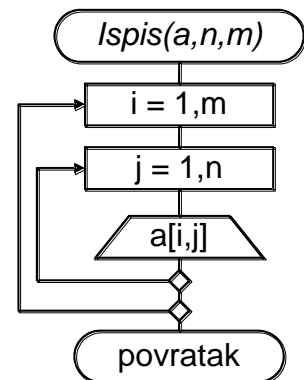
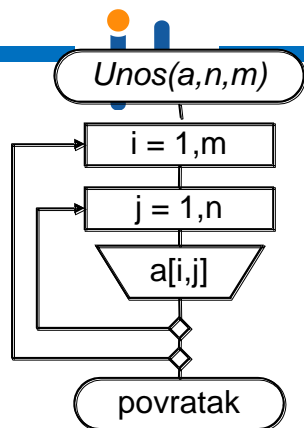
```

#include <stdio.h>

void Unos(int a[50][50],int m,int n)
{
    int i,j;
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
}

void Ispis(int a[50][50],int m,int n)
{
    int i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            printf("%5d",a[i][j]);
        printf("\n");
    }
}

void IzbaciK(int a[], int n, int k)
{
    int i;
    for(i=k-1;i<n-1;i++) a[i]=a[i+1];
}
  
```



main()

{

```
int a[50][50], b[50][50], t[50];
```

```
int i, j, n, m, p, q;
```

```
scanf("%d%d", &m, &n);
```

```
Unos(a, m, n);
```

```
scanf("%d%d", &p, &q);
```

```
for(i=0; i<m; i++)
```

```
{
```

```
    for(j=0; j<n; j++)
```

```
        t[j]=a[i][j];
```

```
    IzbaciK(t, n, q);
```

```
    for(j=0; j<n-1; j++)
```

```
        a[i][j]=t[j];
```

```
}
```

```
for(j=0; j<n; j++)
```

```
{
```

```
    for(i=0; i<m; i++)
```

```
        t[i]=a[i][j];
```

```
    IzbaciK(t, m, p);
```

```
    for(i=0; i<m-1; i++)
```

```
        a[i][j]=t[i];
```

```
}
```

```
Ispis(a, m-1, n-1);
```

```
}
```

```
}
```

```
}
```


FUNKCIJE ZA RAD SA STRINGOVIMA

```
#include <stdio.h>
#include <string.h>
/* Izracunava duzinu stringa */
int duzina_stringa(char s[])
{
    int i;
    for (i = 0; s[i]!='\0'; i++) ;
    return i;
}

/* Kopira string src u string dest.
Pretpostavlja da u dest ima dovoljno prostora.
*/
void kopiraj_string(char dest[], char src[])
{
    int i;
    /* Kopira karakter po karakter, sve dok
nije iskopiran karakter '\0' */
    for (i = 0; src[i]!='\0'; i++)
        dest[i]=src[i];

    dest[i]='\0';
}

void kopiraj_deo(char dest[], char src[], int
poz, int br)
{
    int i;
    for (i = 0; i<br; i++)
        dest[i]=src[poz+i];
    dest[i]='\0';
}
```

```
/* Nadovezuje string t na kraj stringa s.
Pretpostavlja da u s ima dovoljno prostora. */
void nadovezi_stringove(char s[], char t[])
{
    int i, j;
    /* Pronalazimo kraj stringa s */
    for (i = 0; s[i]!='\0'; i++);
    /* Vrsi se kopiranje, slicno funkciji
kopiraj_string */
    for (j = 0; s[i] = t[j]; j++, i++);
}

/* Pronalazi prvu poziciju karaktera c u stringu s,
odnosno -1
ukoliko s ne sadrzi c */
int string_char(char s[], char c)
{
    int i;
    for (i = 0; s[i]!='\0'; i++)
        if (s[i] == c)
            return i;
    /* nikako
else
return -1;
*/
    /* Nije nadjeno */
    return -1;
}
```

FUNKCIJE ZA RAD SA STRINGOVIMA

```
/* Proverava da li string str sadrzi string
sub.
Vraca poziciju na kojoj sub pocinje, odnosno -
1 ukoliko ga nema
*/
int string_string(char str[], char sub[])
{
    int i, j;
    /* Proveravamo da li sub pocinje na
    svakoj poziciji i */
    for (i = 0; str[i]; i++)
        /* Poredimo sub sa str pocevsi od
        poziciji i
        sve dok ne naidjemo na razliku */
        for (j = 0; str[i+j] == sub[j];
        j++)
            /* Nismo naisli na razliku a
            ispitali smo
            sve karaktere niske sub */
            if (sub[j+1] == '\0')
                return i;
    /* Nije nadjeno */
    return -1;
}

void delete_deo(char str[], int poz, int br)
{
    int i;
    for (i = 0; str[i+poz+br] != '\0'; i++)
        str[poz+i] = str[poz+br+i];

    str[poz+i] = '\0';
}
```

```
main()
{
    char s[]="Kragujevac";
    char s1[20],s2[20],s3[20];
    char t1[]="Dobar ", t2[]="dan";
    char c='a';
    //duzina
    printf("Nasa funkcija: %d\n",duzina_stringa(s));
    printf("Standardna funkcija: %d\n",strlen(s));
    //kopiranje
    kopiraj_string(s1,s);
    printf("Nasa funkcija: %s\n",s1);
    strcpy(s2,s);
    printf("Standardna funkcija: %s\n",s2);
    kopiraj_deo(s3,s,3,2);
    printf("Nasa funkcija: %s\n",s3);

    nadovezi_stringove(t1,t2);
    //strcat(t1,t2);
    printf("Nasa funkcija: %s\n",t1);
    /*Poredjenje dva stringa*/
    printf("%d\n", strcmp("abc","abc"));
    printf("%d\n", strcmp("abc","bcd"));
    printf("%d\n", strcmp("bcd","abc"));

    printf("Prvu poziciju karaktera:
    %d\n",string_char(s,c));
    printf("Poziciju podstringa:
    %d\n",string_string("Kragujevac","ra"));

    delete_deo(s,2,6);
}
```



- Napisati program koji uneti string u formatu

Prezime*Broj*Ime
prevodi u format
Ime Prezime Broj

```
main()
{
    char s[50],p[20],b[20]; int n;
    scanf("%s",s);

    n = string_char(s,'*');
    kopiraj_deo(p,s,0,n);
    delete_deo(s,0,n+1);

    n = string_char(s,'*');
    kopiraj_deo(b,s,0,n);
    delete_deo(s,0,n+1);

    strcat(s," ");
    strcat(s,p);
    strcat(s," ");
    strcat(s,b);

    printf("%s",s);

}
```

- Napisati f-ju koji za unetu reč određuje da li je palindrom ili ne. Za niz unetih reči štampati palindrome među njima.



```
#include <stdio.h>
#include <string.h>
int palindrom(char rec[])
{
    int i,j,duzinaReci = strlen(rec);
    int jestePalindrom=0;
    i = 0;
    j = duzinaReci-1;
    while((rec[i]==rec[j]) && (i<j))
    {
        i++;
        j--;
    }
    if (i<j) return 0;
    else return 1;
}
main()
{
    char nizReci[100][100];
    int n,i;
    printf("Koliko ima reci? \n");
    scanf("%d",&n);
    for (i=0;i<n;i++)
        scanf("%s",&nizReci[i]);
    for (i=0;i<n;i++)
        if (palindrom(nizReci[i])) printf("%s\n",nizReci[i]);
}
```