



STRUKTURE PODATAKA I ALGORITMI 1

VEŽBE 8

Tijana Ristović
Aleksa Cerovina
Filip Radovanović
Đorđe Nedić



- Napisati program koji izračunava sumu kvadrata, kubova i dvostrukih vrednosti od 1 do n.

```
#include <stdio.h>
int kvadrat(int n) {
    return n*n;
}
int kub(int n){
    return n*n*n;
}
int parni_broj(int n){
    return 2*n;
}
int sumiraj(int (*f) (int), int n) {
    int i, suma=0;
    for (i=1; i<=n; i++)
        suma += (*f) (i);
    return suma;
}
```

```
main()
{
    printf("Suma kvadrata brojeva od jedan do 3 je %d\n",
           sumiraj(&kvadrat,3));
    printf("Suma kubova brojeva od jedan do 3 je %d\n",
           sumiraj(&kub,3));
    printf("Suma prvih pet parnih brojeva je %d\n",
           sumiraj(&parni_broj,5));
}
```



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char ime[100];
    char *opis;
    strcpy(ime, "Marko");
    opis = malloc( 200 * sizeof(char) );
    if( opis == NULL )
    {
        fprintf(stderr, "Error \n");
    }
    else
    {
        strcpy( opis, "Marko, student IMI");
    }
    printf("Ime = %s\n", ime );
    printf("Opis: %s\n", opis );
}
```



```
main()
{
    int a[3][3] = {{0, 1, 2}, {10, 11, 12}, {20, 21, 22}};
    int i, j;
    /* Alternativni unos elemenata matrice
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
            {
                printf("a[%d][%d] = ", i, j);
                scanf("%d", &a[i][j]);
            }

    */

    a[1][1] = a[0][0] + a[2][2];
    /* a[1][1] = 0 + 22 = 22 */
}
```



```
printf("%d\n", a[1][1]); /* 22 */  
/* Stampanje elemenata matrice*/  
for(i=0; i<3; i++)  
{  
    for(j=0; j<3; j++)  
        printf("%d\t", a[i][j]);  
    printf("\n");  
}
```

- Nama je potrebno da imamo veću fleksibilnost, tj da se dimenzije matrice mogu uneti kao parametri našeg programa. Zbog toga je neophodno koristiti dinamičku alokaciju memorije.



IMPLEMENTACIJA MATRICE PREKO NIZA.

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

/* Makro pristupa članu na poziciji i, j matrice koja ima
m vrsta i n kolona */
#define a(i,j) a[(i)*n+(j)]

main()
{
    int m, n; /* Dimenzije matrice */
    int *a; /* Matrica */
    int i,j;
    int s=0; /* Suma elemenata matrice */
    /* Unos i alokacija */
    scanf("%d",&m);
    scanf("%d",&n);
    a=malloc(m*n*sizeof(int));
    if (a == NULL) {
        printf("Greska prilikom alokacije memorije!\n");
        exit(1);
    }
}
```



IMPLEMENTACIJA MATRICE PREKO NIZA.

```
for (i=0; i<m; i++)
    for (j=0; j<n; j++)
    {
        printf("Unesi element na poziciji (%d,%d) : ",i,j);
        scanf("%d",&a(i,j));
    }

/* Racunamo sumu elemenata matrice */
for (i=0; i<m; i++)
    for (j=0; j<n; j++)
        s+=a(i,j);

/* Ispis unete matrice */
printf("Uneli ste matricu : \n");
for (i=0; i<m; i++)
    { for (j=0; j<n; j++)
        printf("%d ",a(i,j));
        printf("\n");
    }

printf("Suma elemenata matrice je %d\n", s);

/* Oslobadjamo memoriju */
free(a);
}
```