



STRUKTURE PODATAKA I ALGORITMI 1

VEŽBE 7

Nikola Bačanin

Nikola Rnjak



- 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.

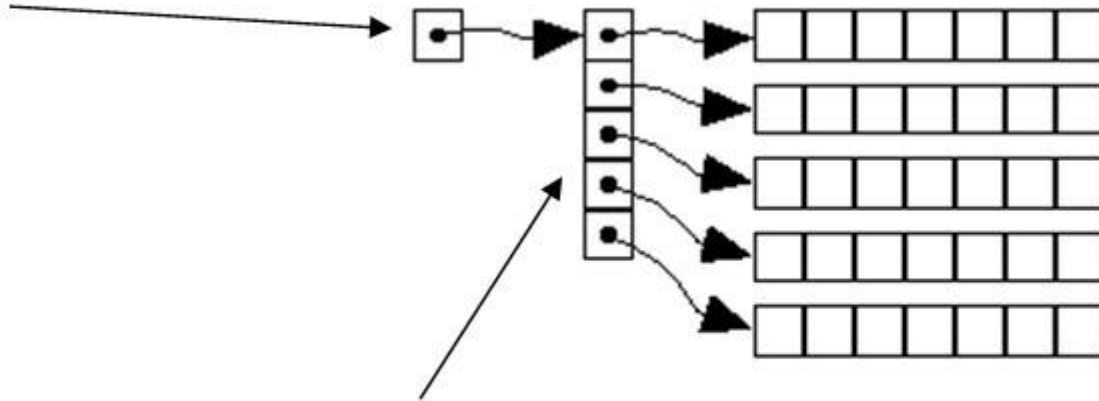
```
#include <stdlib.h>
#include <stdio.h>
#define a(i, j) a[(i)*n + (j)]
int main()
{
    int m, n; /* Dimenzije matrice */
    int *a;    /* Matrica */
    int i, j;
    int s = 0; /* Suma elemenata matrice */
    /* Unos i alokacija */
    scanf("%d", &m); scanf("%d", &n);
    a = malloc(m * n * sizeof(int));
    if (a == NULL) {
        printf("Greska prilikom alokacije memorije!\n");
        return 1;
    }
}
```



```
for (i = 0; i < m; i++)
    for (j = 0; j < n; j++) {
        printf("Unesi a(%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 */
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);
}
```



Показивач на
низ показивача



Елементи низа, показивачи
који ће у себи чувати адресе
динамички алоцираних
блокова меморије.

- Име niza je pokazivač na nulti element niza.
- Za svaki pokazivač u nizu pokazivača se radi dinamičko zauzimanje nekog broja elementa



- Primer alokacije 2Dimenzinalnog niza - niza pokazivača
- Sve razdvojeno po funkcijama
- Napraviti funkciju koja za datu matricu formira niz koji predstavlja broj pozitivnih elemenata po vrstama

```
#include <stdlib.h>
#include <stdio.h>
/* Dinamicka matrica je odredjena adresom
pocetka niza pokazivaca i dimenzijama tj.
int** a;
int m,n;
*/
/* Alokacija kvadratne matrice nxm */
int** alociraj(int m, int n)
{
    int** mat;
    int i;
    mat=malloc(m*sizeof(int*));
```



```
if (mat == NULL)
{
    printf("Greska prilikom alokacije memorije!\n");
    exit(1);
}
for (i=0; i<n; i++)
{
    mat[i]=malloc(n*sizeof(int));
    if (mat[i] == NULL)
    {
        int k;
        printf("Greska prilikom alokacije
memorije!\n");
        for(k=0;k<i;k++)
            free(mat[k]);
        exit(1);
    }
}
return mat;
}
```



```
/* Ispis matrice */
void ispisi_matricu(int** mat, int n, int m)
{
    int i, j;
    for (i=0; i<m; i++)
    {
        for (j=0; j<n; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }
}

/* Dealokacija matrice dimenzije nxn */
void obrisi(int** mat, int n)
{
    int i;
    for (i=0; i<n; i++)
        free(mat[i]);
    free(mat);
}
```



```
void formiraj(int** mat, int *niz, int n, int m)
{
    int i, j, br;
    for (i=0; i<m; i++)
    {
        br=0;
        for (j=0; j<n; j++)
            if (mat[i][j]>0) br++;
        niz[i]=br;
    }
}

/* Ispis niza */
void ispisi_niz(int* niz, int n)
{
    int i;
    for (i=0; i<n; i++)
        printf("%d ", niz[i]);
    printf("\n");
}
```



```
int main() {
    int **mat, *niz;
    int n, m;
    int i,j;
    printf("Unesi dimenziju matrice : ");
    scanf("%d",&m);
    scanf("%d",&n);
    mat=alociraj(m,n);
    niz=malloc(m*sizeof(int));
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
            scanf("%d",&mat[i][j]);
    printf("Uneli ste matricu : \n");
    ispisi_matricu(mat,n,m);
    formiraj(mat,niz,n,m);
    ispisi_niz(niz,m);
    obrisi(mat,n);
    return 0;
}
```



- Napisati program koji izracunava 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;
}
```



```
int 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));

    return 0;
}
```



```
#include <stdio.h>
#include <stdlib.h>
int main ( int argc, char *argv[] )
{
    printf( "BROJ ARGUMENATA=%d\n\n", argc );

    for( int i=0; i<argc; i++ )
        printf("%d.ARGUMENT=%s\n", i, argv[i] );
}
```

./izv jedan dva tri cetiri pet

BROJ ARGUMENATA=6

1.ARGUMENT=./izv
2.ARGUMENT=jedan
3.ARGUMENT=dva
4.ARGUMENT=tri
5.ARGUMENT=cetiri
5.ARGUMENT=pet