

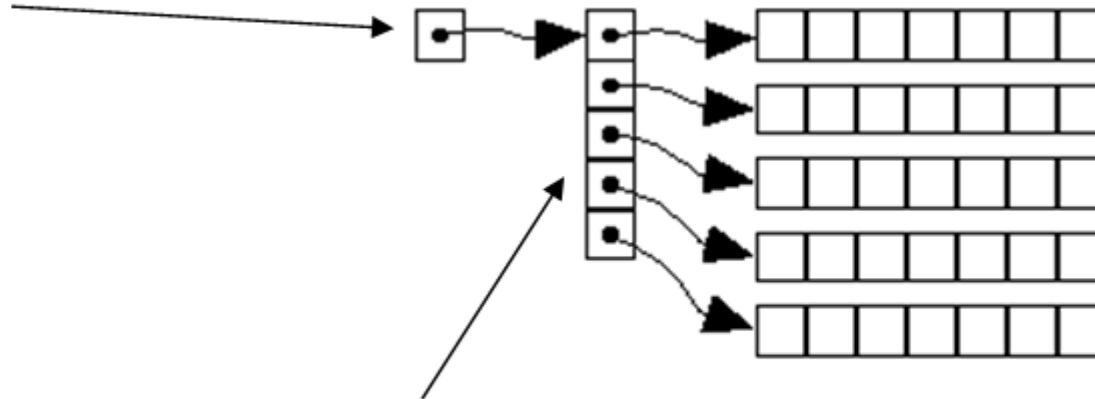


# STRUKTURE PODATAKA I ALGORITMI 1

## VEŽBE 9

Tijana Ristović  
Aleksa Cerovina  
Filip Radovanović  
Đorđe Nedić

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
- Svaki pokazivač će dobiti adresu bloka memorije koji će biti dinamički zauzet
- Napraviti na taj način jediničnu matricu date dimenzije n

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

// BROJ REDOVA (VRSTA) JE ISTI KAO I BROJ KOLONA!!!
main() {
    int** matrica,n,i,j; // Deklaracija pokazivaca na niz
    pokazivaca
    // ZAUZIMANJE NIZA POKAZIVACA
        scanf("%d",&n);
        matrica = malloc(n * sizeof(int *));
        if(matrica == NULL) {
            printf("Nema dovoljno memorije!!!\n");
            exit(EXIT_FAILURE);
        }
}
```



```
for(i = 0; i < n; i++){
    // DODELA BLOKA MEMORIJE SVAKOM POKAZIVACU
    matrica[i] = malloc(n * sizeof(int));
    if(matrica [i] == NULL){
        printf("Nema dovoljno memorije!!!\n");
        exit(EXIT_FAILURE);
    }
}

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

for (i=0; i<n; i++)
    matrica[i][i]=1;
for(i=0; i<n; i++)
{
    for(j=0; j<n; j++)
        printf("%d\t", matrica[i][j]);
    printf("\n");
}
}
```

- 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 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<m; 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");
}
```



```
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,m);
}
```