

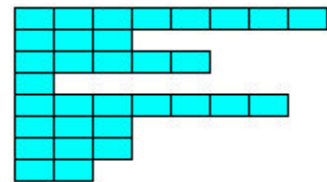
Strukture podataka i algoritmi 1
Septembar, 2010

1. U realnim problemima je često pogodno koristiti matrice čiji redovi nisu jednakih dužina (videti sliku). Takve matrice se nazivaju *nazubljenim* ili *testerastim* matricama. Napraviti program za rad sa ovakvim matricama na sledeći način:
- Napisati funkciju *Ucitaj* koja iz tekstualne datoteke učitava nazubljenu matricu celih brojeva. U prvom redu datoteke zapisan je broj redova matrice, a zatim su jedan ispod drugog navedeni red po red matrice. Svaki red matrice je u datoteci predstavljen sa dve linije teksta. U prvoj liniji je zapisan broj elemenata u tom redu, dok su u drugoj liniji navedeni svi elementi tog reda matrice.
(10 poena)
 - Napisati funkciju *ObrisiElement* koja uklanja iz matrice element u zadatom redu i koloni.
(3 poena)
 - Napisati funkciju *ObrisiRed* koja uklanja zadati red iz matrice.
(4 poena)
 - Napisati funkciju *ObrisiKolonu* koja uklanja zadatu kolonu iz matrice.
(6 poena)
 - Napisati funkciju *Rasporedi* koja redove matrice raspoređuje od najdužeg ka najkraćem posmatrano u smeru odozgo na dole.
(7 poena)

NAPOMENA: Maksimalan broj redova i kolona matrice nije unapred poznat.

Primer ulazne datoteke za matricu prikazanu na slici:

```
8
8
1 4 2 7 4 3 8 6
3
5 7 3
5
7 4 9 3 5
1
6
7
5 7 3 8 3 6 7
3
7 3 4
3
8 8 5
2
1 9
```



Rešenje

(Sanja Milanovic 68/09)

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

typedef struct red {int n;
                  int *a;} RED;

RED *nizredova;
int br;

void Ucitaj(FILE *f)
{
    int i,j;
    fscanf(f, "%d", &br);
    nizredova=(RED*)malloc(br*sizeof(RED));
    for (i=0;i<br;i++)
    {
        fscanf(f, "%d", &nizredova[i].n);
```

```

        nizredova[i].a=(int*)malloc(nizredova[i].n*sizeof(int));
        for (j=0;j<nizredova[i].n;j++)
            fscanf(f,"%d",&nizredova[i].a[j]);
    }
}

void ObrisiElement(int m,int n)
{
    int i,k=nizredova[m].n;
    for (i=n;i<k-1;i++)
        nizredova[m].a[i]=nizredova[m].a[i+1];
    nizredova[m].n--;
}

void ObrisiRed(int m)
{
    int i;
    for (i=m;i<br-1;i++)
        nizredova[i]=nizredova[i+1];
    br--;
}

void ObrisiKolonu(int n)
{
    int i;
    for (i=0;i<br;i++)
        if (nizredova[i].n>=n) ObrisiElement(i,n);
}

void Rasporedi(void)
{
    int i,j;
    RED pom;
    for (i=0;i<br-1;i++)
        for (j=i+1;j<br;j++)
            if (nizredova[i].n<nizredova[j].n)
            {
                pom=nizredova[i];
                nizredova[i]=nizredova[j];
                nizredova[j]=pom;
            }
}

void Stampaj(void)
{
    int i,j;
    for (i=0;i<br;i++)
    {
        for (j=0;j<nizredova[i].n;j++)
            printf("%d ",nizredova[i].a[j]);
        printf("\n");
    }
}

main()
{
    int n,m;
    FILE *f;
    char file[50];
    printf("Unesite naziv datoteke:\n");
    scanf("%s",&file);
    f=fopen(file,"r");
    Ucitaj(f);
    fclose(f);
    printf("Unesite koju kolonu zelite da obrisete:\n");
    scanf("%d",&n);
    ObrisiKolonu(n);
    printf("Unesite koji red zelite da obrisete:\n");
    scanf("%d",&m);
    ObrisiRed(m);
    Rasporedi();
    Stampaj();
}

```