

Tema 3

Više dimenzioni nizovi, algoritmi za sortiranje i pokazivači

Višedimenzionalni nizovi

- Opšta sintaksa:

`tip_podatka ime[dimenzija1][dimenzija2]...[dimenzijan]`

- Primer:

```
float tabela[5][12];
```

- Pristup elementu niza:

```
b = tabela[1][3];
```

```
tabela[4][10] = 15.33;
```

Primer 1

Napisati program koji učitava dve kvadratne matrice A i B dimenzije n , ispisuje njihovu sumu $A+B$ i proizvod $A*B$.

```

#include<stdio.h>
#define MAX 10

int main() {
    int n, i, j, k;
    int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX];

    printf("Unesite velicinu celobrojnih kvadratnih matrica: ");
    scanf("%d", &n);

    printf("Unos matrice A:\n");
    for (i=0; i<n ; i++)
        for (j=0; j<n ; j++) {
            printf("A[%d][%d]=", i+1, j+1);
            scanf("%d", &A[i][j]);
        }

    /* Unesi matricu B ... */
    /* U matricu C upisujemo zbir matrica A i B:  C = A + B */
    for (i=0; i<n ; i++)
        for (j=0; j<n ; j++)
            C[i][j]=A[i][j]+B[i][j];
    printf("\nIspis matrice C=A+B:\n");
    for (i=0; i<n ; i++) {
        for (j=0; j<n ; j++)
            printf("%d\t", C[i][j]);
        printf("\n");
    }
    /* U matricu C upisujemo proizvod matrica A i B: C = A * B */
    for (i=0; i<n; i++)
        for (j=0; j<n; j++) {
            C[i][j]=0;
            for (k=0; k<n; k++)
                C[i][j]+=A[i][k]*B[k][j];
        }

    /* ...      Ispis matrice C=A*B*                               */
    return 0;
} // Završi samostalno

```

Zadatak 1

Napisati program koji učitava matricu A i ispisuje sumu elemenata koji se na sporednoj dijagonali matrice.

Zadatak 2

Napisati program koji učitava matricu A i ispisuje sumu elemenata koji se nalaze ispod glavne dijagonale matrice.

Zadatak 3

Napisati program koji učitava matricu A i razmenjuje vrstu koja sadrži element najveće vrednosti i vrstu koja sadrži lement najmanje vrednosti.

Zadatak 4

Napisati program za transponovanje kvadratne matrice sa realnim elementima. Na početku programa potrebno je učitati dimenziju i elemente matrice. (Zadatak 3.12 iz zbirke L. K.)

Zadatak 5

Napisati program za uređivanje kolona pravougaone matrice sa realnim elementima po neopadajućem redosledu zbira elemenata po kolonama. (Zadatak 3.13 iz zbirke L. K.)

Sortiranje nizova

- Postoje različiti algoritmi za sortiranje
 - selection sort
 - insertion sort
 - bubble sort
 - shell sort
 - merge sort
 - quick sort
 - ...
- (za višu ocenu potrebno je znati neki složeniji (brži) algoritam – shell, merge ili quick)

Zadatak 6

Dat je niz od maksimalno 20 realnih elemenata. Učitati n elemenata, a zatim sortirati niz u rastućem redosledu.

- koristiti algoritam po izboru

Pokazivači

- Pokazivač je adresa u memoriji, broj.
 - ukazuje na lokaciju u memoriji
 - sadržaj pokazivačke promenljive je adresa (lokacija u memoriji).
- Pokazivačke promenljive pokazuju na druge promenljive ili na početak memorijskog bloka
 - pokazivači na promenljive tipa int, float, itd.

Pokazivači

- Deklaracija:

```
int a = 5;
```

```
int b, c;
```

```
int *p1, *p2;
```

- Dodela vrednosti:

```
p1 = &a;
```

```
p2 = &b;
```

- Pristup lokaciji:

```
c = *p1;    // c <- a
```

```
*p2 = 6;    // b <- 6
```

p1	11000	10000
p2	11002	10002
...		
a	5	11000
b	6	11002

Referenciranje

- Unarni operator & daje adresu promenljive
- Izraz `p=&a` dodeljuje adresu promenljive a promenljivoj p, pa sada p pokazuje na a
- Da bi se odštampana vrednost pokazivača koristi se konverzija %p.

Dereferenciranje

- Primenom unarnog operatora * može se posredno pristupiti nekom podatku pomoću memorijske adrese.

Specijalna konstanta NULL

- Konstanta koja se nalazi u `stdio.h`
- Ako pokazivačka promenljiva ima vrednost NULL, onda ne pokazuje ni na šta.
- Primer:

```
int *p;  
p = NULL;
```
- Neinicijalizovana vrednost NEMA NULL!
 - mora se eksplicitno inicijalizovati na NULL

Primer 2

```
1  #include <stdio.h>
2
3  int main() {
4      int i;
5      int *pi;
6
7      i = 7;
8      pi = &i;
9
10     printf("Promenljiva - adresa:\t %p, vrednost:\t %d\n\n", &i, i);
11     printf("Pokazivac   - adresa:\t %p, vrednost:\t %p\n\n", &pi, pi);
12     printf("Pokazivac   - vrednost:\t %p, sadrzaj:\t %d\n\n", pi, *pi);
13
14     i = 10;
15
16     printf("Pokazivac   - vrednost:\t %p, sadrzaj:\t %d\n\n", pi, *pi);
17
18     (*pi)++;
19
20     printf("Promenljiva - adresa:\t %p, vrednost:\t %d\n\n", &i, i);
21
22     return 0;
23 }
```

Pokazivači i nizovi

- Nizovi se mogu posmatrati kao pokazivači
- Kada se definiše niz, alocira se navedeni broj memorijskih lokacija za smeštaj elemenata niza. Promenljiva koja predstavlja niz se postavlja tako da pokazuje na prvu od ovih lokacija.

Pokazivači i nizovi

- Identifikator niza je zapravo pokazivač na prvi element u memoriji (koja je dodeljena tom nizu).
- Primer:

```
int a[20]; // a - je identifikator niza  
int *pa;
```

`pa = a;` je isto što i: `pa = &a[0];`

`a+2 == &a[2] == pa+2`

`*(a+3) == a[3] == *(pa+3)`

Primer 3 - pristup elementima niza preko pokazivača.

```
1  #include <stdio.h>
2
3  int main ( )
4  {
5      /* Niz se moze inicijalizovati i prilikom deklaracije */
6      int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
7
8      int *pa;
9      int i;
10
11     pa = a;
12     printf("%d\n\n", *pa);
13
14     printf("%d\n\n", *(a+1));
15
16     // ispis elementa sa parnim indexom
17     for (i=0; i<10; i+=2) {
18         pa = a+i;
19         printf("%d\n", *pa);
20     }
21
22     return 0;
23 }
```

Vrlo česte greške

- Nemoguće je definisati pokazivač na konstantu ili izraz.
- Nemoguće je promeniti adresu promenljive (jer to i ne zavisi od nas).
- Zbog ovoga sledeći izrazi su pogrešni:
 - `i = &3;`
 - `j = &(k+5);`
 - `k = &(a==b);`
 - `&a = &b;`
 - `&a = 150;`

Zadatak 7

Napisati program za obrtanje redosleda elemenata (zamena prvog sa poslednjim, drugog sa pretposlednjim, itd.) zadatog niza celih brojeva. Za pristup elementima niza koristiti pokazivače. (Zadatak 4.2 iz zbirke L. K.)