

Laboratorijska vježba 6

U ovoj vježbi studenti će raditi naprednije zadatke iz oblasti nizova, kao što je sortiranje, te upoznati se sa konceptom višedimenzionalnog niza (na konkretnom primjeru 2D niza). Zadaci uključuju učitavanje elemenata niza, ispis elemenata niza, te neke manipulacije nad dvodimenzionalnim nizom.

I UVODNI DIO VJEŽBE

Unesite sljedeći program sa predavanja (Selection Sort):

```
#include <stdio.h>
#define DUZINA 10
int main() {
    double niz[DUZINA], temp;
    int i, j, min;
    /* unos niza */
    for (i=0; i<DUZINA; i++) {
        printf("Unesite %d. realan broj: ", i+1);
        scanf("%lf", niz[i]);
    }

    for (i=0; i<DUZINA; i++) {
        min=i;
        for (j=i+1; j<n; j++) {
            if (niz[j] < niz[min])
                min = j;
            temp = niz[i];
            niz[i] = niz[min];
            niz[min] = temp;
        }

        /* ispis niza */
        printf("Sortiran niz:\n");
        for (i=0; i<DUZINA; i++)
            printf("%g ", niz[i]);
        printf("\n");
    }
    return 0;
}
```

Selection Sort je jedan od algoritama za sortiranje (slaganje niza po veličini) koji je krajnje neefikasan, ali je lagan za razumijevanje i ima kratak kod. Drugi algoritmi se obrađuju na predmetu Algoritmi i strukture podataka (3. semestar RI). Do kraja semestra ćemo koristiti sortiranje u nizu zadataka, stoga je preporučljivo da ga naučite i shvatite!

- a) Analizirajte program.
- b) Kompajlirajte program.
- c) Testirajte program tako što ćete unositi različite ulazne vrijednosti.

II ZADACI ZA PROVJERU RAZUMIJEVANJA UVODNOG ZADATKA

2. Dati program sortira brojeve od manjih ka većim. Prepravite ga tako da ih slaže od većih ka manjim.

III ZADACI ZA SAMOSTALNU VJEŽBU

3. Napisati program koji najprije traži da se unesu dva niza pozitivnih cijelih brojeva A i B. Unos niza se prekida kada korisnik unese broj -1, a maksimalan broj elemenata u nizovima je 10. Nakon unosa, nizove A i B je potrebno spojiti u novi niz C. Konačno, tako formirani niz C treba ispisati na ekranu.

Ovaj zadatak je vrlo jednostavno riješiti bez korištenja niza C, no za vježbu pokušajte ga uraditi tačno po koracima koji su dati iznad.

Primjer ulaza i izlaza:

```
Unesite elemente niza A: 5 3 8 -1
Unesite elemente niza B: 2 3 1 5 1 -1
Niz C glasi: 5,3,8,2,3,1,5,1
```

***Napomena:** U gornjem primjeru demonstrirana je pojava da se više uzastopnih naredbi `scanf` može izvršiti tako što se vrijednosti unose razdvojene razmakom. Npr. ako kod glasi:*

```
scanf ("%d", &a) ;
scanf ("%d", &b) ;
scanf ("%d", &c) ;
```

korisnik može kucati

```
1 2 3 (Enter)
```

i ova tri broja će biti upisana u varijable a, b i c respektivno.

4. Napisati program koji iz datog niza cijelih brojeva izbacuje uneseni element pri čemu se ne mijenja redoslijed ostalih elemenata. Korisnik najprije unosi niz cijelih pozitivnih brojeva (-1

za kraj unosa, maksimalno 10 elemenata). Zatim se unosi element koji treba izbaciti. Program treba iz niza izbaciti sva pojavljivanja unesenog elementa, i na kraju ispisati novi niz na ekranu.

Primjer:

```
Unesite elemente niza: 2 3 1 5 1 -1
Unesite element koji treba izbaciti: 1
Novi niz glasi: 2,3,5
```

Napomena: Ponovo, ovaj zadatak se može uraditi tako što se u petlji za ispis na ekran stavi npr.:

```
if (niz[i] != trazeni) printf(...)
```

ali radi vježbe probajte uraditi zadatak tako da se najprije izbac element iz niza, a zatim ispiše tako modifikovan niz. Ovakav zadatak je vrlo važan za drugi parcijalni ispit! U mnogim zadacima na ispitu traži se neka varijanta izbacivanja člana iz niza (nevezano za ispis na ekranu) što studenti često ne uspiju ispravno uraditi.

5. Napisati program koji učitava matricu cijelih brojeva dimenzija 10x10 te pronalazi najmanji element na glavnoj dijagonali.

6. Napravite program koji učitava realnu matricu dimenzija A×B te pronalazi sumu elemenata na rubu matrice. Korisnik najprije unosi dimenzije A i B (ne veće od 100) a zatim unosi elemente matrice. Rub matrice sačinjavaju elementi prvog i zadnjeg reda, te elementi prve i zadnje kolone. Pri tome pazite da se elementi u ćoškovima ne budu dvaput uračunati u sumu! Primjer:

a_{11}	a_{12}	a_{13}	...	a_{1n}
a_{21}	a_{22}	a_{23}	...	a_{2n}
a_{31}	a_{32}	a_{33}	...	a_{3n}
		...		
a_{m1}	a_{m2}	a_{m3}	...	a_{mn}

$$Suma = a_{11} + a_{12} + a_{13} + \dots + a_{1n} + a_{2n} + a_{3n} + \dots + a_{mn} + \dots + a_{m3} + a_{m2} + a_{m1} + \dots + a_{31} + a_{21}$$

Primjer ulaza i izlaza:

```
Unesite dimenzije matrice: 2 2
Unesite elemente matrice: 1 2 3 4
Suma elemenata na rubu je 10.00
```

7. Napravite program koji učitava matricu cijelih brojeva dimenzija 10x10, a zatim pronalazi red sa najvećim i red sa najmanjim elementom. Ako se oba elementa nalaze u istom redu, program treba ispisati poruku "Najveci i najmanji element se nalaze u istom redu." i završiti s radom. U suprotnom, potrebno je zamijeniti mjesta ta dva reda u matrici i ispisati tako modifikovanu matricu na ekranu.

Primjer ulaza i izlaza:

Unesite elemente matrice: 1 2 3 4 ...

Nakon zamjene matrica glasi:

1 2 3 4 ...