

Programiranje II

Beleške sa vežbi

Smer *Informatika*
Matematički fakultet, Beograd

Sana Stojanović

28.02.08.

Sadržaj

1	Matrice	3
2	Pokazivači - ponavljanje	8

1 Matrice

1. Napisati program koji iz datoteke "matrica.txt" učitava prvo dimenziju kvadratne matrice ($n < 10$) a zatim redom elemente matrice (u okviru jedne dvostruke *for* petlje) pa zatim ispisuje elemente matrice na standardni izlaz (u okviru druge dvostruke *for* petlje). Datoteka može da izgleda npr. ovako:

```
3

1 2 3
4 5 6
7 8 9

#include <stdio.h>

main()
{
    int a[10][10];      /* Deklaracija matrice */
    int n;              /* Dimenzija kvadratne matrice */
    FILE* in;          /* Datoteka iz koje učitavamo matricu */
    int i, j;

    /* Otvaramo datoteku za citanje */
    in=fopen("matrica.txt", "r");
    if (in==NULL)
    {
        fprintf(stderr, "\nGreska pri otkrivanju datoteke: matrica.txt\n");
        exit(1);
    }

    /* Učitavamo dimenziju kvadratne matrice */
    fscanf(in, "%d", &n);

    /* Učitavamo elemente kvadratne matrice */
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            fscanf(in, "%d", &a[i][j]);

    /* Ispisujemo elemente kvadratne matrice. Prilikom ispisivanja
    elemenata matrice treba da vodimo racuna o tome da treba da
    ispisemo i karakter za novi red nakon svakog ispisanoog reda
    matrice. */
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
```

```

        printf("%d\t", a[i][j]);
    printf("\n");
    }
}

```

2. Napisati program koji iz datoteke "matrica.txt" učitava prvo dimenziju kvadratne matrice ($n < 10$) a zatim redom elemente matrice. Odrediti sumu elemenata matrice, sumu elemenata koji se nalaze na glavnoj dijagonali, sumu elemenata koji se nalaze iznad glavne dijagonale.

```

#include <stdio.h>

main()
{
    int a[10][10];    /* Deklaracija matrice */
    int n;            /* Dimenzija kvadratne matrice */
    FILE* in;        /* Datoteka iz koje učitavamo matricu */
    int i, j;
    int suma, suma_dijag, suma_iznad;

    /* Otvaramo datoteku za citanje */
    in=fopen("matrica.txt", "r");
    if (in==NULL)
    {
        fprintf(stderr, "\nGreska pri otkrivanju datoteke: matrica.txt\n");
        exit(1);
    }

    /* Učitavamo dimenziju kvadratne matrice */
    fscanf(in, "%d", &n);

    /* Učitavamo elemente kvadratne matrice */
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            fscanf(in, "%d", &a[i][j]);

    /* Dvostruka for petlja koja izracunava sumu elemenata matrice */
    suma = 0;
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            suma = suma + a[i][j];

    printf("Suma elemenata matrice je: %d\n", suma);

    /* Dvostruka petlja koja izracunava sumu elemenata koji se nalaze
    na glavnoj dijagonali (to su elementi kojima su indeksi "i" i "j"

```

```

        jednaki */
suma_dijag = 0;
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        if (i==j)
            suma_dijag = suma_dijag + a[i][j];

printf("Suma elemenata na glavnoj dijagonali je: %d\n", suma_dijag);

/* Kako znamo da su elementi na glavnoj dijagonali a[i][j] odredjeni
jednakoscju i==j mozemo na efikasniji nacin izracunati sumu tih
elemenata:

suma_dijag = 0;
for(i=0; i<n; i++)
    suma_dijag = suma_dijag + a[i][i];
*/

/* Dvostruka petlja koja izracunava sumu elemenata iznad glavne
dijagonale (to su elementi kojima indeksi "i" i "j" zadovoljavaju
jednakost i<j) */
suma_iznad = 0;
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        if (i < j)
            suma_iznad = suma_iznad + a[i][j];

printf("Suma elemenata iznad glavne dijagonale je: %d\n", suma_iznad);

/* Slicno kao u prethodnom slucaju, kako elementi iznad glavne dijagonale
imaju indkse koji zadovoljavaju uslov i<j mozemo na efikasniji
nacin izracunati sumu tih elemenata:

suma_iznad = 0;
//brojac "i" i dalje pustamo da ide od 0 do n
for(i=0; i<n; i++)
    //ali brojac "j" opisujemo tako da cuva relaciju i<j pa
    //zato je pocetna vrednost j=i+1
    for(j=i+1; j<n; j++)
        suma_iznad = suma_iznad + a[i][j];
*/
}

```

3. Napisati program koji iz datoteke "matrica.txt" učitava prvo dimenziju kvadratne matrice ($n < 10$) a zatim redom elemente matrice. Za uneto k sa standardnog ulaza ispisati sumu elemenata u k -tom redu i sumu

elemenata u k -toj koloni.

```
#include <stdio.h>

main()
{
    int a[10][10];      /* Deklaracija matrice */
    int n;              /* Dimenzija kvadratne matrice */
    FILE* in;          /* Datoteka iz koje učitavamo matricu */
    int i, j, k, suma;

    /* Otvaramo datoteku za citanje */
    in=fopen("matrica.txt", "r");
    if (in==NULL)
    {
        fprintf(stderr, "\nGreska pri otkrivanju datoteke: matrica.txt\n");
        exit(1);
    }

    /* Učitavamo dimenziju kvadratne matrice */
    fscanf(in, "%d", &n);

    /* Učitavamo elemente kvadratne matrice */
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            fscanf(in, "%d", &a[i][j]);

    printf("Unesite ceo broj:\n");
    scanf("%d", &k);

    /* Racunanje sume elemenata u k-tom redu. To su elementi kojima je
       prvi indeks jednak k pa pomocu samo jedne for petlje mozemo da
       dobijemo zeljenu sumu. */
    suma = 0;
    for(j=0; j<n; j++)
        suma = suma + a[k][j];

    printf("Suma elemenata koji se nalaze u %d-tom redu je: %d\n", k, suma);

    /* Racunanje sume elemenata u k-toj koloni (elementi kojima je drugi
       indeks jednak k. */
    suma = 0;
    for(i=0; i<n; i++)
        suma = suma + a[i][k];
}
```

```
}
```

4. Napisati program koji iz datoteke "matrica.txt" učitava prvo dimenziju kvadratne matrice ($n < 10$) a zatim redom elemente matrice. Datoteka može da izgleda npr.

```
3
```

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

Nakon toga ispisati na standardni izlaz elemente matrice po trakama paralelnim glavnoj dijagonali. Tj. ispisati elemente matrice na sledeći način (ako posmatramo elemente matrice preko njihovih koeficijenata):

```
a[0][2]
a[0][1] a[1][2]
a[0][0] a[1][1] a[2][2]
a[1][0] a[2][1]
a[2][0]
```

Odnosno, za matricu navedenu u ovom primeru izlaz treba da izgleda ovako:

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

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

```
main()
```

```
{
```

```
    int a[10][10];    /* Matrica koju učitavamo */
    int n;            /* Dimenzija kvadratne matrice */
    FILE* in;
    int i, j, d;
```

```
    /* Otvaramo datoteku za citanje */
```

```
    in=fopen("matrica.txt", "r");
```

```
    if (in==NULL)
```

```
    {
```

```
        fprintf(stderr, "\nGreska pri otkrivanju datoteke: matrica.txt\n");
```

```

        exit(1);
    }

    /* Ucitavamo dimenziju kvadratne matrice */
    fscanf(in, "%d", &n);

    /* Ucitavamo elemente kvadratne matrice */
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            fscanf(in, "%d", &a[i][j]);

    /* Ispis matrice paralelno glavnoj dijagonali podelicemo na dva dela.
       Ispis matrice iznad glavne dijagonale (ukljucujuci i glavnu
       dijagonalu) i ispis matrice ispod glavne dijagonale */

    /* Ispis matrice iznad glavne dijagonale */
    /* Primetimo da su razlike indeksa elemenata matrice u svakoj traci
       jednake. Iz tog razloga stampanje elemenata ce se odvijati u
       petlji ciji ce brojac predstavljati razliku tih indeksa. */
    for(d = n-1; d >= 0; d--)
    {
        /* U okviru jedne trake primetimo da prvi indeks uvek krece od 0
           i da raste, a da se drugi indeks zavrsava na n-1 uvek. */
        for(i=0; d+i <= n-1; i++)
            printf("%d ", a[i][d+i]);
        printf("\n");
    }

    /* Ispis matrice ispod glavne dijagonale */
    /* Opet vazi da su razlike indeksa u svakoj traci jednake, jedina
       razlika je u tome sto je sada prvi indeks veci od drugog */
    for(d=1; d <= n-1; d++)
    {
        for(i=d; i <= n-1; i++)
            printf("%d ", a[i][i-d]);
        printf("\n");
    }
}

```

2 Pokazivači - ponavljanje

```

1. /* Pokazivaci - osnovni pojam */
#include <stdio.h>
main()
{

```



```

int x = 3;

/* Adresu promenljive x zapamticemo u novoj promeljivoj.
   Nova promenljiva je tipa pokazivaca na int (int*) */
int* px;

printf("Adresa promenljive x je : %p\n", &x);
printf("Vrednost promenljive x je : %d\n", x);

px = &x;
printf("Vrednost promenljive px je (tj. px) : %p\n", px);
printf("Vrednost promenljive na koju ukazuje px (tj. *px) je: %d\n", *px);

/* Menjamo vrednost promenljive na koju ukazuje px */
*px = 6;
printf("Vrednost promenljive na koju ukazuje px (tj. *px) je: %d\n", *px);

/* Posto px sadrzi adresu promenljive x, ona ukazuje na x tako da je
   posredno promenjena i vrednost promenljive x */
printf("Vrednost promenljive x je : %d\n", x);

}

```

2. Napisati funkciju koja menja vrednosti svojim argumentima i program koji je poziva.

```

/* swap : Demonstracija prenosa argumenata preko pokazivaca */
#include <stdio.h>

/* Pogresna verzija funkcije swap. Zbog prenosa po vrednosti, funkcija
   razmenjuje kopije promenljivih iz main-a, a ne samih promenljivih */
void swap_wrong(int x, int y)
{
    int tmp;

    printf("swap_wrong: ");
    printf("Funkcija menja vrednosti promenljivim na adresama : \n");
    printf("x : %p\n", &x);
    printf("y : %p\n", &y);

    tmp = x;
    x = y;

```

```

    y = tmp;
}

/* Resenje je prenos argumenata preko pokazivaca */
void swap(int* px, int* py)
{
    int tmp;

    printf("swap : Funkcija menja vrednosti promenljivim na adresama : \n");
    printf("px = %p\n", px);
    printf("py = %p\n", py);

    tmp = *px;
    *px = *py;
    *py = tmp;
}

main()
{
    int x = 3, y = 5;
    printf("Adresa promenljive x je %p\n", &x);
    printf("Vrednost promenljive x je %d\n", x);
    printf("Adresa promenljive y je %p\n", &y);
    printf("Vrednost promenljive y je %d\n", y);

    /* Pokusavamo zamenu koristeći pogresnu verziju funkcije */
    swap_wrong(x, y);

    printf("Posle swap_wrong:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);

    /* Vrsimo ispravnu zamenu. Funkciji swap saljemo adrese promenljivih
       x i y, a ne njihove vrednosti */
    swap(&x, &y);

    printf("Posle swap:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);
}

```

3. Napisati funkciju `void podeli(int n, int m, int *ceo, int *ost)` koja računa ceo deo i ostatak pri deljenju broja n brojem m i program koji sa standardnog ulaza učitava cele brojeve n i m i pozivom funkcije `podeli` određuje ceo deo i ostatak pri deljenju prvog broja drugim i štampa ih na

standardni izlaz.

```
#include <stdio.h>

/* Funkcija koja pomocu pokazivaca racuna ceo deo i ostatak pri deljenju
   jednog broja drugim. Koriscenje pokazivaca je u ovom slucaju neophodno
   posto funkcija ima dve povratne vrednosti. */
void podeli(int n, int m, int *ceo, int *ost)
{
    *ceo = n/m;
    *ost = n%m;
}

main()
{
    int n,m;
    int ceo, ost;

    printf("Unesite dva cela broja:\n");
    scanf("%d%d", &n, &m);

    /* Poziv funkcije. Kako zelimo da nam funkcija preko pokazivaca
       izracuna ceo deo i ostatak, celobrojne promenljive prosledjujemo
       sa njihovim adresama (da bi funkcija mogla da promeni vrednost obe) */
    podeli(n, m, &ceo, &ost);

    printf("Ceo deo pri deljenju broja %d sa %d je %d\n", n, m, ceo);
    printf("Ostatak pri deljenju broja %d sa %d je %d\n", n, m, ost);
}
```

4. Napisati funkciju void srednji(int a[], int n, int* sr, int* br) koja racuna srednju vrednosti niza celih brojeva i broj elemenata niza koji su veći od te srednje vrednosti. Napisati program koji sa standardnog ulaza ucitava prvo broj elemenata niza (ceo broj manji od 20) pa zatim elemente niza, pozivom funkcije srednji izracunava srednju vrednost niza i broj elemenata koji su veći od proseka i štampa te vrednosti na standardni izlaz.

```
#include <stdio.h>

void srednji(int a[], int n, int* sr, int* br)
{
    int i, suma;

    /* Da bi izracunali srednju vrednost niza moramo prvo izracunati
       sumu svih elemenata niza */
```

```

suma = 0;
for(i=0; i<n; i++)
    suma = suma + a[i];

/* Srednja vrednost niza se dobija kada se suma svih elemenata niza
   podeli brojem elemenata niza */
*sr = suma/n;

/* Sada brojimo elemente koji su veci od prosecnog */
*br = 0;
for(i=0; i<n; i++)
    if (a[i] > *sr)
        (*br)++;    //Obratiti paznju: ovde moraju stajati zagrade!
}

main()
{
    int a[20];
    int n, i;
    int sr, br;

    printf("Unesite broj elemenata niza:\n");
    scanf("%d", &n);

    printf("Unesite elemente niza:\n");
    printf("%d", &a[i]);

    srednji(a, n, &sr, &br);

    printf("Srednja vrednost niza je %d\n", sr);
    printf("Broj elemenata niza koji su veci od prosecnog je %d\n", br);
}

```