Programiranje I

Beleške sa vežbi

Smer *Informatika*

Matematički fakultet, Beograd

Sana Stojanović

November 14, 2007

Sadržaj

1 Unos i ispis podataka **3**

2 Nizovi, deklaracija i inicijalizacija **3**

3 Karakterske niske **5**

4 Funkcije **6**

5 Oblast važenja lokalnih promenljivih **9**

1 Unos i ispis podataka

1. Napisati program koji od korisnika traži da unese dva cela broja koja zatim sabira i ispisuje rezultat na standardni izlaz. Program bi trebalo da izgleda ovako:

```
Unesite prvi broj: 3
Unesite drugi broj: 4
Zbir brojeva 3 i 4 je: 7
```

```
#include <stdio.h>

main()
{
    int x, y;    //promenljive u kojima cemo cuvati unesene brojeve
    int z;       //promenljiva u kojoj cemo cuvati rezultat

    //Unosenje prvog broja
    printf("Unesite prvi broj: ");
    scanf("%d", &x);

    //Unosenje drugog broja
    printf("Unesite drugi broj: ");
    scanf("%d", &y);

    //Racunanje rezultata
    z = x + y;

    //Stampanje rezultata u zeljenom obliku
    printf("Zbir brojeva %d i %d je: %d\n", x, y, z);
}
```

2 Nizovi, deklaracija i inicijalizacija

Naredbom:

```
int niz[10];
```

deklarišemo celobrojni niz od 10 elemenata čiji elementi su redom: $niz[0], niz[1], \ldots, niz[9]$.

Naredbom:

```
int niz[4] = {1, 2, 3, 4};
```

deklarišemo i inicijalizujemo celobrojni niz od 4 elementa čiji elementi su redom: $niz[0] = 1, niz[1] = 2, niz[2] = 3, niz[3] = 4$

1. Napisati program koji u sebi sadrži deklaraciju i inicijalizaciju niza od 4 celobrojna elementa koji sa standardnog ulaza učitava ceo broj između 0 i 3 (uključujući i 0 i 3, [1]) i zatim ispisuje element niza koji se nalazi na toj (unetoj) poziciji. Program bi trebalo da izgleda ovako:

```
Unesite indeks niza (izmedju 0 i 3): 1
Element niza koji se nalazi na poziciji 1 je 5
```

```
#include <stdio.h>

main()
{
    int niz[4] = {1, 2, 3, 4};  //Deklaracija i inicijalizacija niza
    int i;                      //Indeks niza

    printf("Unesite indeks niza (izmedju 0 i 3): ");
    scanf("%d", &i);

    //sa niz[i] pristupamo elementu na i-toj poziciji
    printf("Element niza koji se nalazi na poziciji %d je %d\n", i, niz[i]);
}
```

2. Napisati program koji u sebi sadrži deklaraciju i inicijalizaciju niza od 4 celobrojna elementa koji sa standardnog ulaza učitava ceo broj između 0 i 3 (uključujući i 0 i 3), zatim sa standardnog ulaza učitava novu vrednost koju želimo da upišemo na tu poziciju i zatim ispisuje šta se nalazi (nakon promene) na toj poziciji niza. Program bi trebalo da izgleda ovako:

```
Unesite indeks niza (izmedju 0 i 3): 1
Na poziciji 1 nalazi se: 2
Unesite vrednost koju zelite da upisete na tu poziciju: 5
Element niza koji se nalazi na poziciji 1 je 5
```

```
#include <stdio.h>

main()
{
    int niz[4] = {1, 2, 3, 4};  //Deklaracija i inicijalizacija niza
    int i;                      //Indeks niza
    int x;                      //Promenljiva u kojoj cemo cuvati novu
                                //vrednost elementa niza

    printf("Unesite indeks niza (izmedju 0 i 3): ");
```

[1] Zbog granica niza moramo voditi računa o tome da nikada ne pristupamo elementima koji ne pripadaju našem nizu, kao što su, u slučaju niza od 4 elemenata niz[4], niz[5], ...

```
    scanf("%d", &i);

    printf("Na poziciji %d nalazi se: %d\n", i, niz[i]);

    printf("Unesite novu vrednost koju zelite da upisete na tu poziciju: ");
    scanf("%d", &x);

    //upisujemo novu vrednost na i-tu poziciju
    niz[i] = x;

    /* Umesto prethodne dve naredbe (scanf("%d", &x); niz[i] = x; )
       mogli smo krace da zapisemo:
       scanf("%d", &niz[i]); */

    //sa niz[i] pristupamo elementu na i-toj poziciji
    printf("Element niza koji se nalazi na poziciji %d je %d\n", i, niz[i]);
}
```

3 Karakterske niske

Naredbom

```
char tekst[] = {'Z', 'd', 'r', 'a', 'v', 'o'};
```

deklarišemo niz od 6 karaktera sa indeksima 0 do 5. Drugi način da deklarišemo karaktersku nisku je:

```
char tekst1[] = "Zdravo";
```

Razlika između ove dve inicijalizacije jeste da u drugom slučaju dobijamo niz od 7 karaktera (`'Z', 'd', 'r', 'a', 'v', 'o', '\0'`), odnosno rezerviše se i mesto za terminalnu nulu na kraju niske.

1. Napisati program koji inicijalizuje karaktersku nisku na ime i prezime studenta (na primer, "Petar Petrovic") i nakon toga ispisuje ime tog studenta pozivom funkcije *printf* [2].

```
#include <stdio.h>

main()
{
    char ime[] = "Petar Petrovic";

    printf("Ime studenta je: %s\n", ime);
```

[2]Kada funkciji printf predajemo kao argument karaktersku nisku, to mora biti niska koja u sebi sadrži terminalnu nulu na kraju. Ovo važi i za sve ostale funkcije koje rade sa karakterskim niskama.

```
}

Izlaz iz programa:
Ime studenta je: Petar Petrovic
```

4 Funkcije

1. Napisati funkciju koja sabira dva cela broja i program koji poziva tu funkciju. Sa standardnog ulaza unosimo dva cela broja i onda pozivom funkcije koja sabira dva broja računamo zbir i ispisujemo ga.

```
#include <stdio.h>

/* Definicija funkcije koja sabira dva cela broja.
   Funkcija ima dva argumenta tipa int i vraca povratnu vrednost tipa int */
int saberi(int x, int y)
{
    int zbir;        //uvodimo pomocnu promenljivu
    zbir = x + y;    //racunamo zbir
    return zbir;     //naredbom return vracamo izraz pozivnoj funkciji
}

main()
{
    int a, b, c;

    printf("Unesite prvi broj: ");
    scanf("%d", &a);
    printf("Unesite drugi broj: ");
    scanf("%d", &b);

    //poziv funkcije saberi
    c = saberi(a, b);

    printf("Zbir brojeva %d i %d je %d\n", a, b, c);

    /* Umesto prethodne dve naredbe mogli smo da napisemo i
       printf("Zbir brojeva %d i %d je %d\n", a, b, saberi(a, b)); */
}
```

2. Napisati funkciju koja računa kvadrat broja i program koji je poziva.

```
#include <stdio.h>

int kvadrat(int x)
```

```
{
    int kv;
    kv = x*x;
    return kv;
}

main()
{
    int a;

    printf("Unesite vrednost broja: ");
    scanf("%d", &a);

    //Poziv funkcije koja racuna kvadrat broja
    printf("Kvadrat broja %d je %d\n", a, kvadrat(a));
}
```

3. Napisati funkciju koja računa cifru najmanje težine broja.

```
int cifra(int x)
{
    return x%10;
}
```

4. Napisati funkciju koja za ceo broj n vraća vrednost 0 ako je broj paran, odnosno 1 ako je broj neparan.

```
int neparan(int n)
{
    return n%2;
}
```

5. Napisati funkciju koja za ceo broj n vraća vrednost 1 ako je broj paran, odnosno 0 ako je broj neparan.

```
int paran(int n)
{
    return 1-x%2;
}
```

U programima do sad, definiciju funkcije smo celu navodili pre funkcije *main*. Moguće je navoditi definiciju funkcije i nakon funkcije *main* ako se pre funkcije *main* navede prototip funkcije.

Prototip funkcije (koja računa ostatak prilikom deljenja broja x brojem y) je:

```
int ostatak(int x, int y);
```

Na kraju ove naredbe oznaka za kraj naredbe (;) je neophodna. S obzirom da, u ovom slučaju, prototip funkcije navodimo samo da bi se znalo koji je 'tip' funkcije, imena promenljivih (na ovom mestu) nisu neophodna. Tako da bi prototip funkcije mogao da bude zapisan i ovako:

```
int ostatak(int, int);
```

6. Napisati program koji za dva uneta cela broja sa standardnog ulaza poziva funkciju koja računa ostatak prilikom deljenja prvog broja sa drugim i ispisuje rezultat na standradni izlaz. Program bi trebalo da izgleda ovako:

```
Unesite prvi broj: 9
Unesite drugi broj: 3
Ostatak pri deljenju 9 sa 3 je: 0
```

```
#include <stdio.h>

/* Prototip funkcije koja racuna ostatak pri deljenju prvog broja drugim
   brojem. Funkcija prima kao argumente dva cela broja i vraca ceo broj kao
   povratnu vrednost. */
int ostatak(int x, int y);

main()
{
    int a, b;

    printf("Unesite prvi broj: ");
    scanf("%d", &a);

    printf("Unesite drugi broj: ");
    scanf("%d", &b);

    printf("Ostatak pri deljenju %d sa %d je: %d\n", a, b, ostatak(a,b));
}

int ostatak(int x, int y)
{
    return x%y;
}
```

7. Napisati program koji razmenjuje vrednosti dvema promenljivim koje se unose sa standardnog ulaza.

```
#include <stdio.h>

main()
{
    int a, b;    //Promenljive cije vrednosti zelimo da menjamo
    int c;       //Pomocna promenljiva pomocu koje menjamo vrednosti

    printf("Unesite prvi broj: ");
    scanf("%d", &a);

    printf("Unesite drugi broj: ");
    scanf("%d", &b);

    printf("Vrednosti pre zamene su a = %a i b = %b\n", a, b);

    //Zamena vrednosti dva broja preko pomocne promenljive
    c = a;
    a = b;
    b = c;

    printf("Vrednosti nakon zamene su a = %a i b = %b\n", a, b);
}

Izlaz iz programa:
Unesite prvi broj: 3
Unesite drugi broj: 5
Vrednosti pre zamene su a = 3 i b = 5
Vrednosti nakon zamene su a = 5 i b = 3
```

5 Oblast važenja lokalnih promenljivih

1. Ilustracija korišćenja promenljivih sa istim imenom unutar iste funkcije.
   Korišćenje pomoćnog bloka.

```
#include <stdio.h>

main()
{
    int pom = 1;   //pomocna promenljiva u funciji main

    printf("Pre ulaska u unutrasnji blok pom = %d\n", pom);

    //Pravimo pomocni blok
    {
        int pom = 5;    //deklarisemo pomocnu promenljivu sa istim imenom
```

```
                        //kao u main-u
        printf("U unutrasnjem bloku pom = %d\n", pom);
    }

    printf("Nakon izlaska iz unutrasnjeg bloka pom = %d\n", pom);
}

Izlaz iz programa:
Pre ulaska u unutrasnji blok pom = 1
U unutrasnjem bloku pom = 5
Nakon izlaska iz unutrasnjeg bloka pom = 1
```