

Programiranje I

Beleške sa vežbi

Smer *Informatika*
Matematički fakultet, Beograd

Sana Stojanović

November 22, 2007

Sadržaj

1	Prenos parametara po vrednosti, primeri	3
2	Operatori i izrazi dodeljivanja vrednosti, prefiksni i postfiksni operator	4
3	Logičke i relacijske operacije	5
4	If - else naredba	5
5	For petlja	10
6	Nizovi	15

1 Prenos parametara po vrednosti, primeri

1. Demonstracija prenosa parametara po vrednosti - preneti parametri se ne mogu menjati.

```
/* Prenos parametara po vrednosti */
#include <stdio.h>

/* Funkcija koja pokusava da uveca ceo broj */
void f(int a)
{
    a++;
    printf("Vrednost promenljive u funkciji je: %d\n", a);
}

main()
{
    int a = 3;
    //poziv funkcije
    f(a);
    printf("Vrednost promenljive u main-u je %d\n", b);
}
```

2. #include <stdio.h>

```
/* Funkcija pokusava da ucita ceo broj.
   Zbog prenosa parametara po vrednosti ovo ne uspeva */
void get_num(int x)
{
    scanf("%d", &x);
    printf("U funkciji je x = %d\n", x);
}

main()
{
    int x = 0;
    printf("Unesi broj : ");
    //poziv funkcije get_num koja pokusava da promeni vrednost broja x
    get_num(x);
    //stampamo vrednost broja x u funkciji main i vidimo da se ta
    //vrednost nije promenila
    printf("U main-u je x = %d\n", x);
}
```

2 Operatori i izrazi dodeljivanja vrednosti, prefiksni i postfiksni operator

Naredba

```
i = i + 2;
```

ekvivalentna je sa naredbom:

```
i+=2;
```

Ovo važi i za sledeće operatore:

```
+ - * / % << >> ^ |
```

Tj. "izraz1 op = izraz2" je ekvivalentan sa "izraz1 = (izraz1) op (izraz2)"

1. Korišćenje prefiksnog i postfiksnog operatora ++

```
/* Ilustracija prefiksnog i postfiksnog operatora ++ */

#include <stdio.h>
main()
{
    int x, y;
    int a = 0, b = 0;

    printf("Na pocetku : \na = %d\nb = %d\n", a, b);

    /* Ukoliko se vrednost izraza ne koristi, prefiksni i
       postfiksni operator se ne razlikuju */
    a++;
    ++b;
    printf("Posle : a++; ++b; \na = %d\nb = %d\n", a, b);

    /* Prefiksni operator uvecava promenjivu pre dodele, i rezultat
       je uvecana vrednost, odnosno sledeca naredba daje isti rezultat
       kao da smo zapisali: a = a+1; x = a; */
    x = ++a;

    /* Postfiksni operator uvecava promenjivu nakon dodele, i rezultat je
       stara (neuvecana) vrednost, odnosno y = b; b = b + 1; */
    y = b++;

    printf("Posle : x = ++a; \na = %d\nx = %d\n", a, x);
    printf("Posle : y = b++; \nb = %d\ny = %d\n", b, y);
}
```

Izlaz iz programa: Na pocetku: a = 0 b = 0 Posle : a++; ++b; a = 1
b = 1 Posle : x = ++a; a = 2 x = 2 Posle : y = b++; b = 2 y = 1

3 Logičke i relacijske operacije

1. Ilustracija logickih i relacijskih operatora.

```
#include <stdio.h>

main()
{
    int a = 3>5, /* manje */
        b = 5>3, /* vece */
        c = 3==5, /* jednako */
        d = 3!=5; /* razlicito */

    //stampamo logicke vrednosti ovih izraza
    printf("3>5 - %d\n", a);
    printf("5>3 - %d\n", b);
    printf("3==5 - %d\n", c);
    printf("3!=5 - %d\n", d);

    printf("Konjunkcija : 3>5 && 5>3 - %d\n", a && b);
    printf("Disjunkcija : 3>5 || 5>3 - %d\n", a || b);
    printf("Negacija : !(3>5) - %d\n", !a);
}
```

Izlaz iz programa:
3>5 - 0
5>3 - 1
3==5 - 0
3!=5 - 1
Konjunkcija : 3>5 && 5>3 - 0
Disjunkcija : 3>5 || 5>3 - 1
Negacija : !(3>5) - 1

4 If - else naredba

1. Program ilustruje *if* naredbu i ispisuje ukoliko je uneti ceo broj negativan.

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

```

main()
{
    int a;    //deklarisanje celobrojne promenljive

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

    if (a<0)
        printf("Uneti broj je negativan!\n");
}

```

Izlaz iz programa:

Unesite ceo broj:

3

- (nema izlaza)

Unesite ceo broj:

-2

Uneti broj je negativan!

2. Program ilustruje *if-else* naredbu i proverava da li je uneti ceo broj negativan. Ako jeste ispisuje da je negativan a ako nije ispisuje da je broj veći ili jednak nuli.

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

```

main()
{
    int a;    //deklarisanje celobrojne promenljive

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

    if (a<0)
        printf("Uneti broj je negativan!\n");
    else
        printf("Uneti broj je veci ili jednak nuli!\n");
}

```

Izlaz iz programa:

Unesite ceo broj:

3

Uneti broj je veci ili jednak nuli!

Unesite ceo broj:

-2

Uneti broj je negativan!

3. Program ilustruje *if-else* naredbu i ispituje znak broja.

```
#include <stdio.h>

main()
{
    int a;    //deklarisanje celobrojne promenljive

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

    if (a<0)
        printf("Uneti broj je negativan!\n");
    else if (a == 0)
        printf("Uneti broj je nula!\n");
    else
        printf("Uneti broj je pozitivan!\n");
}
```

Izlaz iz programa:

Unesite ceo broj:

3

Uneti broj je pozitivan!

Unesite ceo broj:

0

Uneti broj je nula

Unesite ceo broj:

-2

Uneti broj je negativan!

4. Ilustracija logičkih vrednosti (0 - netačno, razlicito od 0 - tačno).

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

```
main()
```

```
{
```

```
    int a;
```

```
    printf("Unesi ceo broj : ");
```

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

```
    /* Naredbom if prelazimo u tacnu granu samo ako je vrednost izraza u
```

```

        zagradama razlicita od 0. Za vrednost izraza jednaku 0 prelazimo u
        else granu */
    if (a)
        printf("Logicka vrednost broja je : tacno\n");
    else
        printf("Logicka vrednost broja je : netacno\n");
}

```

Ulaz:

Unesi ceo broj : 3

Izlaz: Logicka vrednost broja je : tacno

Ulaz: Unesi ceo broj : 0

Izlaz: Logicka vrednost broja je : netacno

5. Ilustracija programa sa greškom. Korišćenje dodele umesto poredjenja.

```

#include <stdio.h>

```

```

main()

```

```

{

```

```

    int b;

```

```

    printf("Unesi ceo broj : ");

```

```

    scanf("%d", &b);

```

```

    /* Obratiti paznju na = umesto ==.

```

```

       Usled dodele b=0 vrednost promenljive b je promenjena na 0 i
       ceo izraz b=0 ima vrednost nula a to je logicki netacno. */

```

```

    if (b = 0)

```

```

        printf("Broj je nula\n");

```

```

    /* Kako je sada b nula, uslov b<0 je netacan pa... */

```

```

    else if (b < 0)

```

```

        printf("Broj je negativan\n");

```

```

    else

```

```

    /* ...ce se uvek izvorsavati poslednja grana. */

```

```

        printf("Broj je pozitivan\n");

```

```

}

```

Ulaz: Unesi ceo broj:-5 Izlaz: Broj je pozitivan

6. Napisati funkciju koja računa minimum dva broja i program koji je poziva.

```

int min(int a, int b)

```

```

{

```

```

    int min;

```



```

if (a<b)
    min = a;
else
    min = b;

/* Umesto if-else naredbe, prethodna dodela moze da se izvrši i
na sledeci nacin:
min = (a<b)?a:b; */

return min;
}

```

7. Program koji izracunava minimum tri broja.

I nacin:

```

#include<stdio.h>

main()
{
    int a, b, c, min;

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

    if (a < b && a < c)
        min = a;
    else if (b < c)
        min = b;
    else min = c;

    printf("%d\n", min);
}

```

II nacin:

```

#include<stdio.h>

main()
{
    int a, b, c, min;

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

```

```

scanf("%d", &c);

min = (a < b) ? (a < c) ? a : c : (b < c) ? b : c;

printf("%d\n", min);
}

```

8. Napisati funkciju koja računa minimum četiri broja korišćenjem funkcije koja računa minimum dva broja.

```

int min4(int a, int b, int c, int d)
{
    return min(min(a, b), min(c, d));
}

```

9. Koristeći funkciju **max4()** napisati program koji određuje najveću među poslednjim ciframa brojeva a, b, c, d. (npr. za a = 35, b = 140, c = 127, d = 190 štampa se 7)

Poziv funkcije:
max4(a%10, b%10, c%10, d%10);

5 For petlja

1. Napisati program koji na standardni izlaz ispisuje prvih 10 prirodnih brojeva.

```

/* Program ilustruje for petlju */

#include <stdio.h>

int main()
{
    int x;

    /* for petlja u sebi sadrzi inicijalizaciju celobrojne promenljive
       (x=1), uslov koji mora vaziti za vrednosti koje uzima ta promenljiva
       (x<=10) i korak po kome se ta promenljiva menja (x++) */
    for (x = 1; x <= 10; x++)
        printf("x = %d\n",x);
}

```

Izlaz:
x = 1
x = 2

```
x = 3
x = 4
x = 5
x = 6
x = 7
x = 8
x = 9
x = 10
```

2. Za uneto n sa standardnog ulaza izračunati zbir prvih n prirodnih brojeva.

```
#include <stdio.h>

main()
{
    int n, suma;    /* Ceo broj n i suma */
    int i;         /* Brojac u ciklusu */

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

    /* Inicijalizacija pocetne vrednosti sume */
    suma = 0;

    /* U petlji racunamo sumu 1+2+...+n */
    for(i=1; i<=n; i++)
        suma += i;

    printf("Suma brojeva od 1 do %d je %d\n", n, suma);
}
```

3. Program koji za uneto n sa standardnog ulaza računa proizvod prvih n prirodnih brojeva.

```
#include <stdio.h>

main()
{
    int n, proizvod; /* Ceo broj n i proizvod */
    int i;           /* Brojac u ciklusu */

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

    /* Inicijalizacija pocetne vrednosti proizvoda (obratite paznju da
    posto je u pitanju proizvod inicijalna vrednost je 1) */
```

```

    proizvod = 1;

    /* U petlji racunamo proizvod 1*2*...*n */
    for(i=1; i<=n; i++)
        proizvod *= i;

    printf("Proizvod brojeva od 1 do %d je %d\n", n, proizvod);
}

```

4. Napisati program koji za uneto n sa standardnog ulaza učitava n prirodnih brojeva (proizvoljnih) i računa njihov zbir.

```

#include <stdio.h>

main()
{
    int n, suma;    /* Ceo broj n i suma */
    int i, br;     /* Brojac u ciklusu i pomocna promenljiva u kojoj
                   cuvamo jedan po jedan broj koji unosimo */

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

    /* Inicijalizacija pocetne vrednosti sume */
    suma = 0;

    printf("Unesite %d celih brojeva: \n", n);
    /* U petlji racunamo sumu 1+2+...+n */
    for(i=1; i<=n; i++)
    {
        //prvo ucitavamo novi broj
        scanf("%d", &br);
        //pa ga zatim dodajemo u sumu
        suma += br;
    }

    printf("Suma brojeva od 1 do %d je %d\n", n, suma);
}

```

5. Program koji određuje minimum n brojeva sa ulaza.

```

#include<stdio.h>

main()
{
    int x, n, i, min;

```

```

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

/* Ucitavamo prvi broj */
printf("Unesite prvi broj: \n");
scanf("%d", &x);

/* Dodeljujemo pocetnu vrednost minimumu */
min = x;

/* U petlji trazimo minimum. Kako smo prvi broj ucitali van petlje
   petlja ide od 2 do n */
for(i = 2; i <= n; i++)
{
    printf("Unesite sledeci broj: \n");
    scanf("%d", &x);

    if (x < min)
        min = x;
}

printf("Minimum je %d\n", min);
}

```

6. Program koji štampa sve parne brojeve od 1 do 20.

```

#include <stdio.h>

main()
{
    int x;

    /* kako zelimo da odstampamo samo parne brojeve brojac x ide od 2
       do 20 i povecava se sa korakom 2 */
    for(x = 2; x<=20; x+=2)
        printf("x\n", x);
}

```

7. Napisati funkciju koja računa izraz x^k i program koji je poziva.

```

#include <stdio.h>

/* stepenuje x^k tako sto k puta pomnozi x pretpostavka k >= 0 */
float power(float x, int k)
{
    int i;
}

```

```

float s = 1;
for (i = 1; i<=k; i++)
    s*=x;

return s;
}

/* Verzija koja radi i za negativne izlozice */
float power_n(float x, int k)
{
    int i;

    //na pocetku pamtimo u pomocnoj promenljivoj da li je broj k
    //negativan
    int negative = k<0;

    //ako jeste menjamo vrednost k na apsolutnu vrednost
    if (negative)
        k = -k;

    //sada racunamo x^k za pozitivan broj k
    float s = 1;
    for (i = 0; i<k; i++)
        s*=x;

    //na kraju proveravamo da li je broj k u pocetku bio negativan
    //pa ako jeste vracamo 1/s inace vracamo s
    if (negative)
        return 1.0/s;
    else
        return s;
}

main()
{
    /* Poziv funkcije */
    float s = power(2.0,8);
    printf("%f\n", s);
}

```

8. Napisati funkciju koja racuna faktorijal broja i i program koji koriscenjem te funkcije racuna broj kombinacija m -te klase od n elemenata (n nad m) $= n! / (m! * (n-m)!)$, koristeći funkciju za izracunavanje faktorijela.

```

/* Funkcija koja racuna faktorijel broja */
int fakt(int n)

```

```

{
    int i, f=1;
    //faktorijal broja se racuna kao proizvod prvih n prirodnih brojeva
    for(i = 1; i <= n; i++)
        f *= i;
    return f;
}

main()
{
    int n, m;
    int br_komb;

    printf("Unesite vrednosti n i m: \n");
    scanf("%d%d", &n, &m);

    br_komb = fakt(n)/(fakt(m)*fakt(n-m));
    printf("Broj kombinacija je %d\n", br_komb);
}

```

6 Nizovi

1. Napisati program koji sa standardnog ulaza učitava elemente niza i zatim ih ispisuje.

```

#include <stdio.h>

main()
{
    int a[5]; //Deklaracija niza od 5 elemenata
    int n; //Broj elemenata niza
    int i; //Brojac u petljama
    int max; //Maksimalni element

    printf("Unesite elemente niza: \n");

    /* Ucitavamo niz brojeva */
    for (i = 0; i < n; i++)
        scanf("%d",&a[i]);

    printf("Elementi niza su: \n");
    for (i = 0; i < n; i++)
        printf("%d\n", a[i]);
}

```

2. /* Program pronalazi maksimum brojeva sa ulaza - verzija sa nizom */

```

#include <stdio.h>

main()
{
    int a[5]; //Deklaracija niza od 5 elemenata
    int n;    //Broj elemenata niza
    int i;    //Brojac u petljama
    int max;  //Maksimalni element

    printf("Unesite elemente niza: \n");

    /* Ucitavamo niz brojeva */
    for (i = 0; i < n; i++)
        scanf("%d",&a[i]);

    /* Pronalazimo maksimum tako sto ga postavljamo na prvi element niza... */
    max = a[0];
    /* ...pa zatim proveravamo za sve ostale elemente niza (zato brojac
    ide od 1 do n)... */
    for (i = 1; i < n; i++)
        /* ...da li je neki od njih veci od tekuceg maksimuma. */
        if (a[i]>max)
            max = a[i];

    /* Ispisujemo maksimum */
    printf("Max = %d\n",max);
}

```