

Osnovi programiranja

Beleške sa vežbi

Smer *Računarstvo i informatika*
Matematički fakultet, Beograd

Jelena Tomašević

December 25, 2005

Sadržaj

1	Programski jezik C	5
1.1	Tipovi, operatori i izrazi. Kontrola toka.	5
1.2	Imena promenljivih	6
1.3	Deklaracije	6
1.4	Tipovi i veličina podataka	6
1.5	Funkcije printf i scanf	7
1.5.1	Aritmetičke i relacijske operacije	8
1.5.2	Operatori i izrazi dodeljivanja vrednosti	11
1.5.3	Inkrementacija i dekrementacija	11
1.6	Relacioni i logički operatori	12
1.7	Kontrola toka — if, while, do - while, for	13
1.7.1	if	13
1.7.2	Else-if	14
1.7.3	while	16
1.7.4	do-while	16
1.7.5	Switch	19
1.8	Uslovni izraz	20

1

Programski jezik C

1.1 Tipovi, operatori i izrazi. Kontrola toka.

1

Primer 1 Program na standardni izlaz štampa "Zdravo, svete!".

```
#include <stdio.h>

main()
/*iskazi f-je main su zatvoreni u zagrade */
{
/*poziv f-je printf da odštampa poruku*/
printf("Zdravo, svete!\n");
}
```

Izlaz iz programa:
Zdravo, svete!

Primer 2 Šta je izlaz iz sledećeg programa?

```
#include <stdio.h>

main()
{
printf("Zdravo, ");
printf("svete!");
printf("\n");
}
```

Izlaz iz programa:
Zdravo, svete!

¹Zasnovano na primerima sa sajta <http://www.matf.bg.ac.yu/~filip>,
<http://www.matf.bg.ac.yu/~jelenagr>, <http://www.matf.bg.ac.yu/~milena>.

1.2 Imena promenljivih

Postoje ograničenja: u imenu se mogu pojaviti slova i cifre, potcrta "_" se smatra slovom.

Velika i mala slova se razlikuju.

```
int x, X; /*To su dve razlicite promenljive!!!*/
```

Ključne reči kao što su if, else, for, while, se ne mogu koristiti za imena promenljivih.

1.3 Deklaracije

Da bi se promenljiva mogla upotrebljavati ona se mora na početku programa deklarirati. Prilikom deklaracije može se izvršiti i početna inicijalizacija.

```
int broj; /*Deklaracija celog broja*/
int vrednost=5; /*Deklaracija i inicijalizacija celog broja*/
```

Kvalifikator const može biti dodeljen deklaraciji bilo koje promenljive da bi označio da se ona neće menjati

```
const double e=2.71828182845905
```

1.4 Tipovi i veličina podataka

Osnovni tipovi podataka:

```
int      ceo broj
char     znak, jedan bajt
float    realan broj
double   realan broj dvostruke tacnosti

char     jedan bajt, sadrzi jedan znak
int      celobrojna vrednost, 2 ili 4 bajta
float    realan broj, jednostruka tacnost
double   dvostruka tacnost
```

Postoje kvalifikatori koje pridružujemo osnovnim tipovima short(16) i long(32):

```
short int kratak_broj;
long int dugacak_broj;
short kratak;
long dugacak;
```

Važi

$$\text{broj_bajtova}(\text{short}) \leq \text{broj_bajtova}(\text{int}) \leq \text{broj_bajtova}(\text{long})$$

Postoje kvalifikatori signed i unsigned koji se odnose na označene i neoznačene cele brojeve. Npr.

signed char: -128 do 127

dok je

unsigned char: od 0 do 255.

Float, double i long double.

Primer 3 Uvođenje promenljivih u program.

```
#include <stdio.h>

main()
{
    /*deklaracija vise promenljivih
    istog tipa */
    int rez,pom1,pom2;
    pom1=20;
    pom2=15;
    rez=pom1-pom2;

    /*ispisivanje rezultata*/
    printf("Rezultat je %d-%d=%d\n",pom1,pom2,rez);
}
```

Izlaz iz programa:
Rezultat je 20-15=5

Iskaz dodele:
 pom1=20;
 pom2=15;
 Individualni iskazi se završavaju sa ;

1.5 Funkcije printf i scanf

```
printf("%d\t%d\n", broj1, broj2);
```

uvek je prvi argument izmedju " "

%d ceo broj

\t tab izmedju

\n novi red

Svaka % konstrukcija je u paru sa argumentom koji sledi.

Primer 4

```
#include <stdio.h>
main()
{
    printf("Slova:\n%3c\n%5c\n", 'z' , 'Z');
}
```

Izlaz iz programa:

Slova:

```
z
Z
```

%c je za stampanje karaktera
 %3c je za stampanje karaktera na tri pozicije
 Isto tako smo mogli i %3d za stampanje broja na tri pozicije ili %6d za stampanje broja na 6 pozicija.

Pravila:

%d stampaj kao ceo broj
 %6d stampaj kao ceo broj širok najvise 6 znakova
 %f stampaj kao realan broj
 %6f stampaj kao realan broj širok najvise 6 znakova

`%.2f` stampaj kao realan broj sa dve decimale
`%6.2f` stampaj kao realan broj širok najviše 6 znakova a od toga 2 iza decimalne tacke
`%c` karakter
`%s` string
`%x` heksadecimalni broj
`%%` je procenat

Primer 5 *Prikazuje unos celog broja koristeći `scanf("%d", &x)`*

```
#include <stdio.h>

main()
{
    int x;
    printf("Unesi ceo broj : ");

    /* Obratiti paznju na znak &
       (operator uzimanja adrese)
       pre imena promenljive u funkciji
       scanf */
    scanf("%d",&x);

    /* U funkciji printf nije
       potrebno stavljati & */
    printf("Uneli ste broj %d\n", x);
}
```

Primer 6 *Program sabira dva uneta cela broja*

```
#include <stdio.h>

main()
{
    int a, b, c;
    printf("Unesi prvi broj : ");
    scanf("%d", &a);
    printf("Unesi drugi broj : ");
    scanf("%d", &b);
    c = a + b;
    printf("%d + %d = %d\n", a, b, c);
}

Ulaz:
Unesi prvi broj : 2 <enter>
Unesi drugi broj : 3 <enter>
Izlaz:
2 + 3 = 5
```

1.5.1 Aritmetičke i relacijske operacije

Primer 7 *Program vrši oduzimanje dva cela broja.*

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



```
main()
{
    /*deklaracija vise promenljivih istog tipa */
    int rez,pom1,pom2; /*rezultat oduzimanja pom1-pom2 -> rez*/
    pom1=20;
    pom2=15;
    rez=pom1-pom2;

    /*ispisivanje rezultata*/
    printf("Rezultat je %d-%d=%d\n",pom1,pom2,rez);
}
```

Izlaz iz programa:
Rezultat je 20-15=5

Primer 8 Program sabira dva uneta cela broja

```
#include <stdio.h>

int main()
{
    int a, b, c;
    printf("Unesi prvi broj : ");
    scanf("%d", &a);
    printf("Unesi drugi broj : ");
    scanf("%d", &b);
    c = a + b;
    printf("%d + %d = %d\n", a, b, c);
    return 0;
}
```

Ulaz:
Unesi prvi broj : 2 <enter>
Unesi drugi broj : 3 <enter>
Izlaz:
2 + 3 = 5

Primer 9 Program ilustruje neke od aritmetičkih operacija.

```
#include <stdio.h>
main()
{
    int a, b;
    printf("Unesi prvi broj : ");
    scanf("%d",&a);

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

```

    printf("Zbir a+b je : %d\n",a+b);
    printf("Razlika a-b je : %d\n",a-b);
    printf("Proizvod a*b je : %d\n",a*b);
    printf("Celobrojni kolicnik a/b je : %d\n", a/b);
    printf("Pogresan pokusaj racunanja realnog kolicnika a/b je : %f\n", a/b);
    printf("Realni kolicnik a/b je : %f\n", (float)a/(float)b);
    printf("Ostatak pri deljenju a/b je : %d\n", a/b);
}
Ulaz:
Unesi prvi broj : 2 <enter>
Unesi drugi broj : 3 <enter>
Izlaz:
Zbir a+b je : 5
Razlika a-b je : -1
Proizvod a*b je : 6
Celobrojni kolicnik a/b je : 0
Pogresan pokusaj racunanja realnog kolicnika a/b je : 0.000000
Realni kolicnik a/b je : 0.666667
Ostatak pri deljenju a/b je : 2

```

Primer 10 Program ilustruje celobrojno i realno deljenje.

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

```

main()
{
    int a = 5;
    int b = 2;
    int d = 5/2;    /* Celobrojno deljenje - rezultat je 2 */
    float c = a/b;  /* Iako je c float, vrsi se celobrojno deljenje jer su i a i b celi */

    /* Neocekivani rezultat 3.000000 */
    printf("c = %f\n",c);

    printf("Uzrok problema : 5/2 = %f\n", 5/2);

    printf("Popravljeno : 5.0/2.0 = %f\n", 5.0/2.0);

    printf("Moze i : 5/2.0 = %f i 5.0/2 = %f \n", 5/2.0, 5.0/2);

    printf("Za promenljive mora kastovanje : %f\n", (float)a/(float)b);
}

```

Izlaz iz programa:

```

c = 2.000000
Uzrok problema : 5/2 = 0.000000
Popravljeno : 5.0/2.0 = 2.500000
Moze i : 5/2.0 = 2.500000 i 5.0/2 = 2.500000
Za promenljive mora kastovanje : 2.500000

```

1.5.2 Operatori i izrazi dodeljivanja vrednosti

```
i = i + 2;  
ekvivalento je sa  
i+=2;
```

Moze i za:
+ - * / % << >> ^ |
izraz1 op = izraz2
je ekvivalentno sa
izraz1 = (izraz1) op (izraz2)

x*= y+1 je ekvivalento sa x = x * (y+1)

Takvo pisanje je krace i efikasnije.

1.5.3 Inkrementacija i dekrementacija

Operatori ++ i --

x=++n; se razlikuje od x=n++;

y=(x++)*(++z);

Primer 11 *Ilustracija prefiksnog i postfiksno 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, i rezultat  
       je uvecana vrednost */  
    x = ++a;  
  
    /* Postfiksni operator uvecava promenjivu, i rezultat je  
       stara (neuvecana) vrednost */  
    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

```

1.6 Relacioni i logički operatori

Relacioni operatori:

```

>   >=      <   <= isti prioritet
==  !=      nizi prioritet

```

```

(3<5)
(a<=10)
a < 5 != 1  <=> (a < 5)!=1

```

Logicki operatori:

```

! unarna negacija (najvisi prioritet)
&& logicko i (visi prioritet od ili)
|| logicko ili izracunavaju se sleva na desno!

```

```

5 && 4 vrednost je tacno
10 || 0 vrednost je tacno
0 && 5 vrednost je 0
!1 vrednost je 0
!9 vrednost je 0
!0 vrednost je 1
!(2>3) je 1
a>b && b>c || b>d je isto sto i ((a>b) && (b>c)) || (b>d)
koja je vrednost ako je a=10, b=5, c=1, d=15?

```

Primer 12 *Ilustracija logičkih vrednosti (0 - netačno, razlicito od 0 - tačno).*

```

#include <stdio.h>

main()
{
    int a;

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

```

```

        printf("Logicka vrednost broja je : tacno\n");
    else
        printf("Logicka vrednost broja je : netacno\n");
}

```

```

Ulaz:
Unesi ceo broj : 3 <enter>
Izlaz:
Logicka vrednost broja je : tacno

```

```

Ulaz:
Unesi ceo broj : 0 <enter>
Izlaz:
Logicka vrednost broja je : netacno

```

Primer 13 *Ilustracija ogičkih i relacijskih operatora.*

```

#include <stdio.h>

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

    printf("3>5 - %d\n5>3 - %d\n3==5 - %d\n3!=5 - %d\n", a, b, c, 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

```

1.7 Kontrola toka — if, while, do - while, for

1.7.1 if

```

if (izraz)
    iskaz1

```

```
else
    iskaz2
```

Primer 14 Program ilustruje if i ispisuje ukoliko je uneti ceo broj negativan.

```
#include <stdio.h>

int main()
{
    int b;
    printf("Unesi ceo broj:");
    scanf("%d", &b);
    if (b < 0)
        printf("Broj je negativan\n");
    return 0;
}
```

```
Ulaz:
Unesi ceo broj:-5
Izlaz:
Broj je negativan
```

```
Ulaz:
Unesi ceo broj:5
Izlaz:
```

Else se odnosi na prvi neuparen if, voditi o tome računa, ako želimo drugačije moramo da navedemo vitičaste zagrade.

```
if (izraz)
    if (izraz1) iskaz 1
else iskaz
```

ovo else se odnosi na drugo if a ne na prvo if!

```
if (izraz)
{
    if (izraz1) iskaz 1
}
else iskaz
```

tek sada se else odnosi na prvo if!!!

1.7.2 Else-if

```
if (izraz1)
    iskaz1
else if (izraz2)
    iskaz2
else if (izraz3)
    iskaz3
else if (izraz4)
```

```
    iskaz4
else iskaz

npr if (a<5)
    printf("A je manje od 5\n");
else if (a=5)
    printf("A je jednako 5\n");
else if (a>10)
    printf("A je vece od 10\n");
else if (a=10)
    printf("A je jednako 10\n");
else printf("A je vece od pet i manje od 10\n");
```

Primer 15 Program ilustruje if-else konstrukciju i ispituje znak broja.

```
#include <stdio.h>

int main()
{
    int b;
    printf("Unesi ceo broj : ");
    scanf("%d", &b);
    if (b < 0)
        printf("Broj je negativan\n");
    else if (b == 0)
        printf("Broj je nula\n");
    else
        printf("Broj je pozitivan\n");
    return 0;
}
```

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

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

Primer 16 Pogresan program sa dodelom = umesto poredjenja ==.

```
#include <stdio.h>

int main()
{
    int b;
    printf("Unesi ceo broj : ");
    scanf("%d", &b);
```

```

/* Obratiti paznju na = umesto == Analizirati rad programa*/
if (b = 0)
    printf("Broj je nula\n");
else if (b < 0)
    printf("Broj je negativan\n");
else
    printf("Broj je pozitivan\n");
return 0;
}

```

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

1.7.3 while

```
while(uslov) { ... }
```

Uslov u zagradi se testira i ako je ispunjen telo petlje se izvršava. Zatim se uslov ponovo testira i ako je ispunjen ponovo se izvršava telo petlje. I tako sve dok uslov ne bude ispunjen. Tada se izlazi iz petlje i nastavlja sa prvom sledecom naredbom u programu.

Ukoliko iza while sledi samo jedna naredba nema potrebe za zagradama.

```

while (i<j)
    i=2*i;

```

1.7.4 do-while

Ovo je slično paskalskom repeat-until izrazu.

```
do iskaz while (izraz)
```

Primer 17 Program ilustruje petlju - while.

```

#include <stdio.h>

int main()
{
    int x;

    x = 1;
    while (x<10)
    {
        printf("x = %d\n",x);
        x++; /* x++ je isto kao i x=x+1 */
    }
}

Izlaz:
x = 1

```



```
x = 2
x = 3
x = 4
x = 5
x = 6
x = 7
x = 8
x = 9
```

Primer 18 Program ilustruje petlju do-while.

```
#include <stdio.h>

int main()
{
    int x;

    x = 1;
    do
    {
        printf("x = %d\n",x);
        x++; /* x++ je isto kao i x=x+1 */
    }
    while (x<=10);
}
```

Izlaz:

```
x = 1
x = 2
x = 3
x = 4
x = 5
x = 6
x = 7
x = 8
x = 9
x = 10
```

Primer 19 Program ilustruje petlju - for.

```
#include <stdio.h>

int main()
{
    int 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
```

Primer 20 *Konverzija centimetara u inče - while petlja.*

```
#include <stdio.h>

/* Definicija simbolickih konstanti preko #define direktiva */
/* U fazi preprocesiranja se vrsi doslovna zamena konstanti
   njihovim vrednostima */

#define POCETAK 0
#define KRAJ 20
#define KORAK 10

int main()
{
    int a;
    a = POCETAK;
    while (a <= KRAJ)
    {
        printf("%d cm = %f in\n", a, a/2.54);
        a += KORAK; /* isto sto i a = a + KORAK; */
    }
    return 0;
}
```

```
Izlaz:
0 cm = 0.000000 in
10 cm = 3.937008 in
20 cm = 7.874016 in
```

Primer 21 *Konverzija centimetara u inče - for petlja.*

```
#include <stdio.h>
#define POCETAK 0
#define KRAJ 20
#define KORAK 10

int main()
{
    int a;
```

```

    for (a = POČETAK; a <= KRAJ; a += KORAK)
        printf("%d cm = %f in\n", a, a/2.54);

    return 0;
}

Izlaz:
0 cm = 0.000000 in
10 cm = 3.937008 in
20 cm = 7.874016 in

```

1.7.5 Switch

```

switch (iskaz) {
    case konstantan_izraz1: iskazi1
    case konstantan_izraz2: iskazi2
    ...
    default: iskazi
}

```

Primer 22 *Ilustracija switch konstrukcije.*

```

#include<stdio.h>
int main()
{
    int n;
    printf("Unesi paran broj manji od 10\n");
    scanf("%d",&n);
    switch(n)
    {
    case 0:
        printf("Uneli ste nulu\n");
        break;
    case 2:
        printf("Uneli ste dvojku\n");
        break;
    case 4:
        printf("Uneli ste cetvorku\n");
        break;
    case 6:
        printf("Uneli ste sesticu\n");
        break;
    case 8:
        printf("Uneli ste osmicu\n");
        break;
    default:
        printf("Uneli ste nesto sto nije paran broj\n");
    }
    return 0;
}

```

Ulaz:
Unesi paran broj manji od 10

```
2
Izlaz:
Uneli ste dvojku
```

1.8 Uslovni izraz

Slično kao if.

```
izraz1 ? izraz2 : izraz3
```

```
z = (a<b)? a : b; /*z=min(a,b)*/
max = (a>b)? a : b;
```

Zadaci za vežbu:

Zadatak 1 Šta će biti ispisano nakon izvršavanja sledećeg programa?

```
#include <stdio.h>
main()
{
    int x=506, y=3, z=21, t=2;
    printf("x=%d y=%d\n",x,y);
    printf("z - t=%d\n", z-t);
    printf("z / t =%d\n",z / t);
    printf("-x=%d\n",- x);
    printf("x %% y=%d\n", x%y);
}
```

Zadatak 2 Napisati program koji sabira dva cela broja sa ulaza.

Zadatak 3 Napisati program za razmenu vrednosti dva cela broja.

Zadatak 4 Izvršiti štampanje parnih brojeva od 1 do 100 (for, while i do-while).

Zadatak 5 Napisati program koji izračunava sumu i maksimum brojeva koji se unose na standardni ulaz pri čemu je poslednji uneti broj 0 (for, while).

Zadatak 6 Napisati program koji ispisuje kvadrate svih brojeva od 5 do 35. Nakon svakog petog kvadrata odštampati znak za novi red (for, while).