

Programiranje 1  
*Beleške sa vežbi*  
*Školska 2008/2009 godina*

Matematički fakultet, Beograd

Jelena Graovac (Tomašević)

November 15, 2008



# Sadržaj

<b>1</b>	<b>Programski jezik C</b>	<b>5</b>
1.1	Ugnježdena petlja . . . . .	5
1.2	Oblast važenja lokalnih promenljivih . . . . .	6
1.3	Znakovni ulaz i izlaz - getchar i putchar . . . . .	6



# 1

## Programski jezik C

1

### 1.1 Ugnježdene petlje

Primer 1 *Ilustracija dve ugnježdene petlje.*

```
#include<stdio.h>

int main() {
    int i,j;
    for(i=1; i<=3; i++)
        {
            for(j=1; j<=3; j++)
                printf("%d * %d = %d\t", i, j, i*j);
            printf("\n");
        }
}
```

Izlaz:

```
1 * 1 = 1    1 * 2 = 2    1 * 3 = 3
2 * 1 = 2    2 * 2 = 4    2 * 3 = 6
3 * 1 = 3    3 * 2 = 6    3 * 3 = 9
```

Primer 2 *Program koji ispisuje tablicu množenja*

```
#include<stdio.h>

main() {
    int n, m; /* Dimenzije tablice */
    int i, j; /* Brojaci */

    scanf("%d", &n);
    scanf("%d", &m);

    /* Petlja po redovima... */
    for(i = 0; i < n; i++) {
```

---

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

```

    /* unutrašnja petlja */
    for(j = 0; j < m; j++)
        printf("%d * %d = %d\t", i, j, i*j);
    /* na kraju prelazimo u sledeci red */
    printf("\n");
}
}

```

## 1.2 Oblast važenja lokalnih promenljivih

### Primer 3

```

#include <stdio.h>

main()
{
    int pom=1;
    printf("Pre ulaska u unutrašnji blok pom=%d\n",pom);
    {
        int pom=50;
        printf("Pre izlaska iz unutrašnjeg bloka pom=%d\n",pom);
    }
    printf("Nakon izlaska iz unutrašnjeg bloka pom=%d\n",pom);
}

```

Izlaz:

Pre ulaska u unutrašnji blok pom=1

Pre izlaska iz unutrašnjeg bloka pom=50

Nakon izlaska iz unutrašnjeg bloka pom=1

*Primer 4 Program koji ispisuje prvih n prostih brojeva*

## 1.3 Znakovni ulaz i izlaz - getchar i putchar

Standardna biblioteka obezbeđuje nekoliko funkcija za čitanje i pisanje po jednog znaka posebno, od kojih su `getchar` i `putchar` najprostije.

Primer korišćenja funkcije za čitanje jednog znaka sa ulaza je:

```
c = getchar();
```

Nakon izvršetka ove naredbe, promenljiva `c` će sadržati jedan znak sa standardnog ulaza.

Primer korišćenja funkcije za štampanje jednog znaka na izlazu je:

```
putchar(c);
```

Nakon izvršetka ove naredbe, štampa se karakter koji je sadržaj promenljive `c` na standardnom izlazu.

Konstanta `EOF` je celobrojna vrednost definisana u biblioteci `<stdio.h>`. Ovo je vrednost koju vraća funkcija `getchar()` kada nema više ni jednog znaka na ulazu, odnosno kada se stigne do kraja ulazne datoteke. Nazvana je `EOF` kao `End Of File`, engleski prevod za kraj datoteke. Ova vrednost mora da se razlikuje od svake vrednosti koja može da bude karakter. Zato promenljiva `c` za koju se poziva `c=getchar()` treba da bude tipa dovoljno velikog da može da prihvati sve što može da vrati poziv ove funkcije, dakle sve što je karakter i plus `EOF`. Zbog toga se za `c` koristi tip `int` a ne `char`.

**Napomena:** `EOF` se pod Linux-om označava karakterom `ctrl-d` a pod Windows-om u Dev-C++ sa `enter ctrl-z enter`.

**Primer 5** Program vrši demonstraciju poziva funkcija `putchar` i `getchar`.

```
#include <stdio.h>

main()
{
    int c1, c2;
    c1 = getchar();
    printf("-----\n");
    c2 = getchar();

    printf("c1 = %d, c2 = %d\n",c1, c2);
    printf("c1 = %c, c2 = %c\n",c1, c2);

    putchar(c1); /* isto je kao i printf("%c",c1); */
    putchar(c2); /* isto je kao i printf("%c",c2); */
    putchar('\n');

    /* Za ispisivanje karaktera a */
    putchar('a');
    /* dozvoljeno je : printf("abc"); printf("a"); */
    /* nedozvoljeno je : printf('a'); putchar('abc'); putchar("abc"); */
}
```

Ulaz: ab

Izlaz:

```
-----
c1 = 97, c2 = 98
c1 = a, c2 = b
ab
a
```

**Primer 6** Program čita jedan karakter sa standardnog ulaza i ispisuje ga na standardnom izlazu.

```
#include <stdio.h>

main()
{
    int c;          /* Karakter - obratiti paznju na int */
    c = getchar(); /* cita karakter sa standardnog ulaza */
    putchar(c);    /* pise karakter c na standardni izlaz */

    putchar('\n'); /* prelazak u novi red */
    putchar('a');  /* ispisuje malo a */
    putchar(97);   /* ekvivalentno prethodnom */
}
```

Ulaz: s

Izlaz iz programa:

```
s
aa
```

**Primer 7** Program vrši prebrojavanje cifara unetih na ulazu.

```
#include <stdio.h>

/* zbog isdigit */
#include <ctype.h>
main()
{
    int c;
    int br_cifara = 0;
    while ((c = getchar()) != EOF)
        if (c>='0' && c<='9') /* moze i if (isdigit(c)) */
            br_cifara++;

    printf("Broj cifara je : %d\n", br_cifara);
}
```

**Napomena:** Obratiti pažnju na zagrade u okviru while ( (znak=getchar() ) != EOF). Ako bi se izostavile onda bi se prvo ispitalo da li je getchar() != EOF pa bi se vrednost (1 ili 0) dodelila promenljivoj znak, što nije željeni efekat.

**Primer 8** Napisati program koji prepisuje ulaz na izlaz tako što mala slova pretvara u velika, velika u mala, a ostale karaktere prepisuje nepromenjene.

```
#include <stdio.h>

main()
{
    int c;
    while ((c = getchar()) != EOF)
        if (c>='a' && c<='z')
            putchar(c-'a'+'A');
        else if (c>='A' && c<='Z')
            putchar(c+'a'-'A');
        else putchar(c);
}
```

**Primer 9** Program broji linije i znakove na ulazu.

```
#include <stdio.h>
main()
{
    int znak; /* Prihvata znak sa ulaza */
    long br_linija=0 ; /* Brojac linija */
    long br_znakova=0; /* Brojac znakova na ulazu */

    while ( (znak=getchar() ) != EOF)
    {
        br_znakova++;
        if (znak=='\n') br_linija ++;
    }

    printf("Prelazaka u novi red: %ld, karakteri: %ld \n",br_linija,br_znakova);
}
```



**Primer 10** Program broji blankove, horizontalne tabulatore.

```
#include <stdio.h>

main()
{
    int znak;          /* Prihvata znak sa ulaza */
    int br_blankova=0; /* Brojac blankova */
    int br_tabulatora=0; /* Brojac horizontalnih tabulatora */

    /* UOCITI: blok naredbi while ciklusa NIJE OGRADJEN
    viticastim zagradaama jer postoji samo jedna if naredba! */
    while( (znak=getchar())!=EOF )
        if( znak==' ' ) ++br_blankova; /* brojimo blanko simbole */
        else if( znak=='\t' ) ++br_tabulatora; /* brojimo tab-ove */

    /* Izdavanje rezultata na standardnom izlazu */
    printf("Broj blankova je %d a tabulatora %d.\n", br_blankova, br_tabulatora);
}
```

**Primer 11** Napisati program koji prepisuje ulaz na izlaz čineći tabulatore, nove linije i backslash-ove vidljivim.

```
#include <stdio.h> main() {
    int znak;
    znak=getchar();
    while( znak!=EOF )
    {
        if( znak=='\t' ) /*uciniti tab vidljivim */
            { putchar('\t'); putchar('t'); }
        else if( znak=='\n' ) /*uciniti new line vidljiv */
            { putchar('\n'); putchar('n'); putchar('\n'); }
        else if( znak=='\'\'\' ) /*backslash udvojiti */
            { putchar('\t'); putchar('\t'); }
        else putchar(znak);

        znak=getchar();
    } /* while( znak!=EOF ) */

} /* main() */
```

**Primer 12** Šta će biti izlaz iz sledećeg programa?

```
#include <stdio.h>

main()
{
    printf("\nZdravo, svima\n\n");
    printf("\n\tprelazak u novi red\n");
    printf("\t\ttabulator\n");
    printf("\t\t\tkosa crta\n");
    printf("\t\t\t\tprocenat\n");
}
```

Izlaz iz programa:  
 "Zdravo, svima"  
 \n prelazak u novi red  
 \t tabulator  
 \\ kosa crta  
 %% procenat

### Primer 13 *A šta iz ovog?*

```
#include <stdio.h>

main()
{
    putchar('\\');
    putchar('t');
    putchar('\t');
    printf("Za %d ispisujem %c", '\\', '\\');
    printf("\n\n\n\\n\\n\\n\n\n");
}
```

Izlaz iz programa:  
 \t Za 92 ispisujem \

\n\  
 \\n

### Primer 14

```
#include <stdio.h>
main()
{
    int vrednost;
    vrednost='A';
    printf("%s\nkarakter=%3c\nvrednost=%3d\n",
           "Veliko slovo",vrednost,vrednost);
    vrednost='a';
    printf("%s\nkarakter=%3c\nvrednost=%3d\n",
           "Malo",vrednost,vrednost);
}
```

Izlaz (u slucaju ASCII):  
 Veliko slovo  
 karakter= A  
 vrednost= 65  
 Malo  
 karakter= a  
 vrednost= 97

### Primer 15 *Napisati program koji ispisuje ascii tabelu.*

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

```
main()
{
    int c;
    for (c = 0; c<128; c++)
        printf("%d - %c\n",c,c);
}
```

**Primer 16** *Napisati program za razmenu vrednosti dva cela broja.*

Prvi nacin:

```
#include <stdio.h>

main()
{
    int a = 10;
    int b = 15;
    int tmp;

    tmp = a;
    a = b;
    b = tmp;

    printf ("a=%d, b=%d\n", a, b);
}
```

Drugi nacin:

```
#include <stdio.h>

main()
{
    int a = 10;
    int b = 15;

    b = a+b; /* a = 10; b = 25; */
    a = b-a; /* a = 15; b = 25; */
    b = b-a; /* a = 15; b = 10; */

    printf ("a=%d, b=%d\n", a, b);
}
```

**Primer 17** *Napisati program koji menja mesta cifara u broju.*

```
#include <stdio.h>
main(){
    int n,t=0;
    printf("Unesite broj\n");
    scanf("%d",&n);
    while(n)
    { t=t*10+n%10;
      n/=10;
    }
}
```

```

    printf("Novi broj je %d\n", t);
    return 0;
}

```

Izlaz:

```

Unesite broj
1234
Novi broj je 4321

```

**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** Dat je fragment C programa:

```

i=1; j=1;
while (i+j<10)
{ ++j; i+=2;}
suma=i+j;

```

- Koliko puta će se ponoviti while ciklus?
- Koje su vrednosti promenljivih  $i$ ,  $j$ ,  $suma$  nakon izvršenja fragmenta?
- Napisati ekvivalentan for ciklus.

Rešenje:

- 3 puta
- $i=7$ ,  $j=4$ ,  $suma=11$
- for( $i=1, j=1$ ;  $i+j<10$ ;  $++j$ ,  $i+=2$ );  
 $suma=i+j$ ;

**Zadatak 3** Dat je fragment C programa:

```

i=1; j=1;
while (i+j<10)
    ++j; i+=2;
suma=i+j;

```

- Koliko puta će se ponoviti while ciklus?
- Koje su vrednosti promenljivih  $i$ ,  $j$ ,  $suma$  nakon izvršenja fragmenta?
- Napisati ekvivalentan for ciklus.

Rešenje:

- a) 8 puta
- b) i=3, j=9, suma=12
- c) 

```
for(i=1,j=1; i+j<10; ++j);  
    i+=2;  
    suma=i+j;
```

**Zadaci za praktikum:**

**Zadatak 4** *Uraditi zadatke sa kolokvijuma.*