

*Matematički fakultet*

# ***Programiranje 1 – programski jezik C***

*Beleške sa časova vežbi  
školska 2010/2011*

*Biljana Stojanović*

*čas 05, 02. 11. 2010.*



# Uvod u programski jezik C

---

## 1. Leksičke konvencije

U upotrebi su velika i mala slova, cifre i specijalni simboli iz ASCII skupa.

**Programski jezik C razlikuje velika i mala slova! (CASE SENSITIVE).**

Komentari se navode između sekvenci `/*` i `*/`. Mogu se prostirati u više linija. Ne mogu biti ugnježdjeni.

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

Komentari u jednoj liniji mogu se obeležavati i pomoću `//`.

## 2. Tokeni

Postoji šest vrsta tokena:

- identifikatori
- ključne reči
- operatori
- separatori
- stringovi i
- konstante.

Tokeni se razdvajaju belinama, tabulatorima i novim redovima.

### 2.1 Identifikatori

**Identifikatori** se sastoje iz slova, cifara i znaka `_` pri čemu prvi karakter nije cifra.

Koriste se za imena promenljivih, tipova, funkcija, itd.

### 2.2 Ključne reči

**Ključne reči** su rezervisane reči koje imaju posebnu ulogu, i ne mogu se koristiti kao identifikatori.

Ključne reči se koriste za:

- definisanje jezičkih konstrukcija (*if*, *while*, *for*),
- imena tipova (*int*, *float*, *char*), itd.

---

<sup>1</sup> Zasnovano na primerima kolega Milana Bankovića ([www.matf.bg.ac.rs/~milan](http://www.matf.bg.ac.rs/~milan)), Jelene Graovac ([www.matf.bg.ac.rs/~jgraovac](http://www.matf.bg.ac.rs/~jgraovac)) i Staše Vujičić ([www.matf.bg.ac.rs/~stasa](http://www.matf.bg.ac.rs/~stasa))



## 3. Osnovni tipovi podataka. Konstante. Promenljive. Deklaracija i inicijalizacija

### 3.1 Osnovni tipovi podataka

Osnovni tipovi podataka:

- `char` — mali ceo broj (ASCII kod karaktera), jedan bajt
- `int` — celobrojna vrednost, najčešće 4 bajta (može i 2 bajta, zavisi od računara)
- `float` — realan broj jednostruke tačnosti, najčešće 4 bajta
- `double` — realan broj dvostruke tačnosti, najčešće 8 bajtova

Osnovnim tipovima podataka mogu se pridružiti kvalifikatori **short** i **long**.

Tipu *int* mogu se pridružiti oba pri čemu navodjenje ključne reči *int* nije obavezno:

Veličine tipova su platformski zavisne.

Na svakom računaru važi sledeće:

$$\text{broj bitova}(\text{short}) \leq \text{broj bitova}(\text{int}) \leq \text{broj bitova}(\text{long})$$

Tipu *double* može se pridružiti **long**, dok se tipu *float* ne može pridružiti ni jedan kvalifikator.

Postoje i kvalifikatori **signed** i **unsigned** koji se mogu pridružiti celobrojnim tipovima (`int`, `char`, `short` i `long`).

Neoznačeni brojevi (`unsigned`) su uvek pozitivni ili nula.

*Primer:*

Promenljiva tipa *signed char* uzima vrednosti od  $-128$  do  $127$ .

Promenljiva tipa *unsigned char* uzima vrednosti od  $0$  do  $255$ .

### 3.2 Konstante

Razlikujemo:

- **Karacterske konstante** — to su konstante tipa **char**. Navode se unutar jednostrukih navodnika. Vrednost karacterske konstante je numerička vrednost tog znaka u skupu znakova računara. Izvesni znakovi se mogu predstaviti pomoću specijalnih sekvenci kao što je `\n` (novi red).
- **Celobrojne konstante** — konstante tipa **int**. Mogu biti dekadne, oktalne i heksadekadne. **Oktalne** počinju nulom, a **heksadekadne** sekvencom **0x** ili **0X**. Mogu imati sufikse **u** ili **U** koji ukazuju da se radi o neoznačenoj celobrojnoj konstanti. Takodje, mogu imati i sufikse **I** ili **L** koji ukazuju da se radi o konstanti tipa **long**. Sufiksi **ul** i **UL** ukazuju na tip **unsigned long**.
- **Realne konstante** — konstante tipa **double** i mogu biti sa ili bez eksponencijalnog dela. Mogu imati sufiks **f** ili **F** koji ukazuje da se radi o konstanti tipa **float**. Realne konstante sa sufiksom **I** ili **L** su tipa **long double**.

### 3.3 Promenljive. Deklaracija i inicijalizacija.

Sve promenljive u C-u se moraju deklarirati. Time se za promenljivu u memoriji rezerviše potreban prostor, a ostatak programa postaje svestan postojanja promenljive i njenog tipa.

Deklaracija se sastoji iz **imena tipa** za kojim slede **imena promenljivih** (identifikatori) koje se deklariraju, i koja su razdvojena zarezima.

Deklaracija se završava simbolom ';'.

Svako ime promenljive u deklaraciji može biti praćeno inicijalizatorom koji se sastoji iz karaktera '=' za kojim sledi **inicijalna vrednost**.

***Da bi se promenljiva mogla upotrebljavati u programu 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 */
```

Postoji i kvalifikator **const** koji može biti dodeljen deklaraciji bilo koje promenljive da bi označio da se ona neće menjati

```
const double e=2.71828182845905;
```

## 4. Program u C-u

Izvorni program (source code) u C-u je običan tekstualni fajl, kreiran u bilo kom editoru teksta (npr. Notepad).

Svaki C program sastoji se od funkcija i promenljivih.

Funkcija sadrži naredbe koje određuju računске operacije koje treba da se obave, a promenljive čuvaju vrednosti koje se koriste tokom izračunavanja.

C funkcije su slične potprogramima i funkcijama u *Fortranu* ili procedurama i funkcijama u *Pascalu*.

Jedan od metoda za razmenu podataka između funkcija je da pozivajuća funkcija obezbedi listu vrednosti, tzv. argumenata, za funkciju koju poziva.

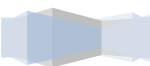
Zagrade iza imena funkcije okružuju listu argumenata. Ako funkcija ne očekuje nijedan argument, to se označava praznom listom ().

Naredbe od kojih se funkcija sastoji čine **telo funkcije**. Naredbe u C-u su razdvojene znakom ';', koji ima ulogu separatora naredbi.

Telo funkcije se navodi iza zaglavlja funkcije, unutar velikih (vitičastih) zagrada { }.

Funkcija koja mora da postoji u svakom programu je **main()**. Izvršavanje programa se svodi na izvršavanje tela ove funkcije.

Telo funkcije se navodi iza zaglavlja funkcije `main()`, između vitičastih zagrada { }.



U telu funkcije se na početku navode deklaracije pomenljivih, nakon čega sledi proizvoljan niz naredbi .

```
int main() /* zaglavlje funkcije main */
{
    /* ovde pocinje telo funkcije */
    /* deklaracije promenljivih */
    /* naredbe */
}
/* kraj tela funkcije */
```

## 4.1 Prvi primeri u C-u

1. Napisati program koji na standardni izlaz ispisuje poruku "Zdravo, svete!".

```
#include <stdio.h>
int main()
{
    printf("Zdravo, svete!\n");
    return 0;
}
```

Izlaz iz programa:

Zdravo, svete!

### Objašnjenje:

```
#include <stdio.h>
/* kazuje prevodiocu da uključi informacije o standardnoj ulazno/izlaznoj biblioteci; ovaj red se nalazi na početku mnogih izvornih datoteka u C-u. */
```

U primeru, funkcija *main* poziva funkciju *printf*. Funkcija se poziva navođenjem njenog imena i iza toga liste argumenata u zagradama.

Sekvenca znakova između dvostrukih navodnika, kao što je „Zdravo, svete!\n“, naziva se znakovni string ili string konstanta.

Sekvenca \n u stringu je C notacija za znak za *novi red*, koji, kada se prikaže, proizvodi prelazak u levu marginu novog reda. Ako se izostavi, vidi se da ne dolazi do prelaska u novi red nakon što se prikaže izlaz, što se može videti na narednom primeru.

**Napomena!** \n predstavlja jedan znak. Specijalna sekvenca, kao što je \n, obezbeđuje opšti i proširivi mehanizam za predstavljanje nevidljivih znakova ili onih kojih nema na tastaturi.

2. Šta je izlaz iz sledećeg programa?

```
#include <stdio.h>
int main()
{
    printf("Zdravo, ");
    printf("svete!");
    printf("\n");
    return 0;
}
```



### 3. Uvođenje promenljivih u program.

```
#include <stdio.h>
int 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);
    return 0;
}
```

Izlaz iz programa:  
Rezultat je 20 - 15 = 5

## 5. Funkcije ulaza i izlaza

Ulaz i izlaz ostvaruju se posredstvom funkcija koje su definisane u standardnoj biblioteci **stdio.h**. Ove funkcije su obične C funkcije, koje se služe direktno servisima operativnog sistema prilikom svog rada.

Za korišćenje ovih funkcija neophodno je uključiti zaglavlje **stdio.h** navođenjem direktive **#include<stdio.h>** pre definicije funkcije main().

Ovo zaglavlje je obilan tekstualni fajl u kome su navedene deklaracije funkcija ulaza i izlaza.

Direktiva **#include** na mestu poziva uključuje kompletan sadržaj fajla koji je naveden, čime funkcije i podaci deklarirani u njemu postaju dostupni funkciji main().

### 5.1 Funkcija printf()

Funkcija **printf** je funkcija standardne biblioteke koja prikazuje izlazne podatke u određenom formatu.

Ovom funkcijom se ispisuje poruka zadata *format-stringom* na standardni izlaz.

Primer korišćenja funkcije printf je:

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

Prvi argument ove funkcije je uvek između " " i određuje format u kome će se podaci ispisati na izlaz. Ova funkcija vraća kao vrednost broj ispisanih znakova na izlazu.

Sekvenca **\n** u okviru prvog argumenta funkcije *printf* je C oznaka za prelazak u novi red, **\t** je oznaka za tabulator, dok **%d** označava da će na tom mestu biti ispisana celobrojna vrednost argumenta koji je sa njim u paru. Svaka % konstrukcija (specifikacija konverzije) je u paru sa odgovarajućim argumentom koji sledi.



Eventualni specifikatori konverzija se zamenjuju vrednostima izraza koji u tom slučaju slede nakon format stringa, kao argumenti funkcije printf(), razdvojeni zarezima i u onom poretku u kome su odgovarajući specifikatori konverzija navedeni.

Tipovi izraza moraju biti u skladu sa tipovima koje određuju konverzioni specifikatori.

## Specifikatori konverzija

specifikator	tip	napomena
%d	int	dekadni ispis
%f	float, double	ispis bez eksponenta
%Lf	long double	ispis bez eksponenta
%hd	short	dekadni ispis
%ld	long	dekadni ispis
%c	char	ispis karaktera

%% koristi se za ispis znaka %

\\ koristi se za ispis znaka \

\" koristi se za ispis znaka ”

Postoji mogućnost da se precizira i širina polja u kome će se ispisati odgovarajuće vrednosti.

Na primer, koristimo **%3c** za ispis karaktera na tri pozicije poravnato s desna.

Koristimo **%3d** za ispis broja na tri pozicije ili **%6d** za ispis broja na 6 pozicija, poravnato s desna. *Desno poravnanje* je podrazumevano poravnanje.

### Primeri:

**%f** — ispisi kao realan broj

**%6f** — ispisi kao realan broj u polju širine najviše 6 znakova

**%.2f** — ispisi kao realan broj sa dve decimale

**%6.2f** — ispisi kao realan broj u polju širine najviše 6 znakova, pri čemu su 2 iza decimalne tačke.

Da bi se izvršilo *levo poravnanje*, između % i odgovarajućeg karaktera dodaje se znak -.

### Primer:

```
#include <stdio.h>
int main()
{
    printf("Slova:\n%3c\n%5c\n", 'z' , 'Z');
    return 0;
}
```

Izlaz iz programa:

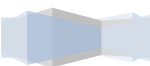
Slova:

z

Z

## 5.2 Funkcija scanf()

Ovom funkcijom se učitavaju podaci sa standardnog ulaza.



Prvi argument je format string u kome se navode specifikatori konverzija kojima se definiše tip podatka koji se očekuje.

Nakon format stringa slede adrese promenljivih, razdvojene zarezima, u koje treba upisati vrednosti učitane sa ulaza. Adresa promenljive **a** navodi se u obliku **&a**.

Adrese se navode u onom poretku u kom su odgovarajući konverzioni specifikatori navedeni u format stringu.

Tipovi promenljivih moraju biti u skladu sa tipovima koje određuju konverzioni specifikatori.

## Specifikatori konverzija

specifikator	tip	napomena
%d	int	opciono označeni dekadni broj
%f	float	realan broj sa opcionim eksponentom
%lf	double	realan broj sa opcionim eksponentom
%Lf	long double	realan broj sa opcionim eksponentom
%hd	short	opciono označeni dekadni broj
%ld	long	opciono označeni dekadni broj
%c	char	karakter iz ASCII skupa

### Primer 1:

```
scanf("%d %d", &broj1, &broj2);
```

Ova funkcija čita sa ulaza dva cela broja i smešta ih na adrese promenljivih *broj1* i *broj2*, redom. Kao rezultat, ova funkcija vraća broj uspešno dodeljenih ulaznih vrednosti.

Naredni poziv funkcije *scanf* nastavlja čitanje neposredno iza poslednjeg znaka koji je već pročitano.

### Primer 2:

Program prikazuje unos celog broja koristeći funkciju *scanf("%d", &x)*.

```
#include <stdio.h>
int main()
{
    int x;
    printf("Unesi ceo broj : ");
    /* Obratiti paznju na znak &
       pre imena promenljive u funkciji scanf. */
    scanf("%d",&x);
    /* U funkciji printf nije
       potrebno stavljati &. */
    printf("Uneli ste broj %d\n", x);
    return 0;
}
```

