

# Programski jezik C – I deo

## 1. Leksičke konvencije

Programski jezik C razlikuje velika i mala slova!

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

## 2. Osnovni znakovi jezika

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

- slova engleske abecede **A, B, C, … , X, Y, Z, a, b, c, … , x, y, z**
- cifre **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
- specijalni znaci: **+ = \_ ( ) \* & % # ! | . ; : ' / ? { } ~ \ [ ] ^**
- neštampajući znaci space (blanko), znak za novu liniju, …

## 3. Komentari

Služe za objašnjenja u toku pisanja programa. Pomocu komentara program postaje citljiviji za čoveka, prevodiocu (kompajleru) je potpuno sve jedno da li postoji sto hiljada linija komentara ili ni jedna jedina. Komentar počinje znakovima **/\***, a završava se znakovima **\*/**. Komentari u jednoj liniji mogu se obeležavati i pomoću **//**.

pr: **/\* ovo je KOMENTAR u jednom ili u više redova \*/**

Ako je za komentar dovoljan jedan red teksta možemo koristiti znakove **//**

pr: **// ovo je komentar**

Mogu se prostirati u više linija. Ne mogu biti ugnježdeni.

## 4. Tokeni

Postoji šest vrsta tokena:

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

Tokeni se razdvajaju blanko znakom, tabulatorima i novim redovima.

## **4.1. Identifikatori (imena)**

Koriste se za označavanje osnovnih objekata jezika: konstanti, promenjivih, funkcija i tipova podataka. Imena čine reči koje se sastoje od: slova, cifara i specijalnog znaka \_ tako da: prvi znak ne može biti cifra i ime može imati najviše 31 znak.

Primeri ispravnih imena:

**skola, y, x, a, zbir, Programska\_jezik\_C, INT**

Nedozvoljeno je:

**3dan, int, x+y, programski jezik, ...**

## **4.2. Ključne reči**

Ključne reči ili službene reči (ima ih 32) su rezervisane reči koje imaju posebnu ulogu. Ključne reči se koriste za:

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

**Ne** mogu se koristiti kao identifikatori.

To su sledeće reči:

auto, break, case, char, const, continue,  
default, do, double, else, enum, extern,  
float, for, goto, if, int, long, register,  
return, short, signed, sizeof, static,  
struct, switch, typedef, union, unsigned,  
void, volatile, while

## **4.3. Operatori**

U C-u postoji veliki broj operatora. Mogu biti

- unarni (imaju jedan argument) i
- binarni (dva argumenta).

Unarni operatori mogu biti prefiksni i sufiksni. Binarni operatori su uvek infiksni. Operatori imaju svoj prioritet i asocijativnost.

## Aritmetički operatori

- negativni predznak
- + operator sabiranja
- operator oduzimanja
- \*
- / operator deljenja
- % operator ostatak pri celobrojnom deljenju

Ovi operatori se primenjuju na celobrojne i realne tipove. Operator / primenjen na cele brojeve daje celobrojni količnik. Operator % se primenjuje samo na cele brojeve i daje ostatak pri deljenju. Aritmetički operatori Rezultat aritmetičkih operacija je izraz istog tipa kao i operandi. Ako operandi nisu istog tipa tada se vrši implicitna konverzija užeg u širi tip. Ako množimo dva broja tipa double, i rezultat smeštamo u proenljivu tipa int, tada će se tu izvršiti implicitna konverzija šireg u uži tip, pa će nam kompjuler prijaviti upozorenje o mogućem gubitku informacija.

## Relacioni operatori

- == operator jednakosti
- != operator nejednako
- < operator manje
- <= operator manje - jednako
- > operator veće
- >= operator veće - jednako

Ovi operatori se primenjuju na celobrojne i realne tipove. Rezultat relacionih operacija je izraz celobrojnog tipa sa vrednošću 1 ako je relacija tačna, 0 u suprotnom. Ako operandi nisu istog tipa tada se vrši implicitna konverzija užeg u širi tip.

## Logički operatori

- ! – unarna negacija
- && - logičko i
- || - logičko ili

U C-u ne postoji logički tip. Svaki izraz koji se može poređiti na jednakost sa nulom se u C-u smatra logičkim izrazom.

- Ako je vrednost izraza različita od 0, tada se on smatra logički tačnim. Izraz jednak nuli se smatra logički netačnim.
- Rezultat logičkih operacija je izraz celobrojnog tipa, jednak 0 ako je rezultat logičke operacije netačno, odnosno 1, ako je rezultat logičke operacije tačno. Logički operatori –

Primeri:

- `5 && 4` - vrednost je 1, tj. tačno
- `0 || 10` - vrednost je 1, tj. tačno
- `0 && 5` - vrednost je 0, tj. netačno
- `!1` - vrednost je 0, tj. netačno
- `!9` - vrednost je 0, tj. netačno
- `!0` - vrednost je 1, tj. tačno
- `!(2>3)` - vrednost je 1. tj. tačno

## Operatori sa bočnim efektima

- Pojava da se prilikom izračunavanja nekog izraza menja vrednost neke promenljive naziva se bočni efekat (eng. side effect).
- Operatori koji imaju bočni efekat su operatori dodele i operatori uvećanja i umanjenja.
- Naredba dodele se u C-u predstavlja izrazom dodele, a njeno izvršavanje se zasniva na bočnim efektima operatora dodele.

## Operatori dodele

### • Operator proste dodele je      $=$

Ne treba ga mešati sa relacionim operatorom `==`.

- Levi operand ovog operatora je leva vrednost (ime promenljive).
- Desni operand je proizvoljni izraz.

- Najpre se izračunava izraz na desnoj strani. Njegova vrednost se zatim po potrebi konvertuje u tip promenljive na levoj strani. Nakon toga se dobijena vrednost upisuje u memorijsku lokaciju koja je rezervisana za čuvanje te promenljive.
- Tip izraza dodele je tip promenljive na levoj strani. Vrednost izraza dodele je vrednost koja je dodeljena promenljivoj.

### • Operatori složene dodele su:      $+=, -=, *=, /=, \%=$

- Izraz oblika  $a+=E$  je ekvivalentan izrazu  $a=a+(E)$ .
- Slično je sa ostalim operatorima.

## Operatori umanjenja i uvećanja

- Unarni operatori  **$++$**  i  **$--$**  nazivaju se operatori uvećanja (inkrementacije) i umanjenja (dekrementacije) respektivno.
- Mogu biti prefiksni i sufiksni.

- U oba slučaja vrši se uvećanje promenljive za 1 ali izraz `++n` uvećava promenljivu `n` pre nego što se njena vrednost koristi, dok `n++` uvećava `n` nakon što se njena vrednost koristi.

Tako se `x=++n;` razlikuje od `x=n++;`.

- Slično za operator `--`, s tim što je u pitanju umanjenje za jedan.

## Prioritet i asocijativnost operatora

- Unarni operatori imaju viši prioritet od binarnih.
- Aritmetički operatori su višeg prioriteta od relacionih, a ovi višeg od logičkih.
- Operatori dodele su najnižeg prioriteta.
- Ako dva operatora imaju isti prioritet, onda se u obzir uzima asocijativnost, koja može biti s leva na desno, ili s desna na levo.
- Prioritet operatora se može promeniti korišćenjem zagrade `()`.
- Binarni operatori su leve asocijativnosti, osim operatora dodele koji imaju desnou asocijativnost. Unarni operatori su mahom desne asocijativnosti, ali ima izuzetaka.

### 4.4. Separatori

U jeziku C kraj svake naredbe označava se znakom – separatorom ; (tačka-zarez).

### 4.6. Konstante

Konstanta je objekat u programu koji ima ime i koji dobija vrednost jednom, pre početka izvršavanja programa i ne može se menjati. Konstanta se definiše rezervisanim rečima **define** pre početka glavne funkcije programa (pre `void main()`).

Primer: `#define PI 3.14;`

## 5. Promenljive

Promenljive u programskom jeziku su objekti koji imaju neku vrednost, pri čemu se vrednost može menjati u toku izvršavanja programa. Pri tome to ne može biti bilo koja vrednost, već samo vrednost jednog tipa.

Svaka promenljiva koja se koristi mora biti **deklarisana**. Deklaracijom se određuju:

- ime promenljive (to je ime prema sintaksi jezika C),
- tip promenljive (odnosno tip kome pripada vrednost date promenljive) i
- početna vrednost (vrednost u početnom trenutku) – neobavezno.

**Deklaracija promenljive** ima sledeći zapis:

```
ime_tipa ime_promenjive [=vrednost];
```

gde je *ime\_tipa* tip promenljive,

*ime\_promenjive* ime promenljive, a *vrednost* (koja može ali ne mora biti navedena) početna vrednost date promenljive.

Na primer, deklaracijama:

```
int i, j, n=10;    float a, e=2.71, pi=3.14;           char c, d='?';
```

su deklarisane:

- promenljive celobrojnog tipa *i, j i n*;
- promenljive realnog tipa *a, e i pi*;
- promenljive znakovnog tipa *c i d*.

Promenljive *n, e, pi i d* imaju, redom, vrednosti 10, 2.71, 3.14 i '?'.

## Pravila za dodeljivanja imena promenljivih

Promenljiva može sadrzati proizvoljno ime ali ono treba da opisuje namenu promenljive, dakle ako treba da cuva zbir nekih brojeva verovatno ce te je nazvati „zbir“ ili „rezultat“. Imena promenljivih smeju da sadrze slova, brojeve i znak donje crte( \_ ). Na pocetku imena se mora naci ili slovo ili donja crta, broj ne sme!

## 6. Tipovi promenjivih

### 1. Celobrojni tip – int

Ovaj tip je najkorišćeniji tip podatka. Promenljive koje su tipa int ( skracenica od integer) u 32bitnom sistemu zauzimaju 4 bajta i u takvim promenljivim mozete cuvati decimalne cele brojeve kao sto su 0, 789, 4000, -1, -4569 i sl. Primer deklaracije ovakve promenljive :

```
int broj1;
```

```
int broj2 = 150;
```

## 2. Realni tip – **float**

Nedostatak tipa int je u tome sto ne moze cuvati vrednosti koje imaju decimalnu tacku odnosno „zarez“ (dakle realne brojeve) vec INT sve cifre „iza zareza“ odbacuje. Da bi cuvala ovakve brojeve promenljiva mora biti tipa float ili, Primer deklaracije ovakvih promenljivih:

```
float x;
```

```
double y;
```

## 3. Realni tip – **double**

Ako je realni broj enormno veliki, koristi se tip double.

### **short i long**

Ove kljucne reci mogu se primeniti samo na do sada opisane tipove podataka. Ako deklarisemo promenljivu X kao „short int“ ona ce sadrzati (u 32bit-nom) sistemu 2 bajta, dakle duplo manje od regularnog int-a. Medjutim ona moze sadrzati brojeve ciji su opstezi takodje duplo manji. Kljucna rec long omogucava da promenljiva ima veci opseg od onog koji bi ta promenljiva imala bez ove kljucne reci.

## 4. Znakovni tip – **char**

Gore pomenuti tipovi su cuvali brojeve a promenljive koje su tipa char cuvaju simbole. Svaka promenljiva tipa char zauzima svega jedan bajt i moze da cuva samo jedan simbol. Simbola ima ukupno 255 i tu spadaju kompletan engleska abeceda, brojevi i razni drugi simboli i znakovi. O ovom tipu podatka cemo detaljnije pricati kasnije.

## 5. Neodredjeni tip - **void**

Ovaj tip se koristi ako se ne može unapred utvrditi tip podataka.

**Tablica: Tipovi podataka i intervali vrednosti**

deklaracija	Borland C++ 5.0	interval vrednosti
<b>char</b>	8 bita	[-128, 127]
<b>unsigned char</b>	8 bita	[0 , 255]
<b>short</b>	16 bita	[-32.768 , 32.767]
<b>int</b>	32 bita	[-2.147.483.648 , 2.147.483.647]
<b>long</b>	32 bita	[-2.147.483.648 , 2.147.483.647]
<b>unsigned</b>	32 bita	[0 , 4.294.967.295]
<b>unsigned long</b>	32 bita	[0 , 4.294.967.295]
<b>float</b>	32 bita	( $3.4 \times 10^{-38}$ , $3.4 \times 10^{38}$ )
<b>double</b>	64 bita	( $1.7 \times 10^{-308}$ , $1.7 \times 10^{308}$ )
<b>long double</b>	80 bita	( $3.4 \times 10^{-4932}$ , $3.4 \times 10^{4932}$ )