

Programiranje 1
Beleške sa vežbi
Školska 2008/2009 godina

Matematički fakultet, Beograd

Jelena Graovac (Tomašević)

November 13, 2008

Sadržaj

1 Programski jezik C	5
1.1 Komentari	5
1.2 Identifikatori i ključne reči	5
1.3 Tipovi i veličina podataka	5
1.4 Promenljive i konstante	6
1.5 Deklaracije	7
1.6 Funkcije printf i scanf	7
1.7 Prvi primeri u C-u	8

1

Programski jezik C

1

Program se sastoji iz niza naredbi kojim se definišu akcije koje izvršava računar. Postupak pisanja programa se naziva programiranje.

1.1 Komentari

U cilju poboljšanja smisla pojedinih delova programa može se koristiti komentar. On počinje parom znakova `/*` i završava se parom znakova `*/`. Na primer,

```
/* Ovo je jedan komentar u programskom jeziku C.*/
```

Komentari se mogu prostirati u više linija i ne mogu biti ugnježdjeni. Komentari u jednoj liniji se obeležavaju sa `//`. Na primer,

```
//I ovo je jedan komentar.
```

1.2 Identifikatori i ključne reči

Imena ili identifikatori se koriste za označavanje osnovnih objekata jezika: konstanti, promenljivih, funkcija i tipova podataka. Mogu se sastojati iz slova i cifara pri čemu ne sme počinjati cifrom. Potcrta `_` se smatra slovom (uglavnom se koristi kod dužih identifikatora). Velika i mala slova se u C-u razlikuju.

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

Ključne reči su osnovni elementi jezika koji imaju fiksno značenje kao na primer `if`, `else`, `for`, `while` ili imena tipova kao što su `int`, `double`, `float`, `char` i tako dalje.

Ključne reči se NE mogu koristiti kao identifikatori.

Na primer, dozvoljeni identifikatori su:

```
skola, x, y, moja_promenljiva
```

a nedozvoljeni su:

```
int (to je ključna reč), 5nedelja (počinje cifrom), x-y (sadrži znak -), for (ključna reč).
```

1.3 Tipovi i veličina podataka

Tipom se zadaje oblast definisanosti promenljivih, funkcija i izraza koji mu pripadaju, skup operacija koje se mogu izvršiti nad njima i način registrovanja u računaru.

Tipovi podataka mogu biti osnovni i složeni.

¹Zasnovano na primerima sa sajtova <http://www.matf.bg.ac.yu/~milan>, <http://www.matf.bg.ac.yu/~filip> i knjige Milana Čabarkape "C - osnovi programiranja".

Osnovni tipovi podataka su:

int — celobrojna vrednost, najčešće 4 bajta (može i 2 bajta — zavisi od računara),

char — jedan znak (celobrojna vrednost), jedan bajt,

float — realan broj jednostruke tačnosti, najčešće 4 bajta,

double — realan broj dvostruke tačnosti, najčešće 8 bajtova.

Postoje modifikatori koje možemo pridružiti osnovnim tipovima a to su **short** i **long**.

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

```
short int kratak_broj; /*zauzima 2 bajta*/
long int dugacak_broj; /*zauzima 4 bajta*/
short kratak;
long dugacak;
```

Na svakom računaru važi:

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

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

Postoje još i modifikatori **signed** i **unsigned** koji se mogu pridružiti celobrojnim tipovima (**int**, **char**, **short** i **long**). Oni se odnose na označene i neoznačene cele brojeve (nenegativne cele brojeve).

Ako je promenljiva celobrojnog tipa koji zauzima n bajtova, odnosno $8 \cdot n$ bitova, to znači da vrednosti koje mogu biti smeštene u toj promenljivoj pripadaju opsegu od $-2^{8 \cdot n - 1}$ do $2^{8 \cdot n - 1} - 1$. Ako su tipa **unsigned** onda pripadaju opsegu od 0 do $2^{8 \cdot n} - 1$.

Npr. promenljiva tipa **signed char** uzima vrednosti od -128 do 127 (odnosno od -2^7 do $2^7 - 1$) dok promenljiva tipa **unsigned char** uzima vrednosti od 0 do 255 (odnosno od 0 do $2^8 - 1$).

Napomena: Važniji stepeni broja 2: $2^7 = 128$, $2^8 = 256$, $2^{15} = 32768$, $2^{16} = 65536$, $2^{31} = 2\,147\,483\,648$ i $2^{32} = 4\,294\,967\,296$.

Na primer, optimalni tip za skup vrednosti $S = \{3, 5, 189, \dots, 15700\}$ je **short** ili **unsigned short** zato što sve vrednosti iz skupa pripadaju opsegu od -2^{15} do $2^{15} - 1$ kao i opsegu od 0 do $2^{16} - 1$.

Primer složenih tipova podataka su nizovi, struktura, unija i drugo.

1.4 Promenljive i konstante

Promenljiva je objekt jezika koji ima ime i kome se mogu dodeljivati različite vrednosti u toku izvršavanja programa. Konstante dobijaju vrednost pre početka izvršavanja programa i ne mogu se menjati u toku rada programa.

Razlikujemo:

- *Karacterske konstante* — To su konstante tipa **char**. Navode se između jednostrukih navodnika, na primer 'A', 'a';
- *Celobrojne konstante* — To su 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 nagoveštavaju da se radi o neoznačenoj konstanti. Takođe mogu imati i sufikse l ili L koji nagoveštavaju da se radi o konstanti tipa **long**. Na primer, dekadne konstante su 12, 111, oktalne su 012 (ima dekadnu vrednost 10), 0111 (dekadno 73) a heksadekadne su 0x12 (što je dekadno 18), 0X2f (dekadno 47) i tako dalje;
- *Realne konstante* — To su konstante tipa **double** i mogu biti sa ili bez eksponencijalnog dela. Znak za eksponent je e ili E, on u zapisu broja izražava decimalni poredak i čita se "pomnožiti sa 10 na". Na primer, 3.26E6 u C-u je $3.26 \cdot 10^6$ u matematici. Mogu imati sufiks f ili F i on označava da se radi o konstanti tipa **float**. Realne konstante sa sufiksom l ili L su tipa **long double**.

1.5 Deklaracije

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
```

1.6 Funkcije printf i scanf

Funkcija `printf` je biblioteka funkcija koja prikazuje izlazne podatke u određenom formatu. Primer korišćenja funkcije `printf` je:

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

Prvi argument ove funkcije je uvek između duplih navodnika (`"`) i određuje format u kome će se podaci ispisati na izlaz. Ova funkcija vraća kao vrednosti broj upisanih 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 je u paru sa odgovarajućim argumentom koji sledi.

Da bi se ispisala vrednost promenljive koja je tipa `char`, u okviru prvog argumenta funkcije `printf` na odgovarajućem mestu se navodi sekvenca `%c`.

`%o` se koristi za ispis oktalnog broja,

`%x` za ispis heksadekadnog broja,

`%u` za ispis neoznačenog dekadnog broja,

`%f`, `%e` ili `%g` koristi se za ispis realnog broja (zavisi od toga na koji način želimo da ispišemo realan broj) i na kraju:

`%%` 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 štampanje karaktera na tri pozicije poravnato sdesna. Isto tako, koristimo `%3d` za štampanje broja na tri pozicije ili `%6d` za štampanje broja na 6 pozicija. Isto tako važi sledeće:

`%f` — štampanje kao realan broj

`%6f` — štampanje kao realan broj širok najviše 6 znakova

`%.2f` — štampanje kao realan broj sa dve decimalne

`%6.2f` — štampanje kao realan broj širok najviše 6 znakova pri čemu su 2 iza decimalne tačke.

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

Formatizovani ulaz se ostvaruje pomoću funkcije `scanf` i ona se poziva slično kao funkcija `printf`. Na primer, ako su deklarirane celobrojne promenljive `broj1` i `broj2` i `broj3` i ako se na ulazu nalazi informacija:

```
2 24 345
```

tada pozivom funkcije `scanf`:

```
scanf("%d %d %d", &broj1, &broj2, &broj3);
```

promenljivim `broj1`, `broj2` i `broj3` se dodeljuju redom vrednosti 1, 24 i 345. Karakterom `&` se označava adresa ovih promenljivih (biće o tome reči kasnije).

Za razdvajanje ulaznih polja funkcija `scanf` koristi specijalne znake: prazninu, tabulator i novu liniju.

Ako se na ulazu unosi informacija:

```
1<tab>24<enter>
345<enter>
```

tada pozivom funkcije `scanf`:

```
scanf("%d %d %d", &broj1, &broj2, &broj3);
```

promenljivim `broj1`, `broj2` i `broj3` takođe dodeljuju redom vrednosti 1, 24 i 345.

Kao rezultat, ova funkcija vraća broj uspešno dodeljenih ulaznih vrednosti. Naredni poziv funkcije `scanf` nastavlja čitanje neposredno iza poslednjeg znaka koji je već pročitan.

1.7 Prvi primeri u C-u

Primer 1 *Napisati program koji na standardnom izlazu štampa poruku "Zdravo, svete!".*

```
#include <stdio.h>

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

Izlaz iz programa:
Zdravo, svete!

Izvršavanje svakog programa započinje funkcijom `main`. Programi mogu da koriste funkcije iz sistemske biblioteke i u tom slučaju u programu se mora navesti direktiva pretprocesoru kojoj prethodi znak `#`.

U navedenom programu direktiva pretprocesoru `#include <stdio.h>` ukazuje da se u programu koriste funkcije iz datoteke `stdio.h` (**s**tandard **i**nput/**o**utput **h**eader).

Funkcija `main` kao i svaka druga funkcija se sastoji iz zaglavlja i tela funkcije. U ovom slučaju, telo funkcije sa sastoji iz poziva jedne funkcije `printf` koja je definisana u `stdio` datoteci.

Karakter `'\n'` označava prelazak u novi red. Svaka prosta naredba u C-u završava se sa znakom `;`.

Primer 2 *Šta je izlaz iz sledećeg programa?*

```
#include <stdio.h>

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

Izlaz iz programa:
Zdravo, svete!

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

Znaci %d u okviru funkcije main ukazuju prevodiocu gde i u kom formatu treba da štampa vrednosti promenljivih pom1, pom2 i rez.

Primer 4

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

Izlaz iz programa:

Slova:
z
Z

Primer 5 *Program prikazuje unos celog broja koristeći funkciju scanf("%d", &x)*

```
#include <stdio.h>

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);
}
```

Zadaci za praktikum:

Zadatak 1 *Napisati program koji sabira dva cela broja sa ulaza.*

Zadatak 2 *Napisati program za razmenu vrednosti dva cela broja.*

Zadatak 3 *Napisati program za izračunavanje površine i zapremine pravog valjka zadatog svojom visinom H i poluprečnikom osnove r .*

Zadatak 4 *Napisati program koji izračunava maksimum i minimum 2 cela broja sa ulaza.*

Zadatak 5 *Napisati program koji izračunava maksimum i minimum 3 cela broja sa ulaza.*

Zadatak 6 *Napisati program koji izračunava broj koji se dobija kada se 110 umanji za 10% (rešenje je 99).*