

Ulazno-izlazne naredbe

Razmenu podataka između programa i standardnog ulaza (tastatura, fajl na disku, ...) i izlaza (ekran, štampač, disk, ...) možemo izvršiti korišćenjem ulazno-izlaznih naredbi. Ulazna naredba programskog jezika C za učitavanje sa tastature je **scanf**, a izlazna naredba za ispisivanje na ekran je **printf**. U programskom jeziku C postoje i druge varijante ovih naredbi, namenjene za učitavanje iz datoteka i stringova, kao i za štampanje u datoteke i stringove.

Ulazne naredbe

Za rad većine programa potrebno je obezbediti neophodne ulazne podatke nad kojima će program obavljati određene aktivnosti. Iako je najčešći slučaj da se ulazni podaci definišu na početku rada programa, nisu retki slučajevi kada program tokom čitavog svog rada zahteva unos određenih podataka, koji su mu neophodni za nastavak.

Kada u programu deklariramo određene promenljive, time smo samo obezbedili memorijsku lokaciju za smeštanje njihovih vrednosti. Kod većine kompajlera deklarirana promenljiva dobija vrednost koja je pre toga bila zapisana na toj lokaciji i koja najčešće nema nikakvog smisla za dati program. Sa druge strane, neki kompajleri prilikom deklaracije automatski postavljaju vrednost promenljive na neku inicijalnu vrednost (na pr. 0 za celobrojnu promenljivu), ali se na to nikada ne treba oslanjati, jer tako napisan program neće funkcionisati pod svim kompajlerima. Iz tog razloga, pre korišćenja sve promenljive moraju dobiti inicijalnu vrednost u samom programu ili se njihove vrednosti moraju učitati sa nekog ulaznog uređaja.

Naredba za učitavanje vrednosti sa standardnog ulaza u programskom jeziku C je **scanf**. Ova naredba preuzima vrednosti sa standardnog ulaza i dodeljuje ih odgovarajućim promenljivama.

Sintaksa

```
scanf("%format_unosa", &naziv_promenljive);
```

Format unosa	Tip promenljive
%d	int
%u, %o, %x	unsigned int
%f	float
%lf	double
%c	char
%s	niz karaktera (String)

Okako napisane naredbe promenljivama $X_1, X_2, \dots, X_n$ dodeljuju n vrednosti koje su unete sa tastature ili nekog drugog standardnog ulaza. Broj, tip i redosled promenljivih moraju da odgovaraju broju, tipu i redosledu podataka koji se unose. Na primer, ukoliko promenljivama i i j gde je i ceo broj, a j realan broj, treba dodeliti vrednosti 41 i 19.6 sa tastature, onda bi naredba izgledala ovako:

```
scanf("%d%f", &i, &j);
```

a na tastaturi bismo uneli:

```
41 19.6↵
```

pri čemu znak ↵ označava pritisnut taster ENTER na tastaturi, odnosno kraj reda u ulaznoj datoteci (fajlu). Nakon unosa podataka promenljive imaju sledeće vrednosti:

Promenljiva	Vrednost
i	41
j	19.6

Za razdvajanje brojeva prilikom unosa mogu se koristiti jedan ili više praznih mesta (taster SPACE), znak za tabulator → (taster TAB) ili znak za kraj reda ↵ (taster ENTER). Tako, na primer, četiri broja možemo uneti i na sledeće načine:

41↵

7↵

19.6↵

12↵

ili

41 7→19.6→12↵

ili pak

41→7↵

19.6 12↵

Napomenimo i to da nijedan od ovih znakova nije vidljiv na ekranu, ali ih ovde prikazujemo kao oznaku da je potrebno pritisnuti odgovarajući taster.

Ukoliko se vrši učitavanje podataka tipa **char**, naredba **scanf** prazna mesta ne tretira kao separatore, već kao ravnopravne znakove. Tako, na primer, ako učitavamo realnu promenljivu *a* i znakovne promenljive *c1* i *c2* korišćenjem komande

```
scanf("%f%c%c", &a, &c1, &c2);
```

ulazna linija

2.54E+2 Pera

će dati sledeće vrednosti promenljivama:

Promenljiva	Vrednost
a	254.0
c1	' '
c2	'P'

Promenljive tipa **string** se učitavaju tako što naredba **scanf** očitava sve karaktere do razmaka. Tako, ukoliko su promenljive deklarisanе na sledeći način:

```
char rec1[10], rec2[5];
```

onda će komande

```
scanf("%s", rec1);
```

```
scanf("%s", rec2);
```

u slučaju sledećeg unosa:

Ana voli Milovana

kao rezultat dati:

Promenljiva	Vrednost
rec1	'Ana'
rec2	'voli'

Podaci logičkog tipa se mogu učitati kao podaci tipa **int**, ali njihova vrednost ne može biti manja od nule.

Izlazne naredbe

Nakon obrade podataka u programu, potrebno je rezultate prikazati korisniku ili ih zabeležiti na nekom medijumu. Da bi se podaci ispisali na ekranu koristi se naredba programskog jezika C **printf**, koja vrši ispisivanje navedenih podataka na standardnom izlazu. Pored ispisa navedenih podataka, korišćenjem ove naredbe moguće je definisati i format u kome se ti podaci ispisuju, kako bi prikaz bio čitljiviji za korisnika.

Naredba **printf** ispisuje na standardnom izlazu vrednosti na osnovu zadatog formata i zadatih vrednosti (promenljivih).

Sintaksa

```
printf("%format_ispisa", vrednost);
```

Na primer, naredba:

```
printf("%d %d", 1, 2);
```

prikazuje na ekranu

```
1 2
```

Naredba:

```
printf("%d + %d = 3", 1, 2);
```

prikazuje na ekranu

```
1 + 2 = 3
```

Ako želimo da u ispisu pređemo u novi red koristimo **\n**.

Primer bi bio:

```
printf("%d", 1);  
printf("\n");  
printf("+ %d\n = 3", 2);
```

onda bismo kao rezultat dobili sledeći prikaz na ekranu:

```
1
```

```
+ 2
```

```
= 3
```

Kao što se vidi iz prethodnog primera, za ispisivanje praznog reda može se koristiti komanda **printf("\n")**.

Prilikom ispisivanja podataka na ekranu ili bilo kom drugom izlaznom uređaju, treba voditi računa o tome da ti podaci budu napisani pregledno kako bi bili lako čitljivi za korisnika. Naredbe **printf** omogućava ispisivanje podataka u određenom formatu i to:

- definisanje širine polja u kome se ispisuje podatak
- definisanje broja decimalnih cifara pri ispisivanju realnih brojeva

Definisanje širine polja u kome se ispisuje podatak se vrši tako što se u formatu unosa zadaje broj karaktera koji će zauzimati taj podatak na ekranu. Ukoliko podatak koji se ispisuje zauzima manje mesta od navedene širine polja, onda se suvišna polja popunjavaju prazninama ispred podatka. Ako je navedena širina polja nedovoljna za ispis podatka, onda se polje proširuje za potreban broj pozicija.

Primer

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

```
main()
```

```
{
```

```
    int x = 428;
```

```
    double y = -8475.54;
```

```
    double z = 5.645e+03;
```

```
    printf("%8d \n", x);
```

```
    printf("%10.21f \n", y);
```

```
    printf("%5d%5d \n", x, x);
```

```
}
```

Nakon izvršenja ovog programa biće ispisan sledeći rezultat na ekranu:

					4	2	8
--	--	--	--	--	---	---	---

		-	8	4	7	5	.	5	4
--	--	---	---	---	---	---	---	---	---

		4	2	8			4	2	8
--	--	---	---	---	--	--	---	---	---