

B

Naredbe za ulaz i izlaz

Na početku knjige dali smo osnovne principe korištenja funkcija `printf` i `scanf` što smo smatrali dovoljnim. Kroz primjere u nastavku knjige mogli ste upoznati još neke načine korištenja ovih funkcija. U ovom prilogu pokušaćemo predstaviti *sve* što je bitno za ulaz i izlaz u programskom jeziku C kao svojevrsnu referencu za sve one situacije kada vam je potrebna tabela svih mogućih vrijednosti.

B.1 `printf` i `scanf`

B.1.1 Formatni stringovi (formati)

U tabeli ispod dati su svi poznati formati, tj. formatni stringovi koje možete koristiti u funkcijama `printf` i `scanf`.

Format	Značenje
<code>%d, %i</code>	Cijeli broj (<code>int</code>)
<code>%u</code>	Nepredznačni cijeli broj (<code>unsigned</code>)
<code>%o</code>	Cijeli broj u oktalnom brojevnom sistemu
<code>%x, %X</code>	Cijeli broj u heksadecimalnom brojevnom sistemu
<code>%f, %F</code>	Realan broj (<code>float</code> , <code>double</code>)
<code>%g, %G</code>	Ispisuje realan broj koristeći minimalan broj decimala
<code>%e, %E</code>	Realan broj u naučnoj notaciji npr. <code>3.9265E+2</code>
<code>%a, %A</code>	Realan broj u heksadecimalnom sistemu npr. <code>-0XC.90FEP-2</code>
<code>%c</code>	Znak (<code>char</code>)
<code>%s</code>	String (niz tipa <code>char</code>)
<code>%p</code>	Adresa u memoriji (pointer)
<code>%n</code>	Ništa se ne ispisuje/učitava, broj obrađenih znakova se upisuje u neku promjenljivu
<code>%%</code>	Znak postotka

Tablica B.1: Formatni stringovi za funkcije `printf` i `scanf`

Kod formatnih stringova koji imaju varijantu sa malim i velikim slovom (npr. `%f` i `%F`) razlika je isključivo u tome da li će se vrijednosti `inf` (beskonačno) i `nan` (ne-broj) ispisati malim ili velikim slovima (`INF`

i `NAN`). Format `%f` očekuje da heksadecimalni broj bude dat malim slovima (npr. `0xf358a`), a format `%F` velikim slovima (`0xF358A`).

Tabela je važeća za `printf` i `scanf`, ali postoji nekoliko sitnih razlika između načina kako ove dvije naredbe tumače pojedine formate. Spominjali smo ranije da za tip `double` u `scanf` trebamo koristiti format `%lf`, dok u `printf` takav format uopšte ne postoji nego se za sve realne tipove treba koristiti `%f`. Formati `%d` i `%i` su sinonimi u funkciji `printf`, a u funkciji `scanf` format `%i` dozvoljava unos oktalnih i heksadecimalnih brojeva tako što započnete unos nulom. Tako npr. ako je korisnik u promjenljivu tipa `int` unio `0123`, sa formatom `%d` vrijednost promjenljive će biti `123`, a sa formatom `%i` će biti `83` jer će se unos tumačiti kao oktalni broj pošto počinje nulom.

Naglasimo da format `%n` i u funkciji `printf` očekuje adresu promjenljive a ne promjenljivu (npr. `&broj`) jer će se u tu promjenljivu upisati broj obrađenih znakova, tj. u slučaju `printf`-a broj znakova ispisanih na ekran:

```
printf("%d plus %d daje %d\n", a, b, a+b, &duzina);
printf("Linija iznad je duga %d znakova\n", duzina);
```

B.1.2 Podspecifikatori

Pojmom **podspecifikator** označavamo sve ono što navodimo između znaka postotka i slova kako bismo pobliže uticali na format. Svi mogući podspecifikatori dati su u tabeli ispod.

Tablica B.2: Podspecifikatori formatnih stringova

Podspecifikator	Funkcije	Značenje
- (minus)	<code>printf</code>	Poravnanje ispisa ulijevo
+ (plus)	<code>printf</code>	Obavezan ispis predznaka broja (čak i kad je to plus)
. (tačka)	<code>printf</code>	Preciznost realnog broja npr. <code>%5.2f</code>
0 (nula)	<code>printf</code>	Popunjava predviđeni prostor nulama
* (zvjezdica)	<code>printf</code>	Prostor i/ili preciznost su dati kao parametri npr. <code>printf("%*.*f", 5, 2, x);</code>
* (zvjezdica)	<code>scanf</code>	Zanemaruje učitano vrijednost
hh	Objekti	Tip <code>char</code>
h	Objekti	Tip <code>short</code>
l	Objekti	Tip <code>long</code> ili <code>double</code>
ll	Objekti	Tip <code>long long</code>
L	Objekti	Tip <code>long double</code>
j	Objekti	Tip <code>intmax_t</code>
z	Objekti	Tip <code>size_t</code>
t	Objekti	Tip <code>ptrdiff_t</code>

Možemo uočiti da su podspecifikatori podijeljeni u tri grupe. Prva četiri utiču na način ispisa na ekran (`printf`) a u funkciji `scanf` nisu dostupni. Primjeri njihovog korištenja su nam poznati od ranije.[†]

Podspecifikator zvjezdica ima potpuno različito značenje u funkcijama `printf` i `scanf` pa smo ga naveli dvaput. U funkciji `printf` se može

[†] Poglavlje 1.2.4, str. 31.

koristiti i sa drugim formatima, pa tako sljedeća naredba ispisuje broj `br` u prostoru od `sir` znakova: `printf("%*d", sir, br);` Ovaj podspecifikator ima efekat “zanemarivanja” u funkciji `scanf`. Npr. sljedeća naredba:

```
scanf("%d%d%d", &a, &b);
```

omogućuje korisniku da unese tri cijela broja razdvojena razmakom, pri čemu će prvi i treći biti uneseni u promjenljive `a` i `b`, a drugi broj će biti zanemaren.

Treća grupa podspecifikatora dostupna je u obje funkcije i služi da se pobliže odredi tip podatka, pri čemu osnovni format ostaje `d` (ili `i`) za cijeli broj, `u` za nepredznačni cijeli broj, `a` i `f` za realan broj. Npr. format `%hhu` odgovara tipu `unsigned char`, format `%ld` tipu `long int`, dok je za `long double` uveden poseban format jer je ovaj tip uveden tek standardom C99. Ako u funkciji `printf` zaboravite ovaj podspecifikator, najčešće se neće desiti ništa strašno, no za `scanf` je praktično garantovano da neće raditi ispravno bez korektne oznake tipa.

B.1.3 Escape sekvence

Escape sekvence (eng. sequence)¹ nam služe kako bismo u funkciji `printf` na ekran ispisali neke znakove koji se ne mogu direktno navesti unutar navodnika. Npr. ako želimo da na ekran ispišemo dvije riječi jednu ispod druge, mogli bismo probati ovako:

```
printf("Zdravo  
svijete");
```

Međutim, ovaj program se u pravilu neće moći prevesti, jer unutar string literala ne smijemo preći u novi red. Rješenje za ovaj problem glasi:

```
printf("Zdravo\nsvijete");
```

Oznaka za prelazak u novi red `\n` se naziva *escape sekvence* za *novi red*. Naziv je dobila po tome što počinje znakom “backslash” odnosno “obrnuta kosa crtica” `\` koji ovdje predstavlja “escape karakter”² tj. “znak za bijeg” jer njime “bježimo” iz normalnog procesa uređivanja teksta.

Nakon znaka “backslash” navodimo znak koji označava naredbu koju želimo da izvršimo, a dostupne naredbe date su u tablici B.3.

Obratite pažnju da neke od navedenih sekvenci nemaju nikakvog smisla na savremenim računarima. Zvučni signal najčešće nećete čuti, a `\f` i `\v` nemaju efekta na ekranu i korisni su samo za štampanje na papir. Što se tiče znakova `\b` i `\r` oni bi mogli imati opisani efekat, ali obratite pažnju da stringovi `"Zdravo"` i `"Zdd\bravo"` nisu jednaki iako će se jednako ispisati na ekranu. Sistemi za automatsko testiranje programa koji upoređuju izlaz vašeg programa sa očekivanim neće prepoznati ova dva teksta kao jednake.

¹ Na žalost u domaćoj literaturi nije šire usvojen neki standardan prevod za ovaj termin. Bukvalan prevod bi bio “nizovi za bijeg”, međutim analogija koju ovo povlači već jako dugo vremena nije adekvatna i smatramo da bi takvo imenovanje bilo štetno za razumijevanje.

² Na starim tekstualnim terminalima, unos znaka za bijeg (najčešće pritiskom na tipku Esc) označavao je izlazak iz editora teksta i ulazak u komandni režim.

Tablica B.3: Dostupne escape sekvence za naredbu printf

Sekvenca	Naziv	Značenje (efekat)
<code>\a</code>	Alert	Proizvodi zvučni signal ako to podržava izvršno okruženje
<code>\b</code>	Backspace	Brisanje znaka lijevo od unesenog
<code>\f</code>	Form feed	Prelazak na sljedeću stranicu
<code>\n</code>	Line feed	Novi red
<code>\r</code>	Carriage Return	Vraćanje na početak reda
<code>\t</code>	Horizontal Tab	Pomjera kursor na imaginarnu okomitu liniju na ekranu
<code>\v</code>	Vertical Tab	Pomjera kursor na imaginarnu vodoravnu liniju na ekranu
<code>\\</code>	Backslash	Ispisuje znak backslash
<code>\'</code>	Apostrof	Ispisuje znak jednostrukih navodnika
<code>\"</code>	Navodnik	Ispisuje znak dvostrukih navodnika
<code>\?</code>	Upitnik	Ispisuje znak upitnika
<code>\nnn</code>	ASCII	Ispisuje znak čija je vrijednost u ASCII tabeli <code>nnn</code> oktalno
<code>\xhh</code>	ASCII	Ispisuje znak čija je vrijednost u ASCII tabeli <code>hh</code> heksadecimalno
<code>\uhhhh</code>	Unicode	Ispisuje znak čija je vrijednost u Unicode tabeli <code>hhhh</code> heksadecimalno

Mnogi bi očekivali da će naredba `printf("abc\\01234");` na ekranu ispisati samo "abc" jer znak `\0` predstavlja oznaku za kraj stringa i sve nakon njega će biti zanemareno.↑ Ali ustvari na ekranu ćemo dobiti:

↑ Str. 264.

```
abc
34
```

Naime, `\012` će biti protumačeno kao oktalna vrijednost znaka u ASCII tabeli, a to je znak čija je ASCII vrijednost 10 koji je jednak prelasku u novi red (`\n`). Igrom slučaja `\0` ima vrijednost nule i u oktalnom sistemu, a to je nul-karakter. Stoga morate paziti ako iz nekog razloga navodite neki tekst nakon znaka `\0` da taj tekst ne počinje ciframa.

Escape znak `\u` (ili `\U`) će za efekat imati da će na to mjesto biti uvršten znak iz tabele Unicode znakova koja sadrži sve međunarodne karaktere, pa tako i kinesko pismo, naša slova itd. Npr. naredba `printf("Grad Biha\u0106");` je jedini način da na ekranu ispišete tekst "Grad Bihać" koji je sukladan C standardu. Preduslov za to je da prozor za ispis koristi Unicode.

Ukoliko navedete neku escape sekvencu u pozivu funkcije `scanf`, ona će najvjerojatnije biti zanemarena.

B.1.4 Povratna vrijednost

Napomenimo još da i `printf` i `scanf` vraćaju vrijednost. Međutim, funkcija `printf` vraća broj *uspješno zapisanih znakova* na ekran, dok funkcija `scanf` vraća broj *uspješno učitanih promjenljivih*. Ovaj fenomen se može ilustrovati sljedećim isječkom koda:

```
int x,y,ps,pp;  
ps = scanf("%d %d", &x, &y);  
pp = printf("%d %d", x, y);  
printf("\n ps=%d pp=%d\n", ps, pp);
```

Vrijednost koju vraća `scanf` (promjenljiva `ps`) bi uvijek trebala biti 2, jer ste unijeli dva broja u dvije promjenljive. Ona će biti manja od dva ako probate unijeti nešto što nije broj. S druge strane, ako ste unijeli npr. brojeve 123 i 45, funkcija `printf` (promjenljiva `pp`) će vratiti broj 6, jer je na ekranu ispisano šest znakova (3+2 cifre te jedan znak razmaka).

