

Obrada ulaznih/izlaznih podataka

Funkcije printf i scanf

Srednja škola fra Slavka Barbarića Čitluk
Izborna nastava

Mario Miletić

Funkcija printf

- Funkcija **printf** se koristi za ispis sadržaja znakovnog niza (*format string*).
- Oblik funkcije printf je sljedeći:
 - `printf (string, izraz1, izraz2, ...)` ;
- Vrijednosti koje se ispisuju mogu biti **konstante**, **varijable** ili neki drugi kompliciraniji izrazi.
- Ne postoji ograničenje za ispis podataka unutar jedne printf naredbe.

Funkcija printf

- Znakovni niz (format string) se navodi unutar dvostrukih navodnika u funkciji printf:

```
printf(" string ", izraz1, izraz2, ...);
```

- Budući da su podaci unutar varijable zapisani u **binarnom obliku**, potrebno ih je konvertirati u **znakovni oblik** kako bi se mogli ispisati unutar funkcije printf.
- Iz tog razloga definiramo **specifikaciju pretvorbe**.

Funkcija printf

- Specifikacija pretvorbe:
 - Počinje znakom %
 - **Vrijednost koja dolazi iza znaka %** označava način na koji se vrši pretvorba podatka unutar varijable iz binarnog u znakovni oblik.
- Na ovaj način moguće je **kontrolirati način ispisa** podataka u programu.

Funkcija printf

- Razlikujemo sljedeće specifikacije pretvorbe u programskom jeziku C:
 - `%d` - koristi se za ispis cijelih brojeva
 - `%u` - koristi se za ispis pozitivnih cijelih brojeva
 - `%f` - koristi se za ispis realnih brojeva u decimalnom obliku
 - `%e` - koristi se za ispis realnih brojeva u eksponencijalnom obliku
 - `%c` – koristi se za ispis znakova (`char`)

Funkcija printf

- Format specifikacije pretvorbe je:

`%m.pX`

m - minimalna veličina polja za ispis podataka

p - preciznost zapisa

X – oblik konverzije (d, f, u, c, ...)

- Vrijednost **p** ovisi o tome koji oblik konverzije (**X**) se primjenjuje (koji oblik podatka se zapisuje).

Funkcija printf

X – oblik konverzije	Preciznost - p
%d, %u	Označava minimalan broj znamenaka za prikaz (ukoliko je potrebno na početak se dodaju nule)
%f, %e	Označava broj znamenaka koje se prikazuju iz decimalne točke (po defaultu je 6)

Primjer:

```
#include <stdio.h>
void main (void) {
    int i;
    float x;

    i=40;
    x=839.21f;

    printf("%d,%5d,%-5d,%5.3d\n", i, i, i, i);
    printf("%10.3f,%-10.3f,%10.3e\n", x, x, x);
}
```

```
40,   40,40   ,   040
839.210, 839.210,8.392e+002

Press any key to continue . . .
```

Funkcija printf

- Ostali znakovi koji se mogu koristiti unutar `printf` funkcije su:

`\n` - prijelaz u novi red

`\t` - horizontalni tab (*horizontal tab*)

- Primjer:

```
printf("Ime\tPrezime\tAdresa\n\tTel\n");
```

```
Ime      Prezime Adresa
        Tel
Press any key to continue . . .
```


Funkcija scanf

- Funkcija `scanf` se koristi za **učitavanje ulaznih podataka** programa u odgovarajućem obliku.
- I u ovom slučaju definiramo **specifikacije pretvorbe** koje su **identične** onima kod `printf` funkcije.
- Oblike naredbe `scanf`:

```
scanf (string, &izraz1, &izraz2, ...);
```

Funkcija scanf

■ Primjer:

```
#include<stdio.h>

void main(void) {

    int i, j;
    float x, y;

    scanf("%d%d%f%f", &i, &j, &x, &y);
    printf("i=%d, j=%d, x=%f, y=%f\n", i, j, x, y);
}
```

```
1 -20 .3 -4.0e3
i=1, j=-20, x=0.300000, y=-4000.000000
```

Podaci koje mi unosimo

Nakon što se izvrši funkcija scanf na redu je funkcija printf

Funkcija scanf

- Prilikom ispisa funkcije `scanf`, potrebno je provjeriti broj ulaznih varijabli i je li pretvorba za određenu varijablu definirana na odgovarajući način.
- Kako radi `scanf`?
 - Za uneseni znakovni niz `scanf` povezuje ulazne podatke s podacima u nizu.
 - Ako podatak nije dobro učitao, `scanf` funkcija se automatski prekida (ne učitava se ostatak niza).
 - Funkcija `scanf` zanemaruje praznine, prelazak u novi red itd..

Funkcija scanf

- Specifikacije pretvorbe scanf funkcije
 - `%d` – gleda se cijeli broj u nizu (gleda se predznak)
 - `%f`, `%e` – gleda se predznak, a potom niz znamenaka uključujući i decimalnu točku (opcionalno se gleda i eksponent – slovo e)