


11.6 Funzione di visualizzazione

Abbiamo ripetutamente utilizzato negli esempi la funzione di immissione dati, `scanf`, e quella di visualizzazione, `printf`. La sintassi di queste funzioni è stata usata solo sommariamente, ed è ora giunto il momento di definirla (in questo paragrafo e nel prossimo) in modo più completo e generale.

La sintassi della funzione `printf` è:

```
int printf(char *format, arg1, arg2, ...)
```

La funzione `printf` converte, ordina in un formato e stampa sul video (detto standard output ) un numero variabile di argomenti `arg1`, `arg2`, ... sotto il controllo della stringa `format`. Questa stringa dettaglia due diverse informazioni: un insieme di caratteri ordinari, che vengono direttamente inviati sul video, e una specifica di conversione per ognuno degli argomenti da visualizzare. Ogni specifica di conversione inizia con il carattere `%` e termina con il carattere di conversione. Dopo il carattere `%` possono essere presenti i simboli illustrati nei punti successivi.

- 1) 1) Zero, uno o più *flag* che modificano il significato della specifica di conversione. Tipici flag per la modifica delle conversioni sono i segni meno, più e il carattere vuoto:

-	il risultato della conversione sarà accostato a sinistra nel campo;
+	il risultato di una conversione con segno inizierà sempre con il segno (+ o -);
carattere	se il primo carattere di una conversione con segno non è un segno, un carattere
vuoto	vuoto precede la visualizzazione del risultato. Ciò significa che se sono specificati sia il carattere vuoto sia il +, il carattere vuoto sarà semplicemente ignorato.

- 2) 2) Una stringa opzionale di cifre decimali che specifica l'ampiezza minima del campo riservato al corrispondente argomento. Se il valore convertito ha meno caratteri di quelli specificati nell'ampiezza del campo, il valore è accostato a sinistra o a destra del campo, dipendentemente dal flag di accostamento. Se il flag di accostamento non è presente il valore viene posto a destra del campo, se è - (vedi punto precedente) il valore viene posto a sinistra.
- 3) 3) Un punto che separa l'ampiezza di campo dalla successiva stringa di cifre, detta precisione.
- 4) 4) La *precisione* che stabilisce:
- - il numero minimo di cifre che debbono apparire per le conversioni `d`, `o`, `u`, `x` e `X`;
 - - il numero di cifre decimali dopo il punto decimale per le conversioni `e` e `f`;
 - - il massimo numero di cifre significative per la conversione `g`;
 - - il massimo numero di caratteri che debbono essere visualizzati in una stringa relativamente alla conversione `s`.

La precisione prende la forma di un punto seguito da una stringa di cifre: una stringa nulla è trattata come zero.

- 6) 1) Una lettera opzionale `l` che specifica l'applicazione delle conversioni `d`, `o`, `u`, `x`, e `X` a un argomento di tipo `long int`.
- 7) 2) Il carattere di conversione da applicare all'argomento corrispondente. I caratteri di conversione sono:
- `d` l'argomento è convertito in notazione decimale;
 - `o` l'argomento è convertito in notazione ottale senza segno e senza lo zero iniziale;
 - `x` l'argomento è convertito in notazione esadecimale senza segno e senza lo `0x` iniziale;
 - `u` l'argomento è convertito in notazione decimale senza segno;
 - `c` l'argomento è considerato un carattere singolo;



- s l'argomento è una stringa; tutti i caratteri che precedono la prima occorrenza del carattere '`\0`' sono visualizzati a meno che non si superi il numero di caratteri specificati nella precisione;
- e l'argomento è interpretato come un `float` o un `double` ed è convertito in notazione decimale secondo il formato `[-]m.nnnnnnE[+/-]xx` dove la lunghezza della stringa di `n` è specificata dalla precisione. Il valore di default della precisione è 6;
- f l'argomento è interpretato come `float` o `double` ed è convertito in notazione decimale secondo il formato `[-]mmm.nnnnn` dove la lunghezza della stringa di `n` è specificata dalla precisione;
- g permette automaticamente di scegliere tra `%e` e `%f` a seconda di quale sia quello più corto; gli zero non significativi non sono visualizzati.

Si osserva che se il carattere che segue il simbolo `%` non è uno dei caratteri di conversione, allora esso è visualizzato. Quindi per stampare il carattere di percentuale basta specificare `%%`. Il primo argomento di `printf` è quello che dice alla funzione, interpretando i caratteri di conversione, quanti argomenti `printf` dovrà trattare e il rispettivo tipo. Se il programmatore non fa in modo che ci sia coerenza tra il numero e il tipo di caratteri di conversione e gli argomenti `arg1, arg2, ...` la funzione `printf` produrrà i più bizzarri effetti. La `printf` ritorna il numero di caratteri che ha visualizzato. Alcuni esempi di specifica relativi a una stringa sono:

```
%10s      |Ciao, lettore|
%-10s     |Ciao, lettore|
%20s      |          Ciao, lettore|
%-20s     |Ciao, lettore          |
%20.10s   |          Ciao, lett|
%-20.10   |Ciao, lett              |
%.10s     |Ciao, lett|
```

■ dove la dimensione del campo è stata convenzionalmente delimitata con il simbolo `|`.