


11.7 Funzione di immissione

La sintassi di `scanf` è:

```
int scanf(char *format, puntat1, puntat2, ...)
```

La funzione `scanf` legge un insieme di caratteri da tastiera (detta standard input ) , li interpreta secondo il formato specificato dalla stringa `format` e li memorizza negli argomenti puntati da `puntat1`, `puntat2` ecc. La stringa di formato, detta anche controllo, contiene le specifiche di conversione, che sono usate da `scanf` per interpretare la sequenza di immissione. Nel controllo possono essere presenti i simboli riportati nei punti successivi:

- caratteri vuoti (blank, tab, newline, formfeed), che sono ignorati;
- caratteri normali escluso `%`, che dovrebbero coincidere con il prossimo carattere non vuoto della sequenza di ingresso;
- le specifiche di conversione formate dal carattere `%`, da un carattere `*` opzionale di soppressione assegnamento, un'ampiezza massima di campo opzionale, un carattere `l` o un carattere `o` opzionali che denotano la dimensione della variabile che ospita il valore in ingresso, e il codice di conversione.

Una specifica di conversione applica la conversione al successivo campo di immissione dati; il risultato della conversione è poi allocato nella variabile puntata dal corrispondente argomento, a meno che non sia presente l'asterisco `*` di soppressione assegnazione. Con quest'ultimo meccanismo è possibile definire dei campi di immissione che sono semplicemente saltati, poiché non danno luogo ad alcuna assegnazione. Un campo di immissione è definito come una stringa di caratteri non vuoti che si estende non oltre l'ampiezza di campo o fino al prossimo carattere vuoto.

Il codice di conversione stabilisce l'interpretazione del campo di immissione. Codici di conversione ammessi sono:

- | | |
|---|---|
| d | ci si aspetta un intero decimale in immissione; il corrispondente argomento deve essere un puntatore a intero; |
| o | ci si aspetta un intero ottale, senza lo zero iniziale; il corrispondente argomento deve essere un puntatore a intero; |
| x | ci si aspetta un intero esadecimale, senza lo 0x iniziale; il corrispondente argomento deve essere un puntatore a intero; |
| h | ci si aspetta un intero <code>short</code> ; il corrispondente argomento deve essere un puntatore a intero; |
| s | ci si aspetta una stringa di caratteri; il corrispondente argomento deve essere un puntatore a un array di |

	caratteri grande abbastanza da contenere i caratteri specificati e il tappo '\0', che è aggiunto automaticamente. Il campo di immissione termina con un carattere vuoto;
c	ci si aspetta un singolo carattere; il corrispondente argomento deve essere un puntatore a carattere. In questo caso non sono trascurati i caratteri vuoti. Per leggere il prossimo carattere non vuoto occorre usare la specifica %1s. Se si specifica una ampiezza di campo, l'argomento corrispondente deve riferirsi a un array di caratteri, per effettuare la lettura del numero di caratteri specificato;
e,f,q	ci si aspetta un numero in virgola mobile; il corrispondente argomento deve essere un puntatore a float. Il formato di immissione dei numeri in virgola mobile è una stringa di cifre, in cui opzionalmente si indica anche il segno, che può contenere un punto decimale, seguito da campo esponente opzionale formato da E oppure e seguito da un intero, eventualmente con segno.

I caratteri di conversione d, u, o e x possono essere preceduti da l oppure da h per specificare il fatto che il corrispondente argomento è un puntatore a long oppure a short invece che a int. Analogamente, i caratteri e, f e g possono essere preceduti da l per indicare che il corrispondente argomento è un puntatore a double invece che a float.

La conversione effettuata dalla funzione scanf sullo standard input termina quando si incontra una costante EOF, alla fine della stringa di controllo, o quando si incontra un carattere in immissione che è in contraddizione con la stringa di controllo. In quest'ultimo caso il carattere che ha provocato la contraddizione non viene letto dallo standard input.

La funzione scanf ritorna il numero di immissioni che è riuscita a concludere con successo. Questo numero può anche essere zero, nel caso in cui si verifichi immediatamente un conflitto tra un carattere in immissione e la stringa di controllo. Se l'immissione si conclude prima del primo conflitto o della prima conversione, scanf restituisce un EOF. Esempi classici di uso di scanf sono:

```
int i; float x; char nome[30];
...
scanf("%d%f%s", &i, &x, nome);
```

Sulla linea di immissione si potrebbe avere

```
8 54.52E-1  Abel
```

memorizzando 8 in i, 5.452 in x, e la stringa "Abel" in nome. Usando sempre la medesima definizione di variabili

```
scanf("%2d%f%d %s", &i, &x, &nome);
```

e avendo sulla linea di immissione

```
56789 0123 babbo
```

verrà assegnato 56 a i, 789.0 a x, verrà ignorato 0123 e assegnato babbo\0 a nome.

Infine si ricordi che l'argomento di scanf deve essere sempre un puntatore. Purtroppo è molto facile scordarlo e di programmi in cui si scrivono istruzioni tipo

```
int n;
...
scanf("%d", n);
```

invece di

```
scanf("%d", &n);
```

è pieno il mondo!