

Problemi con scanf, e come evitarli

Quando si esegue la lettura di un dato da tastiera con la funzione *scanf*, l'utente deve inserire da tastiera il dato e farlo seguire dalla pressione del tasto <enter>.

Il dato inserito viene interpretato secondo il *formato* specificato dal corrispondente carattere di conversione %... inserito nella stringa di formattazione presente nella chiamata a *scanf*.

A seconda del formato associato al dato letto, parte dell'input inserito via tastiera dall'utente può venire trascurata da *scanf*. Ad esempio nell'istruzione

```
[1] scanf ("%c", &VariabileDiTipoChar);
```

il carattere di conversione %c specifica che va letto un char. Se l'utente, prima di premere <enter>, digita più di un carattere, solo il primo di essi viene effettivamente considerato. Tuttavia GLI EVENTUALI CARATTERI SUPERFLUI IMMESSI NON VENGONO ELIMINATI, ma restano "a disposizione" della successiva chiamata a *scanf*, che può quindi produrre risultati inaspettati. Tra questi caratteri che restano a disposizione è incluso IL CARATTERE CORRISPONDENTE AL COMANDO DI "RETURN" (chiamato Carriage Return, o CR). Infatti la pressione del tasto <enter> produce l'immissione di un carattere: il carattere CR, appunto.

Più in generale: tra i caratteri inseriti via tastiera dall'utente *scanf* usa esclusivamente quelli necessari per il tipo di dato specificato dal carattere di formattazione; e comunque, non utilizza mai il carattere CR inserito dall'utente per indicare la fine dell'input.

La cosa non costituisce un problema se quella considerata è l'unica istruzione *scanf* di tutto il programma. Non è un problema nemmeno se esiste una successiva *scanf*, ma questa si riferisce alla lettura di dati numerici: in questo caso, infatti, i caratteri che non possono far parte di un numero (come CR) vengono ignorati.

Se però esiste una istruzione *scanf* successiva e quest'ultima riguarda la lettura di un carattere o di una stringa di caratteri, sorge un problema. Infatti in questo caso il primo (o unico) carattere letto sarà proprio il CR "lasciato indietro" dalla precedente *scanf*. In particolare, se la nuova *scanf* prevede la lettura di un char, all'utente non verrà nemmeno chiesto di inserire nulla da tastiera.

Uno dei modi per evitare ciò è, per ogni *scanf* relativa ad un char o ad una stringa che sia stata preceduta da un'altra *scanf*, eliminare il CR rimasto dalla *scanf* precedente. Ciò si può ottenere aggiungendo in coda alla stringa di controllo della prima *scanf* o in testa alla stringa di controllo della seconda *scanf* il carattere di conversione %*c, che esegue una lettura di carattere *ed elimina il risultato di tale lettura senza inserirlo da nessuna parte*. Più in generale, inserire "*" tra '%' e il carattere che indica il tipo di dato ('c', 'd', 'f', ...) indica che il dato va eliminato.

Ad esempio se all'istruzione [1] segue, ovunque nel programma, una lettura di carattere, perché quest'ultima operi correttamente essa andrà scritta nella forma

```
scanf ("%*c%c", &AltraVariabileDiTipoChar);
```