

Esercitazione 2

Nota 1 Nel linguaggio C, le righe di codice che cominciano con il carattere '#' si dicono *direttive*; non sono vere e proprie istruzioni, ma istruiscono il compilatore a compiere alcune operazioni preliminari, prima della compilazione vera e propria. Le direttive di tipo `define` permettono di associare valori fissi a sigle. Ad esempio, la direttiva

```
#define N 100
```

stabilisce che, da quel punto del programma in poi, ogni occorrenza della sigla N, quando compare isolata, dev'essere sostituita dalla sequenza 100, che, nei casi opportuni, sarà interpretata come numero.

Nota 2 In una espressione, un valore fisso di tipo intero si può scrivere secondo l'usuale notazione posizionale in base 10 (senza punti o virgole per indicare le migliaia). Ad esempio: 193627.

Un valore di tipo `float` o `double` si può scrivere separando la parte decimale con il punto, secondo le convenzioni anglosassoni (ad esempio 1936.27), oppure in una notazione, detta a volte *scientifica*, che illustro attraverso un esempio: 1.782e-2 denota il numero 1.782×10^{-2} , dove uso il punto per separare le parti decimali.

Un valore di tipo `char`, cioè una singola lettera, o, più in generale, un singolo carattere, è denotato racchiudendolo fra singoli apici. Ad esempio, 'a' denota la lettera *a* minuscola, ';' denota il punto e virgola. Alcune combinazioni speciali servono a denotare caratteri "invisibili". Ad esempio, la combinazione '\n' denota l'*a capo*.

Nota 3 D'ora in poi, nella maggior parte degli esercizi, si dirà che il programma deve acquisire, o leggere, i dati dallo *standard input*. Possiamo immaginare che i programmi che scriviamo ricevano i dati attraverso un canale lungo il quale fluiscono dei caratteri. Anche quando usiamo un'istruzione `scanf` per acquisire un numero, in realtà al programma arriva la sequenza di caratteri digitata alla tastiera; la funzione `scanf` ha il compito di interpretare quella sequenza come la rappresentazione di un numero, intero o reale a seconda dei casi. Il canale si chiama appunto *standard input*.

Esercizi

1 Scrivete un programma che legge una serie di dieci numeri interi dallo *standard input*, ne accumula la somma in una variabile, calcola la media aritmetica dei dieci numeri, e stampa somma e media. I dieci valori non devono essere conservati dal programma.

2 Scrivete un programma che legge una serie di numeri reali dallo *standard input*, arrestandosi quando viene inserito il numero zero, calcola la somma dei quadrati dei numeri inseriti e la media dei quadrati, e stampa i risultati. Non potete usare la funzione `pow`.

3 Scrivete un programma che legge una serie di numeri interi dallo *standard input*, arrestandosi quando viene inserito il numero zero, e determina il massimo della serie.

4 Modificate il programma dell'esercizio 2 in questo modo. Attraverso una direttiva `define` fissate un valore di soglia. Scartate dal calcolo tutti i valori inferiori alla soglia. Oltre alla somma e alla media dei quadrati dei valori superstiti, il programma deve anche stampare il numero di valori scartati.