

Programmazione
II Compitino (Versione A)

17 gennaio 2013

Cognome Nome
Matricola

1. Scrivete un programma che chieda ripetutamente all'utente di inserire degli interi, e che termini quando l'utente inserisce il valore 0; potete assumere che l'utente non inserisca più di 100 interi. Al termine deve stampare una sequenza di numeri, in cui l' i -esimo elemento è il numero di volte per cui l' i -esimo numero è comparso. Ad esempio, se l'utente inserisce *61 300 506 61 601 300 300 503 0* il programma deve stampare *2 3 1 2 1 3 3 1* (infatti, il 61 compare 2 volte, il 300 compare 3 volte, il 506 compare 1 volta, il 61 compare 2 volte ecc.).

2. Scrivete una funzione che, data una stringa e un carattere, restituisca il numero di volte che il carattere compare nella stringa *alla fine di una parola*; con questa espressione, intendiamo che il carattere deve o comparire alla fine della stringa, o appena prima di uno spazio. Ad esempio, se la stringa fosse “la mamma ha fatto la manza” e il carattere fosse 'a', la funzione dovrebbe restituire 5.

3. Un magazzino contiene alcune casse, ciascuna caratterizzata da:
- il nome del prodotto contenuto nella cassa (una stringa di al più 100 caratteri);
 - le misure (lunghezza, larghezza, altezza), in metri, della cassa (dei valori float);
 - il peso, in grammi, della cassa (un valore float);
 - il numero di unità di quel prodotto presenti nella cassa.

Definite una struttura `C`, di nome `prodotto`, che rappresenti un prodotto.

Ora scrivete una funzione che, dato un array di prodotti e un numero n (il numero di elementi dell'array effettivamente utilizzati), calcoli e restituisca il peso specifico più alto¹.

¹Il peso specifico è il rapporto fra il peso e il volume della cassa.

4. Scrivete un programma che, ricevendo sulla riga di comando una sequenza di numeri interi, calcoli e stampi la loro media (come valore floating point). Ad esempio, se supponete che il programma si chiami `media`, un tipico esempio di esecuzione potrebbe essere

```
[gnagna]>./media 5 15 3  
7.666666666666666  
[gnagna]>
```

5. Scrivete una funzione con tre argomenti: il primo è un intero (x) il secondo e il terzo sono un puntatori a intero (y e z). La funzione deve operare diversamente a seconda che y e z si riferiscano o no alla stessa zona di memoria; se sì, la funzione deve copiare il valore di x nei due interi (ovvero in uno di essi), mentre in caso contrario deve mettere nella zona puntata da y il valore x e nella zona puntata da z il *doppio* di x .

6. Supponete di avere dichiarato un tipo `persona` come segue:

```
typedef struct {
    char nome[ 500 ], cognome[ 500 ];
    char sesso; /* vale 'm' oppure 'f' */
    int anno_di_nascita;
} persona;
```

Scrivete una funzione che, dato un array di persone e la sua lunghezza, determini e restituisca l'anno di nascita del maschio più vecchio e l'anno di nascita della femmina più vecchia, e memorizzi questi valori in due variabili intere i cui indirizzi sono passati come argomenti.

7. Data la definizione di `prodotto` (vedi sopra), scrivete una funzione che riceve un array di prodotti e la sua lunghezza, il nome di un prodotto, e restituisce quante unità di quel prodotto sono presenti in magazzino (notate che lo stesso prodotto può comparire in più casse).