

## 4.6 Esercizi.

\* 1. Scrivere un programma che, inizializzati in due vettori  $a$  e  $b$  della stessa lunghezza  $n$  valori interi, calcoli la somma incrociata degli elementi:  $a[1] + b[n]$ ,  $a[2] + b[n-1]$ , ... la memorizzi nel vettore  $c$  e visualizzi quindi  $a$ ,  $b$  e  $c$ .

\* 2. Modificare il programma, esaminato nel presente capitolo, che determina il maggiore, il minore e la media degli elementi di un array in modo che vengano diminuiti in media il numero di confronti effettuati nel ciclo durante l'esecuzione.

3. Scrivere un programma che inizializzi e quindi visualizzi un vettore con i valori alternati 0, 1, 0, 1, 0, 1, 0, 1, ... Ripetere l'esercizio con i valori 0, -3, 6, -9, 12, -15, 18, -21, ....

4. Scrivere un programma che, letti gli elementi di un vettore  $v1$  e un numero  $k$ , determini l'elemento di  $v1$  più prossimo a  $k$ .

5. Scrivere un programma che, letti gli elementi di due vettori  $v1$  e  $v2$  di lunghezza 5, determini il vettore  $w$  di lunghezza 10 ottenuto alternando gli elementi di  $v1$  e  $v2$ . Visualizzare  $v1$ ,  $v2$  e  $w$ .  
Per esempio: se  $v1$  e  $v2$  sono i vettori di caratteri

v1	B	N	S	I	O					
v2	E	I	S	M	!					
w	B	E	N	I	S	S	I	M	O	!

si deve ottenere il vettore

6. Scrivere un programma che, letti gli elementi di due vettori  $v1$  e  $v2$  di lunghezza  $n$ , inizializzi un terzo vettore  $w$  di lunghezza  $n$  con i valori

```

w(i) = 1      se v1(i) > v2(i);
w(i) = 0      se v1(i) = v2(i);
w(i) = -1     se v1(i) < v2(i).

```

Visualizzare quindi  $v1$ ,  $v2$  e  $w$ .

7. Scrivere un programma che, inizializzato un vettore di `char` con una stringa di lettere dell'alfabeto e punteggiatura, visualizzi il numero complessivo delle vocali e delle consonanti del vettore.

8. Scrivere un programma di inizializzazione che richieda un elemento controlli, prima di inserirlo nel vettore, se è già presente, nel qual caso chieda che l'elemento sia digitato di nuovo.

9. Scrivere un programma che inizializzi e quindi visualizzi una matrice di `int` in cui ciascun elemento è dato dalla somma dei propri indici.

10. [Matrici simmetriche] Una matrice quadrata  $n \times n$  di un tipo qualsiasi si dice simmetrica se gli elementi simmetrici rispetto alla diagonale principale (dal vertice alto sinistro al vertice basso destro) sono due a due uguali. Scrivere un programma che, letta una matrice quadrata di interi, controlli se è simmetrica.

11. Scrivere un programma che, inizializzata una matrice  $n \times n$ , visualizzi la matrice che si ottiene da quella data scambiando le righe con le colonne.

\* 12. Modificare il programma che calcola il prodotto di due matrici bidimensionali esaminato nel presente capitolo, in modo che sia l'utente a scegliere le dimensioni degli array. Il programma deve verificare la correttezza delle dimensioni inserite.

13. Scrivere un programma che, letta una matrice di interi o reali, individui la colonna con somma degli elementi massima.

\* 14. Scrivere un programma che richieda all'utente i voti delle otto prove sostenute durante l'anno da diciotto studenti di una classe e calcoli la media di ogni studente, la media di ogni prova e la media globale. Il programma dovrà infine visualizzare l'intera matrice e la media globale. [Suggerimento: si utilizzi una matrice di 19 linee e 9 colonne dove nelle prime otto vengono memorizzati in ciascuna linea i voti di uno studente e nella nona la rispettiva media; nella diciannovesima linea viene invece memorizzata la media per prova.]

15. Memorizzare in un array tridimensionale i numeri estratti al gioco del lotto su tutte le ruote per dieci estrazioni consecutive. Verificare su quali ruote e in quali estrazioni si ripete un certo numero passato in ingresso dall'utente.