

3 Array numerici

Questa dispensa propone esercizi sulla scrittura di algoritmi, in linguaggio C, utili alla comprensione dei vettori (*ingl.*, array). La dichiarazione di un vettore di elementi omogenei in C avviene grazie alla seguente dichiarazione:

```
void main() {
tipo_variabile nome_variabile[numero_elementi];

istruzioni;
}
```

dove `tipo_variabile` è il tipo degli elementi del vettore e `numero_elementi` è la lunghezza del vettore (considerata una costante da qui in poi). Solitamente la lunghezza del vettore viene dichiarato come una costante tramite la parola chiave `#define` per migliorare la leggibilità del codice.

Per accedere a singoli elementi dell'array utilizzeremo degli indici che varieranno tra 0 e `numero_elementi - 1`. Per accedere ad un elemento del vettore si utilizzano le parentesi quadre, ad esempio:

```
void main() {
int vettore[10];

vettore[1] = 4;
}
```

il precedente programma dichiara un vettore di interi di dimensione 10 e successivamente inserisce il valore 4 nel secondo elemento del vettore.

Un vettore ha dimensioni fisse. Esse devono essere specificate alla dichiarazione del programma. Non è possibile modificare le dimensioni di un vettore durante l'esecuzione del programma.

3.1 Esercizi

Esercizio 3.1

Scrivere un programma che, dati due vettori di interi di dimensione 10, ne costruisca un terzo di dimensione 20 i cui elementi di posizione pari siano gli elementi del primo vettore e gli elementi di posizione dispari siano gli elementi del secondo vettore.

Provare ad utilizzare un solo vettore di dimensione 20 per risolvere l'esercizio.

Esercizio 3.2

Scrivere un programma che acquisisca un certo numero di valori interi e positivi stabilito dall'utente (massimo 10), e poi stampi a video un vettore contenente tutti i numeri pari della sequenza e un vettore contenente tutti i numeri dispari.

Esercizio 3.3

Scrivere un programma che acquisisca un numero stabilito dall'utente di voti di esami e si assicuri che tutti i voti inseriti siano nell'intervallo [18, 30]. Il numero massimo di voti inseribili è 20, ma l'utente potrebbe decidere di inserire meno voti. Il programma dovrà poi stampare a video un sommario simile a quello di seguito. La media troncata è la media calcolata non contando gli estremi (voti minimo e massimo).

```
STATISTICHE VOTI:  
Numero esami sostenuti: ...  
Media: ...  
Media troncata: ...  
Varianza: ...  
Deviazione standard: ...
```

Per il calcolo la varianza, $\hat{\sigma}^2$, utilizzare la seguente formula:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^N (v_i - \hat{v})^2}{N - 1} \quad (3.1)$$

dove v_i è l' i -esimo voto e \hat{v} è la media campionaria dei voti. La deviazione standard è semplicemente la radice quadrata della varianza.

Esercizio 3.4

Scrivere un programma che, acquisita una sequenza di al massimo 50 numeri interi tra 1 e 100, stampi a video l'istogramma dei divisori. L'istogramma deve avere tutti i valori da 2 al valore massimo immesso diviso per 2. Il programma deve considerare solo i divisori propri, dove un divisore positivo di n diverso da n stesso è chiamato divisore proprio.

Un esempio dell'output del programma è:

```

Quanti numeri vuoi considerare: 5

Inserire il 1o numero: 6
Inserire il 2o numero: 4
Inserire il 3o numero: 3
Inserire il 4o numero: 9
Inserire il 5o numero: 12
2          | ***
3          | ***
4          | *
5          |
6          | *

```

Esercizio 3.5

Scrivere un programma che chieda all'utente di inserire un numero variabile (al massimo 10) di valori reali. Il programma dovrà assicurarsi che l'utente non possa inserire più di 10 valori.

Il programma dovrà cercare, durante l'acquisizione, la posizione del minimo e del massimo.

Infine, il programma dovrà stampare un sommario dei valori inseriti, con l'indicazione della posizione del minimo (indicata con un segno $-$) e del massimo (indicata con un segno $+$), il tutto formattato come indicato di seguito.

```

Quanti elementi vuoi inserire (max: 10)? 100
Il valore inserito è troppo grande!
Quanti elementi vuoi inserire (max: 10)? 4
Inserire il 1o valore: 100
Inserire il 2o valore: 20
Inserire il 3o valore: -1
Inserire il 4o valore: 5.3
+| 100.00
 | 20.00
- | -1.00
 | 5.30

```

Esercizio 3.6

Scrivere un programma che acquisisca un certo numero di valori stabilito dall'utente (massimo 10), e poi acquisisca un eguale quantità di indici, che specificano l'ordine in cui il programma dovrà stampare i valori acquisiti in precedenza. Per esempio:

```

Quanti elementi vuoi inserire (max: 10)? 5
Inserire il 1o valore: 10
Inserire il 2o valore: 20
Inserire il 3o valore: 30
Inserire il 4o valore: 40
Inserire il 5o valore: 50
Qual è il 1o valore che vuoi stampare? 3
Qual è il 2o valore che vuoi stampare? 2
Qual è il 3o valore che vuoi stampare? 1

```

```
Qual è il 4o valore che vuoi stampare? 4
Qual è il 5o valore che vuoi stampare? 5
30
20
10
40
50
```

Esercizio 3.7

Scrivere un programma che acquisisce un numero variabile di interi, con un massimo di 100. Il programma dovrà costruire un insieme (ossia controllare che non ci siano elementi ripetuti) a partire dagli elementi inseriti.

Esercizio 3.8

Implementare le operazioni di intersezione, unione, e differenza insiemistica tra due array acquisiti da tastiera, supponendo siano insiemi (ossia non abbiano elementi ripetuti).