

Laboratorio di programmazione per sistemi ciberfisici

6. Array

Enrico Martini
November 17, 2025

Array

Un array è un insieme di variabili dello **stesso tipo** e con lo **stesso nome**, ma distinte tra loro da un **indice**.

```
1 // Sintassi: tipo nome_array[dimensione];  
2 int voti[10];  
3 float temperature[200];  
4 char testo[1000];
```

Gli array vengono salvati in zone di memoria contigue.

Indicizzazione

Si accede al singolo elemento di questo insieme tramite l'indice, che viene trattato come una variabile:

```
1 voti[0] = 2;      // Accedo al primo elemento
2 temperature[199] = 2.4 + temperature[140];
3 testo[3] = 'c';
```

Il primo elemento di un array ha indice 0 e l'ultimo elemento ha indice pari al numero totale di elementi meno uno. **Il compilatore NON controlla se si utilizza l'indice corretto.**

```
1 // Sintassi: tipo nome_array[dimensione];
2 int voti[6];           // Da 0 a 5
3 float temperature[200]; // Da 0 a 199
4 char testo[1000];      // Da 0 a 999
```

Inizializzazione

Gli array possono essere inizializzati durante la dichiarazione, a partire dal primo elemento. Non è necessario inizializzarli tutti; se vengono inizializzati solo i primi, gli altri sono inizializzati a 0.

```
1 int X[3] = {1,2,3};      // 1, 2, 3
2 int Y[100] = {1,2,3};   // 1, 2, 3, 0, 0, ..., 0
3 int Z[100] = {0};       // 0, 0, ..., 0
4 char text[] = {'H','e','l','l','o'}; // Avrà dimensione 5 automaticamente
```

- ▶ La dimensione di un vettore non può essere mai cambiata dinamicamente
- ▶ Se l'inizializzazione viene fatta insieme alla dichiarazione, allora si può omettere la dimensione

Array multi-dimensionali

Dichiarazione:

```
1 int M[4][5];    // Una matrice di 4 righe e 5 colonne
```

Inizializzazione:

```
1 int M[2][3] = { // Metodo 1: dividere riga per riga  
2 {3,4,5},  
3 {8,7,3}  
4 };
```

```
1 int M[2][3] = {3,4,5,8,7,3};    // Metodo 2: come array mono-dimensionale
```

Internamente vengono salvati come se fossero monodimensionali, riga dopo riga.

Direttive di pre-processing

E' buona pratica scrivere un programma che utilizzi pochi dati alla volta, per poi estenderlo alla quantità di dati completa.

```
1  #include <stdio.h>
2  #define DIM 10
3
4  int main(){
5      int X[DIM];    // X avrà dimensione 10
6      ...
7  }
```

Esercizio 1

Scrivere un programma che chieda all'utente di inserire una serie di 10 numeri e li stampi al contrario.

Esempio:

```
1 in:  1 2 4 5 6 7 8 9 3 2
2 out: 2 3 9 8 7 6 5 4 2 1
```

Suggerimento: utilizzare le direttive di pre-processing per testare il programma su una dimensione minore (ad esempio 3 numeri).

Esercizio 2

Scrivere un programma che chieda all'utente di inserire diverse valutazioni (1-10) e poi ne stampi la distribuzione. Il programma termina quando l'utente inserisce un numero che non sia nel range 1-10.

Esempio:

```
1 in:  5 5 7 9 0
2 out: 0 0 0 0 2 0 1 0 1 0
```


Esercizio 3

Letta in input una sequenza di 5 numeri interi positivi, memorizzarla in un array. Sostituire con 0 tutti gli elementi che sono multipli di 3 e stampare la media di quelli rimanenti.

Esempio:

```
1 in:  3 8 6 4 2
2 out: 4.67
```

Esercizio 4

Letta in input una sequenza di 10 interi ed un intero x, calcolare quante volte compare x nella sequenza, quanti sono gli elementi minori di x e quanti sono quelli maggiori.

Esempio:

```
1 inserisci n numeri: 57 1 3 48 6 4 2 29 1 5
2 7
3 uguali: 0
4 maggiori: 3
5 minori: 7
```

Esercizio 5

Letta in input una sequenza di 10 numeri reali, calcolare quanti di questi numeri sono minori della media.

Esercizio 6

Data una matrice 2x2, se ne calcoli il determinante.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\det(A) = ad - bc$$

Esercizio 7

Scrivere un programma che stampi la tabellina 10x10.

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|----|----|----|----|----|----|----|----|----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 0 | 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 0 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 0 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 0 | 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

Suggerimento: in fase di stampa, usare %3i per spaziare correttamente.

Esercizio 8

Scrivere un programma che legga una matrice di numeri reali 5x5 e poi stampi la somma delle righe e delle colonne.

Esercizio 9

Utilizzando il **crivello di Eratostene** stampare tutti i numeri primi minori di 100.

Il procedimento è il seguente: si scrivono tutti i numeri naturali a partire da 2 fino a n in un elenco, detto setaccio. In seguito si cancellano (setacciano) tutti i multipli del primo numero del setaccio escluso lui stesso. Si prende poi il primo numero non cancellato maggiore di 2 e si cancellano tutti i suoi multipli eccetto lui, e si ripete questa operazione fino a che il primo numero non cancellato maggiore di 2 non presenta multipli tra i numeri rimasti nell'elenco. I numeri che restano sono i numeri primi minori o uguali a n .

Esercizio 10

Dati in input una matrice A di numeri reali con 3 righe e 3 colonne e un numero x , stampare in output, se esiste, il maggiore tra gli elementi di A che è minore di x .

Esercizio 11

Letta in input una sequenza di numeri reali, visualizzare in output le terne consecutive la cui somma è maggiore della media dei valori dell'intera sequenza.