

Laboratorio di Informatica

Corso di laurea triennale in Fisica

CARLO MEREGHETTI

In questa lezione, verrà approfondito l'utilizzo degli array monodimensionali e bidimensionali. Si inizierà inoltre a considerare il concetto di **struct**.

Tutti i file (sorgenti ed eseguibili) prodotti per i *primi tre esercizi* andranno salvati sotto la directory **Lab3**. Per il resto degli esercizi create un'apposita cartella **Lab4**.

1. Data una sequenza numerica x_1, x_2, \dots, x_n , la *media aritmetica* è definita come $\mu = \frac{1}{n} \sum_{i=1}^n x_i$, mentre la *varianza* è definita come $\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$.

Scrivete un programma che chieda all'utente 10 interi e successivamente ne stampi:

- media,
 - varianza,
 - valore minimo,
 - gli interi che distano dalla media più della varianza (a tal proposito, ricordiamo che la funzione `abs(n)` nella libreria `cmath` restituisce il valore assoluto di `n`),
 - tutte le coppie di interi in cui il primo supera il secondo.
2. Scrivete un programma che chieda all'utente 10 interi e un intero `n` e scriva **ESISTE** se `n` compare tra i 10 interi, **NON ESISTE** altrimenti. Modificate il programma per stampare anche il numero di volte in cui `n` occorre tra i 10 numeri.
 3. Scrivete un programma che memorizzi 10 interi chiesti all'utente in un array `A` e successivamente chieda due *posizioni* `x` e `y` entro `A`. Il programma deve controllare che tali posizioni siano ammissibili (i.e., la prima stia alla sinistra della seconda ed entrambe stiano entro i limiti di `A`), chiedendole nuovamente in caso contrario. Il programma deve invertire specularmente `A` tra le posizioni date lasciando inalterato il resto; l'array risultante dev'essere infine stampato.

Ad esempio, se `A` contenesse i valori 1, 2, $\overbrace{3, 4, 5, 6}^{\text{intervallo 2-5}}$, 7, 8, 9, 10 e le posizioni inserite fossero `x = 2` e `y = 5`, riorganizzeremmo `A` come 1, 2, $\underbrace{6, 5, 4, 3}_{\text{intervallo 2-5}}$, 7, 8, 9, 10.

4. Un vettore in \mathbf{R}^n ha la forma $x = (x_1, x_2, \dots, x_n)$, con $x_i \in \mathbf{R}$. Il *prodotto interno* di $x, y \in \mathbf{R}^n$ è definito come $xy = \sum_{i=1}^n x_i y_i$ e la loro *somma* è il vettore $z \in \mathbf{R}^n$ con $z_i = x_i + y_i$. Scrivete un programma che chieda all'utente due vettori in \mathbf{R}^{10} (componenti di tipo `float`) e successivamente ne stampi somma e prodotto interno.
5. Scrivete un programma che chieda all'utente un brano di 10 righe (ricordiamo la funzione `getline(cin, s)` della classe `string` che consente di memorizzare nella variabile `s` di tipo `string` i caratteri inseriti sino al newline, spazi compresi). Successivamente, il programma deve ristampare il brano *mettendo tra due asterischi iniziali e finali* le righe più lunghe del brano (ricordiamo che `s.length()` restituisce la lunghezza della stringa `s`).

6. Dato un array A , uno shift (destra, anche detto permutazione ciclica) di A sposta tutti gli elementi di A a destra di una posizione mettendo infine in prima posizione l'ultimo elemento di A . Ad esempio, operare uno shift sull'array $A = \{1, 2, 3, 4, 5, 6\}$ porterebbe alla situazione $A = \{6, 1, 2, 3, 4, 5\}$. Ovviamente, operando due shift su A otterremo la situazione $A = \{5, 6, 1, 2, 3, 4\}$.

Scrivete un programma che chieda all'utente 10 interi da memorizzare in un array A ed un intero non negativo p (quest'ultimo intero va chiesto nuovamente fintantoché sia negativo). Successivamente, il programma deve operare p shift su A e quindi stamparne il contenuto.

7. Scrivete un programma che chieda all'utente una sequenza di interi terminata da un intero negativo; ogni intero inserito deve essere compreso tra 0 e 5 e altrimenti ignorato. Successivamente il programma deve stampare il *numero di occorrenze* di ogni intero inserito. Ad esempio, se l'utente inserisse la sequenza 1 3 3 4 1 3 -4, il programma dovrebbe stampare:

Numero di occorrenze di 1: 2

Numero di occorrenze di 3: 3

Numero di occorrenze di 4: 1

Suggerimento: ovviamente non possiamo utilizzare un array per memorizzare la sequenza di interi inserita dall'utente, ma un array potrebbe tornare utile per contare le occorrenze di ogni intero inserito. Come?

8. Una matrice M di interi di dimensione $n \times n$ è costituita da n righe di interi ognuna delle quali consiste di n interi (per un totale quindi di n^2 interi disposti a quadrato di lato n). Indicheremo con M_{ij} l'intero posto alla riga i e colonna j , con $1 \leq i, j \leq n$. Dato un vettore $x \in \mathbf{N}^n$ (vedi esercizio 4), si definisce il prodotto Mx come quel vettore $z \in \mathbf{N}^n$ in cui $z_i = \sum_{k=1}^n M_{ik} x_k$.

Scrivete un programma che chieda all'utente una matrice di interi di dimensione 3×3 ed un vettore in \mathbf{N}^3 e ne stampi il prodotto.

9. Utilizzate una **struct** per rappresentare uno studente con *nome*, *cognome* e *numero di matricola* (che potete definire mediante il tipo **int**). Successivamente, memorizzate opportunamente una classe di 5 studenti chiedendo i dati di ogni studente all'utente e stampate gli *omonimi* per cognome e lo studente (o gli studenti) con numero di matricola *maggiore*.