

Esercizi sulle struct — avanzati

Vettori di struct, ricerca, ordinamento e struct annidate in C++

Prima di iniziare. Definisci ogni `struct` a livello globale, dopo `using namespace std;` (per le stringhe serve `#include <string>`). Continua a risolvere suddividendo in funzioni e a leggere le struct *per riferimento*. Le dimensioni dei vettori sono costanti (`const int N = 5;`). Per ogni esercizio trovi la struct da usare e i prototipi suggeriti.

Esercizio 1 — Classifica dei giocatori

Hai un vettore di `N` giocatori, ognuno con nome e punteggio. Ordina il vettore per punteggio decrescente (con selection sort o bubble sort) e stampa la classifica.

vettore di struct · ordinamento

Struct e prototipi suggeriti

```
struct Giocatore { string nome; int punteggio; };  
void leggiGiocatori(Giocatore g[], const int N);  
void ordinaPerPunteggio(Giocatore g[], const int N);  
void stampaGiocatori(Giocatore g[], const int N);
```

Esercizio 2 — Il creator migliore

Dato un vettore di `N` creator (nome, follower e tasso di engagement reale), trova e restituisci il creator con l'engagement più alto, poi stampalo.

vettore di struct · float

Struct e prototipi suggeriti

```
struct Creator { string nome; int follower; float engagement; };  
void leggiCreator(Creator c[], const int N);  
Creator migliorePerEngagement(Creator c[], const int N);
```

Esercizio 3 — Cerca e restituisci

Hai un vettore di `N` studenti (nome, media e numero di assenze). Chiedi un nome e cerca lo studente: se esiste, restituiscilo tramite parametro per riferimento e indica con un valore booleano se è stato trovato.

vettore di struct · ricerca

Struct e prototipi suggeriti

```
struct Studente { string nome; float media; int assenze; };  
void leggiStudenti(Studente s[], const int N);  
bool cercaStudente(Studente s[], const int N, string nome, Studente &trovato);
```

Esercizio 4 — Evento in agenda

Un evento del calendario ha un titolo e una data. La data è a sua volta una struct con giorno, mese e anno. Chiedi all'utente i dati di un evento (struct annidata) e stampalo nel formato `titolo -- gg/mm/aaaa`.

struct annidata

Struct e prototipi suggeriti

```
struct Data { int giorno; int mese; int anno; };
struct Evento { string titolo; Data data; };
void leggiEvento(Evento &e);
void stampaEvento(Evento e);
```

Esercizio 5 — Pagella dello studente

Uno studente ha un nome e un vettore di M voti (numeri reali). Chiedi i dati di uno studente e calcola la sua media. (*Dichiara `const int M = 5`; a livello globale e usalo per il campo `voti[M]`.*)

struct con vettore · float

Struct e prototipi suggeriti

```
struct Studente { string nome; float voti[M]; };
void leggiStudente(Studente &s);
float media(Studente s);
```

Esercizio 6 — Solo i promossi

Dato un vettore di N studenti (nome e media), costruisci un nuovo vettore contenente solo i promossi (media maggiore o uguale a 6), restituendone il numero, e stampalo.

vettore di struct · filtro

Struct e prototipi suggeriti

```
struct Studente { string nome; float media; };
void leggiStudenti(Studente s[], const int N);
int filtraPromossi(Studente s[], const int N, Studente promossi[]);
void stampaStudenti(Studente s[], const int N);
```

Esercizio 7 — Brani per genere

Hai un vettore di N brani, ognuno con titolo e un codice di genere (un intero da 1 a G). Conta quanti brani appartengono a ciascun genere, salvando i conteggi in un apposito vettore.

vettore di struct · conteggio

Struct e prototipi suggeriti

```
struct Brano { string titolo; int genere; };
void leggiBrani(Brano b[], const int N);
void contaPerGenere(Brano b[], const int N, int conteggi[], const int G);
```

Esercizio 8 — Bilancio della stagione

Dato un vettore di N partite (avversario, punti fatti e punti subiti), calcola il numero di vittorie e la differenza punti totale (somma di punti fatti meno punti subiti). Stampa il bilancio.

vettore di struct

Struct e prototipi suggeriti

```
struct Partita { string avversario; int puntiFatti; int puntiSubiti; };  
void leggiPartite(Partita p[], const int N);  
int contaVittorie(Partita p[], const int N);  
int differenzaTotale(Partita p[], const int N);
```

Esercizio 9 — Rifornisci il magazzino

Hai un vettore di N prodotti (nome, prezzo e scorta). A tutti i prodotti con scorta inferiore a una certa soglia, aggiungi una quantità fissa di pezzi. Stampa poi l'elenco aggiornato.

vettore di struct · modifica

Struct e prototipi suggeriti

```
struct Prodotto { string nome; float prezzo; int scorta; };  
void leggiProdotti(Prodotto p[], const int N);  
void rifornisci(Prodotto p[], const int N, int soglia, int aggiunta);  
void stampaProdotti(Prodotto p[], const int N);
```

Esercizio 10 — Ordine alfabetico

Dato un vettore di N studenti (nome e media), ordinalo in ordine alfabetico per nome e stampalo. *(Le stringhe si confrontano direttamente con gli operatori di confronto.)*

vettore di struct · ordinamento

Struct e prototipi suggeriti

```
struct Studente { string nome; float media; };  
void leggiStudenti(Studente s[], const int N);  
void ordinaPerNome(Studente s[], const int N);  
void stampaStudenti(Studente s[], const int N);
```