

Esercizi sui puntatori — avanzati

Aritmetica dei puntatori, struct e array in C++

Prima di iniziare. Una funzione può *restituire* un puntatore, ricevere struct per puntatore (Tipo `*p`, campo con `p->campo`) e usare l'aritmetica dei puntatori per scorrere gli array.

Esercizio 1 — Puntatore al massimo

Scrivi una funzione che restituisca un *puntatore* all'elemento massimo di un vettore.

puntatore + vettore

Prototipo suggerito

```
int* puntatoreMassimo(int v[], const int N);
```

Esercizio 2 — Indirizzi successivi

Stampa gli indirizzi dei primi 5 elementi di un vettore di interi e osserva di quanti byte aumentano.

aritmetica puntatori

Esercizio 3 — Riferimenti vs puntatori

Scrivi due versioni della funzione di scambio: una con i riferimenti (`&`) e una con i puntatori (`*`). Confrontale.

riferimento vs puntatore

Prototipi suggeriti

```
void scambiaRif(int &a, int &b);  
void scambiaPtr(int *a, int *b);
```

Esercizio 4 — Struct per puntatore

Scrivi una funzione che, ricevuta una struct `Punto` per puntatore, la sposti aggiungendo un valore a `x` e `y`.

puntatore + struct

Struct e prototipo suggerito

```
struct Punto { float x; float y; };  
void sposta(Punto *p, float dx, float dy);
```

Esercizio 5 — Doppio risultato

Scrivi una funzione che, dati due interi, restituisca quoziente e resto tramite due puntatori.

puntatore

Prototipo suggerito

```
void dividi(int a, int b, int *quoziente, int *resto);
```

Esercizio 6 — Inverti con due puntatori

Inverti un vettore usando due puntatori che si muovono uno dall'inizio e uno dalla fine.
aritmetica puntatori

Prototipo suggerito

```
void inverti(int v[], const int N);
```

Esercizio 7 — Conta via puntatore

Scrivi una funzione che conti gli elementi pari di un vettore e restituisca il risultato tramite un puntatore. **puntatore + vettore**

Prototipo suggerito

```
void contaPari(int v[], const int N, int *risultato);
```

Esercizio 8 — Lunghezza C-stringa

Scrivi una funzione che calcoli la lunghezza di una C-stringa scorrendola con un puntatore fino al terminatore. **puntatore + C-stringa**

Prototipo suggerito

```
int lunghezza(char *s);
```

Esercizio 9 — Minimo e massimo insieme

Scrivi una funzione che trovi minimo e massimo di un vettore restituendoli tramite due puntatori. **puntatore + vettore**

Prototipo suggerito

```
void minMax(int v[], const int N, int *minimo, int *massimo);
```

Esercizio 10 — Somma di matrice per puntatore

Scrivi una funzione che riceva una matrice e ne restituisca la somma totale degli elementi, accedendovi tramite puntatore. **puntatore + matrice**

Prototipo suggerito

```
int sommaMatrice(int m[][C], const int R);
```