

Esercizi sui puntatori — base

Indirizzi, dereferenziazione e passaggio per riferimento in C++

Prima di iniziare. Un puntatore contiene l'*indirizzo* di una variabile. Con `&x` ottieni l'indirizzo di `x`, con `*p` accedi al valore puntato. Stampa pure gli indirizzi per capire cosa succede in memoria.

Esercizio 1 — Indirizzo e valore

Dichiara una variabile intera e un puntatore a essa. Stampa il valore della variabile, il suo indirizzo e il valore letto tramite il puntatore.

puntatore

Esercizio 2 — Modifica via puntatore

Modifica il valore di una variabile usando il puntatore (non la variabile direttamente) e verifica il cambiamento.

puntatore

Esercizio 3 — Scambia con puntatori

Scrivi una funzione che scambi due variabili intere usando i puntatori.

puntatore

Prototipo suggerito

```
void scambia(int *a, int *b);
```

Esercizio 4 — Raddoppia

Scrivi una funzione che raddoppi il valore di una variabile passata tramite puntatore.

puntatore

Prototipo suggerito

```
void raddoppia(int *n);
```

Esercizio 5 — Somma e risultato via puntatore

Scrivi una funzione che calcoli la somma di due numeri e la restituisca scrivendola in una variabile passata per puntatore.

puntatore

Prototipo suggerito

```
void somma(int a, int b, int *risultato);
```

Esercizio 6 — Puntatore a un elemento

Dato un vettore, fai puntare un puntatore al suo terzo elemento e stampa il valore puntato.

puntatore + vettore

Esercizio 7 — Scorri con il puntatore

Stampa tutti gli elementi di un vettore usando un puntatore che avanza (aritmetica dei puntatori).

puntatore + vettore

Esercizio 8 — Incrementa tutti

Scrivi una funzione che, ricevuto un vettore, incrementi di 1 ogni elemento (le modifiche restano nel chiamante).

puntatore + vettore

Prototipo suggerito

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

Esercizio 9 — Per valore o per riferimento

Scrivi due funzioni che provano a raddoppiare un numero: una riceve il parametro *per valore*, l'altra *per riferimento*. Verifica quale modifica davvero la variabile del chiamante e spiega perché.

valore vs riferimento

Prototipi suggeriti

```
void perValore(int n);  
void perRiferimento(int &n);
```

Esercizio 10 — Puntatore nullo

Dichiara un puntatore inizializzato a `nullptr` e, prima di usarlo, controlla che non sia nullo; assegnagli poi l'indirizzo di una variabile e usalo.

puntatore