3 - Iterazione

Selezione multipla

L'istruzione `switch` in C è utile quando si vuole eseguire diverse istruzioni basate sui possibili valori di una variabile, è particolarmente comoda quando ci sono molti casi da gestire, evitando il sovraccarico di selezioni `if-else` innestate.
Non esiste un costrutto equivalente nei flowchart, rendendo lo `switch` più una pratica del linguaggio di programmazione che un concetto teorico.

Iterazioni e cicli

I cicli sono fondamentali per ripetere un insieme di istruzioni finché una condizione è vera. Si distinguono in base al momento in cui la condizione viene verificata:

- **Iterazione pre-condizionale (`while`)**: La condizione viene verificata prima dell'esecuzione del ciclo. Questo tipo di ciclo è utile quando non si sa quante volte bisogna ripetere le istruzioni, ma si vuole essere sicuri che la condizione sia soddisfatta dall'inizio.
- **Iterazione post-condizionale (`do-while`)**: Qui la condizione viene verificata dopo l'esecuzione di almeno un'iterazione. Questo significa che il ciclo verrà sempre eseguito almeno una volta, anche se la condizione è falsa inizialmente.
- **Iterazione limitata (`for`)**: Perfetta quando il numero di ripetizioni è noto. In un'unica istruzione si inizializza un contatore, si definisce la condizione di esecuzione e si specifica come il contatore deve essere aggiornato. È ideale per eseguire operazioni come sommare numeri o calcolare medie su una serie predefinita.

Il ciclo `for` è particolarmente efficiente perché riduce gli accessi in memoria, tutte le operazioni legate al contatore (inizializzazione, condizione e aggiornamento) avvengono in un'unica istruzione.
L'uso di costanti per definire il numero massimo di iterazioni rende il codice più flessibile e facile da aggiornare.

Sentinelle e Flag

Quando non si sa a priori quante volte ripetere un ciclo, si possono usare **valori sentinella** o **flag**:

- **Sentinella**: Un valore che segna la fine dell'input. Ad esempio, un programma può continuare a leggere numeri finché non si inserisce un valore specifico come `-1`.
- **Flag**: Una variabile booleana che viene impostata a `true` o `false` per controllare l'esecuzione del ciclo.

Incremento e decremento

C in particolare offre operatori unari per incrementare (`++`) o decrementare (`--`) il valore di una variabile. Possono essere usati in due modalità:

- **Prefissa** (`++variabile`): Incrementa il valore prima di utilizzarlo.
- **Postfissa** (`variabile++`): Utilizza il valore e poi lo incrementa.

Abbreviazioni

Il linguaggio C consente di scrivere in modo più compatto alcune istruzioni. Ad esempio, invece di scrivere `conta = conta + 1`, si può usare `conta += 1`. Questo si applica a tutti gli operatori aritmetici come `+`, `-`, `*`, `/` e `%`.