

1 Teoria

- Qual è la differenza tra i cicli `for`, `while` e `do-while`?
- In quali casi è preferibile usare un ciclo `for` anziché un `while`?
- Il ciclo `do-while` può non eseguire mai il blocco di codice? Motiva la risposta.
- È possibile creare un ciclo infinito? In quali casi potrebbe servire?
- Cosa succede se non aggiorni la variabile di controllo all'interno del ciclo?
- Qual è la sintassi corretta di un ciclo `for` in Java?
- È possibile annidare cicli? Quali accorgimenti bisogna prendere?
- Quando conviene usare la parola chiave `break` all'interno di un ciclo?
- A cosa serve la parola chiave `continue` in un ciclo?

2 Strutture Iterative

1. **Stampa numeri da 1 a 10**
Usa un ciclo `for` per stampare i numeri da 1 a 10 inclusi.
2. **Somma dei primi 100 numeri naturali**
Calcola e stampa la somma dei numeri da 1 a 100 usando un ciclo.
3. **Tabellina di un numero**
Chiedi all'utente un numero intero e stampa la sua tabellina fino a 10.
4. **Conto alla rovescia**
Stampa un conto alla rovescia da 10 a 1 utilizzando un ciclo `while`.
5. **Numeri pari da 0 a 50**
Usa un ciclo `for` per stampare solo i numeri pari compresi tra 0 e 50.
6. **Somma fino a zero**
Chiedi numeri interi all'utente finché non inserisce 0. Alla fine, stampa la somma di tutti i numeri inseriti (escludendo lo 0).
7. **Password corretta**
Usa un ciclo `do-while` per continuare a chiedere la password finché non viene inserita quella corretta.
8. **Fattoriale di un numero**
Chiedi un numero intero positivo e calcola il suo fattoriale.
9. **Conta cifre di un numero**
Chiedi un numero intero positivo e conta quante cifre contiene (es. 1234 → 4 cifre).

10. **Numeri divisibili per 3**

Stampa tutti i numeri da 1 a 100 divisibili per 3, saltando quelli divisibili anche per 5 (`continue`).