

# Esercitazione 4



UNIVERSITÀ  
DEGLI STUDI  
DI GENOVA

Dibris

Gualtiero Volpe  
[gualtiero.volpe@unige.it](mailto:gualtiero.volpe@unige.it)

# Soluzione Esercizi Terza Esercitazione

Trovate le soluzioni di questi esercizi qui:

[https://github.com/gualtierovolpe/fondamenti di informatica 2023-24](https://github.com/gualtierovolpe/fondamenti_di_informatica_2023-24)

Le soluzioni delle esercitazioni vengono rilasciate il giorno prima dell'esercitazione successiva.



# Un po' di teoria

*Individuare e correggere gli errori nel seguente testo:*

In un programma, un ciclo può essere eseguito un numero prefissato o un numero imprecisato di volte. Nel primo caso si parla di ciclo indefinito, nel secondo di ciclo definito. I cicli definiti sono controllati da un contatore che conta il numero di volte che il ciclo viene eseguito. Il contatore è posto a zero prima di iniziare il ciclo e quindi decrementato a ogni iterazione. I cicli indefiniti sono controllati da valori sentinella. Tali valori indicano generalmente la fine dei dati e vanno sempre scelti tra i valori che ricorrono frequentemente all'interno dei dati in esame. Nel caso di ciclo indefinito, la condizione di terminazione dipende quindi dal valore sentinella che viene aggiornato ad ogni iterazione all'interno del corpo del ciclo. In C++ i cicli indefiniti vengono in genere implementati con le istruzioni *while* e *do ... while*, mentre quelli definiti sono usualmente implementati con il ciclo *for*.

---

# Un po' di teoria

Si scriva il valore assunto dalle variabili  $a$  e  $b$  dopo l'esecuzione delle seguenti righe di codice:

```
1 int a = 9; int q = 3;
  for (int i = 0; i < 3; i++)
    q += (a * ((q+i) != 0));
```

```
2 int b = 1; int z = 2;
  while (z >= 0) {
    z--; b++; }
  b -= z;
```



# Un po' di teoria

## Prova intermedia del 5 febbraio 2019, esercizio 3

1. Si trasformi il seguente ciclo *for* in un ciclo *while*:

```
const double pi = 3.1415;  
double r = 0.0; cin >> r;  
for (int i = 20; i > 0 && r > 0; i--) {  
    cout << pi * r * r << endl;  
    cin >> r;  
}
```



# Un po' di teoria

## Prova intermedia del 5 febbraio 2019, esercizio 3

2. Si trasformi il seguente ciclo *do ... while* in un ciclo *while*:

```
const double pi = 3.1415;  
double r = 0.0;  
do {  
    cin >> r;  
    cout << 2.0 * pi * r << endl;  
} while (r > 0);
```



# Esercizi da Svolgere

## Esercizio 4.1 – Scomposizione di un numero intero in fattori primi

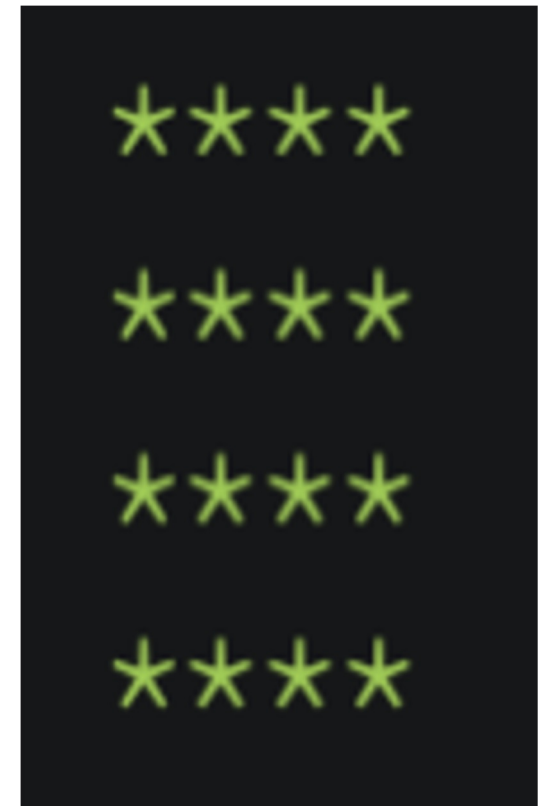
*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera un numero intero positivo e calcoli e stampi a video la sua scomposizione in fattori primi. Ad esempio, se l'utente immette il numero 60, il programma stamperà a video 2 2 3 5. Prima di effettuare il calcolo, il programma verifica se il numero immesso sia effettivamente un numero positivo. In caso contrario, il programma stampa a video un messaggio di errore e chiede all'utente di immettere nuovamente il numero. Inoltre, l'utente potrà rieseguire il programma finché lo desidera.*

---

# Esercizi da Svolgere

## Esercizio 4.2 – Stampa di un pattern

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera un numero intero  $n$  compreso tra 2 e 10 e stampi a video un pattern predefinito, costituito da  $n$  righe di  $n$  asterischi ('\*') ciascuna. Ad esempio, se l'utente immette il numero 4, il programma stamperà a video il pattern visualizzato a lato. Prima di effettuare la stampa, il programma verifica se il numero immesso sia effettivamente compreso tra 2 e 10. In caso contrario, il programma stampa a video un messaggio di errore e chiede all'utente di immettere nuovamente il numero.*

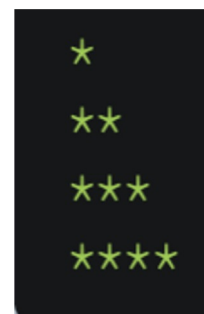




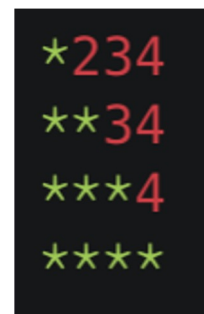
# Esercizi da Svolgere

## Esercizio 4.2 – Stampa di un pattern, estensioni

- a) *Modificare il programma in modo che stampi a video un pattern a forma di triangolo invece che a forma di quadrato. Il pattern è cioè composto da  $n$  righe e la riga  $i$ -esima contiene  $i$  asterischi. Si veda la figura a lato.*
- b) *Modificare ancora il programma in modo che stampi a video un pattern composto da  $n$  righe, nel quale la riga  $i$ -esima contenga inizialmente  $i$  asterischi ('\*') e i successivi elementi siano i numeri da  $i + 1$  a  $n$ . Si veda la figura a lato.*



```
*  
**  
***  
****
```



```
*234  
**34  
***4  
****
```

# Esercizi da Svolgere

## Esercizio 4.3 – Numeri pari e dispari in una sequenza di numeri

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera, uno ad uno, una sequenza di numeri interi positivi, finché non viene inserito un numero negativo. A quel punto, il programma stampa a video quanti numeri pari e quanti numeri dispari sono stati immessi. La sequenza può eventualmente essere vuota.*

## Esercizio 4.4 – Media di una sequenza di numeri

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera, uno ad uno, una sequenza di numeri reali. Al termine, il programma stampa a video la media dei numeri immessi. La sequenza dovrà consistere di almeno un numero.*

---

# Esercizi da Svolgere

## Esercizio 4.5 – Cifre di un numero

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera un numero intero e calcoli e stampi a video di quante cifre il numero è composto.*

## Esercizio 4.6 – Equivalenze tra unità di misura per il peso

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera una sequenza di pesi e, per ciascuno, un carattere alfabetico. Tale carattere è uguale a 'm' se il peso è espresso in milligrammi, a 'g' se è espresso in grammi, a 'h' è espresso in ettogrammi e a 'k' se è espresso in chilogrammi. Il programma converte i pesi in grammi e li stampa a video. Se il peso è già espresso in grammi, il programma passa al peso successivo nella sequenza. La sequenza termina se viene inserito un numero negativo.*

---

# Istruzioni Finali

Cercate di svolgere gli esercizi in autonomia/a piccoli gruppi  
(è il miglior modo per vedere se avete capito)

Per ogni domanda, siamo a disposizione

Troverete le soluzioni di questi esercizi qui:

[https://github.com/gualtierovolpe/fondamenti\\_di\\_informatica\\_2023-24](https://github.com/gualtierovolpe/fondamenti_di_informatica_2023-24)

Le soluzioni delle esercitazioni vengono rilasciate il giorno prima dell'esercitazione successiva.

---