



Esercizi su liste e matrici

Livello INTERMEDIO

 Python Biella Group



Esercizio 1

“Il serpente pigro e le mele numerate”

Il **serpente pigro** vive su una linea retta (una lista di caselle) e vuole mangiare solo le mele che si trovano in posizioni *pari*.

1. Genera una **lista di N numeri casuali** tra 1 e 20, dove ogni numero rappresenta il “valore nutrizionale” di una mela.
Esempio: [3, 8, 2, 15, 9, 4]
2. Fai stampare la lista completa.
3. Usa un **ciclo for** sugli indici per sommare solo i valori delle mele nelle **posizioni pari** (0, 2, 4, ...).



4. Poi usa un ciclo `while` per contare quante mele hanno un valore **maggiore di 10** (cioè "mele giganti").
5. Stampa:
 - la **somma dei valori** delle mele in posizione pari,
 - e il **numero di mele giganti**.

Esempio di output

```
Lista delle mele: [3, 8, 2, 15, 9, 4]  
Somma delle mele in posizione pari: 14  
Numero di mele giganti: 1
```




Esercizio 2

"La Scuola di Magia e la Pagella Stregata"

Nella **Scuola di Magia di Hogwarts**, ogni studente ha un registro di voti (una riga di una matrice) nelle diverse materie: Pozioni, Trasfigurazione, Difesa e Incantesimi.

1. Chiedi all'utente quanti studenti ci sono (`n_studenti`).
2. Crea una **matrice di voti casuali** (da 1 a 10) di dimensione `n_studenti x 4`.
Le colonne corrispondono a: Pozioni, Trasfigurazione, Difesa, Incantesimi.
3. Stampa la **tabella dei voti** in modo leggibile.
4. Usa un **ciclo `for` annidato** per calcolare la **media dei voti di ogni studente**.



5. Usa un ciclo `while` per contare quanti studenti hanno una media ≥ 6 (cioè sono “promossi”).
6. Stampa le medie e il numero totale di studenti promossi.

Esempio di output

Tabella dei voti:

```
[[7, 5, 8, 6],  
 [3, 4, 5, 5],  
 [9, 10, 8, 9]]
```

Medie studenti: [6.5, 4.25, 9.0]

Studenti promossi: 2



Esercizio 3

Il Detective delle Password Deboli

Sei stato assunto come consulente di sicurezza informatica per una scuola che ha scoperto che molti studenti usano password ridicolmente deboli tipo "12345" o "password".

Il tuo compito:

Scrivi un programma che analizzi una lista di password e identifichi quelle "imbarazzanti".

Specifiche:

1. Crea una lista con almeno 8 password (alcune forti, alcune deboli)
2. Una password è considerata "imbarazzante" se:
 - È più corta di 6 caratteri, OPPURE
 - Contiene solo numeri consecutivi (es. "12345", "6789"), OPPURE
 - È nella lista delle password più usate: ["password", "123456", "qwerty", "admin"]



3. Il programma deve:

- Usare un ciclo `for` per scorrere le password
- Usare un ciclo `while` per controllare se ci sono numeri consecutivi
- Stampare per ogni password se è "sicura" o "IMBARAZZANTE" con un messaggio divertente
- Alla fine, mostrare quante password imbarazzanti hai trovato

Esempio di output:

```
Password "abc123XYZ!" -> ✓ Sicura (ma potevi fare di meglio)
Password "123456" -> ✗ IMBARAZZANTE! Anche tua nonna farebbe di meglio!
Password "admin" -> ✗ IMBARAZZANTE! Seriamente?!
...
Totale password imbarazzanti: 3 su 8. Dobbiamo parlare...
```




Esercizio 4

La Battaglia Navale dei Compiti

I professori hanno nascosto i compiti in una griglia 5x5.




Alcuni sono compiti facili () , altri sono verifiche a sorpresa () , e alcuni quadrati sono vuoti () .

Il tuo compito:

Simula una "battaglia navale" dove cerchi di trovare tutti i compiti evitando le verifiche!

Specifiche:

1. Crea una matrice 5x5 con:

- Almeno 5 "" (compiti normali)
- Almeno 3 "" (verifiche)
- Il resto "" (vuoto)



2. Crea una seconda matrice 5x5 di "?" che rappresenta cosa vede lo studente

3. Il gioco:

- Usa un ciclo `while` per far continuare il gioco finché chi sta giocando non trova tutti i compiti o prende 3 verifiche
- Ad ogni turno, lo studente sceglie coordinate (riga, colonna)
- Usa cicli `for` per aggiornare e mostrare la griglia dopo ogni scelta
- Tieni il punteggio: +10 punti per compito trovato, -20 per verifica!



4. Alla fine mostra:

- Quanti compiti ha trovato
- Quante verifiche ha preso
- Punteggio finale
- Un messaggio spiritoso in base al risultato

Esempio di gioco:



Griglia attuale:

```
? ? ? ? ?  
? ? ? ? ?  
? ? ? ? ?  
? ? ? ? ?  
? ? ? ? ?
```

Scegli riga (0-4): 2

Scegli colonna (0-4): 3

Hai trovato:  Compito! +10 punti


[continua...]

GAME OVER!

Compiti trovati: 4/5

Verifiche prese: 2/3

Punteggio: 0

Verdetto: "Potresti fare meglio... o forse no? 



Suggerimento: Per posizionare casualmente i simboli nella matrice, puoi usare la libreria `random`:

```
import random  
simbolo = random.choice(["📝", "💀", "🌊"])
```