

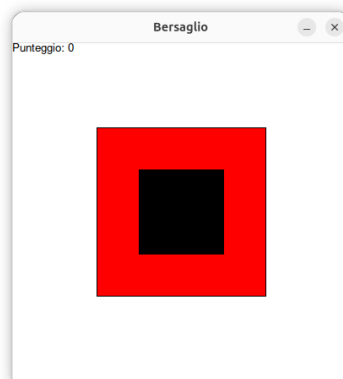
Programmazione e Algoritmi 1 - Prova pratica

A.A. 2025/26 — Compito del 12/6/2026

prof. Gianluca Amato

Esercizio 1 (8 punti)

Scrivere una programma che usa la libreria `ezgraphics` per disegnare un bersaglio composto da due quadrati centrati uno dentro l'altro. Il quadrato interno ha colore nero, quello esterno ha colore rosso. In alto nella finestra deve essere visualizzato un punteggio, inizialmente pari a zero. Ogni volta che l'utente clicca con il mouse all'interno del quadrato interno, il punteggio aumenta di 2 punti. Se invece l'utente clicca all'interno del quadrato esterno ma fuori da quello interno, il punteggio aumenta di 5 punti. In ogni altro caso, il punteggio non cambia. Il programma termina quando l'utente chiude la finestra. Potete vedere qui sotto un esempio di come dovrebbe apparire la finestra del programma.



Esercizio 2 (6 punti)

Scrivere una funzione `decompress` che prende come input una lista di coppie di interi. La funzione restituisce una lista in cui ogni coppia (n, m) è sostituita da una lista di n elementi, ciascuno dei quali è uguale ad m . Ad esempio:

- `decompress([(3, 5), (2, 7)])` restituisce `[5, 5, 5, 7, 7]`;
- `decompress([(1, 0), (4, 2), (2, 9)])` restituisce `[0, 2, 2, 2, 2, 9, 9]`.

Scrivere quindi la funzione che esegue l'operazione inversa, chiamata `compress`, che prende come input una lista di interi e restituisce una lista di coppie di interi, in cui ogni coppia (n, m) rappresenta una sequenza di n elementi uguali a m . Ad esempio:

- `compress([5, 5, 5, 7, 7])` restituisce `[(3, 5), (2, 7)]`;
- `compress([0, 2, 2, 2, 2, 9, 9])` restituisce `[(1, 0), (4, 2), (2, 9)]`.

A puro titolo di curiosità, queste due funzioni implementano un algoritmo di compressione e decompressione dei dati chiamato Run-Length Encoding.

Esercizio 3 (3 punti)

Scrivere due test nel framework `pytest` per verificare il corretto funzionamento delle funzioni `compress` e `decompress`. In particolare, si implementino due test:

- un test basato sugli esempi forniti sopra;
- un test per comprimere e decomprimere una lista vuota.

Programmazione e Algoritmi 1 - Prova scritta

A.A. 2025/26 — Compito del 12/6/2026

prof. Gianluca Amato

Esercizio 4 (7 punti)

Eseguire passo-passo il seguente codice Python, utilizzando l'apposito modulo:

```
1  def f(l1, l2):
2      if len(l1) == 0 or len(l2) == 0:
3          return []
4      elif l1[-1] != l2[-1]:
5          return []
6      else:
7          rest = f(l1[:-1], l2[:-1])
8          new = [l1[-1]]
9          return rest + new
10
11 print(f([1, 2, 3], [4, 3, 2, 3]))
```

Esercizio 5 (5 punti)

Scrivere una funzione `common_prefix(s1, s2)` che prende in input due stringhe e restituisce la più lunga sottostringa iniziale che è comune a entrambe. Ad esempio, `common_prefix("cane", "cavallo")` restituisce `"ca"`, mentre `common_prefix("cane", "zebra")` restituisce `""`.

Esercizio 6 (5 punti)

Descrivere in dettaglio il funzionamento degli algoritmi di ordinamento per selezione e per fusione. In quali casi è preferibile utilizzare uno rispetto all'altro? Illustrare il funzionamento di entrambi sul seguente esempio: [5, 2, 9, 1, 5, 6, 0, 10].