

Programmazione e Algoritmi 1 - Prova pratica

A.A. 2024/25 — Compito del 17/9/2025

prof. Gianluca Amato

Esercizio 1 (8 punti)

Scrivere una programma che legge un file di testo dal nome `input.txt`. Ogni riga del file è composta da due numeri interi separati da uno spazio, da interpretare come coordinate x ed y di un pixel in una finestra grafica. Tramite la libreria `ezgraphics`, per ogni riga del file disegnare un cerchio di raggio 10 pixel centrato alle coordinate indicate in quella riga. Il cerchio deve essere riempito alternativamente con i colori rosso e verde.

Ad esempio, se il file `input.txt` contiene i valori

```
50 70
90 200
10 10
```

il programma disegnerà tre cerchi: il primo rosso centrato alle coordinate (50,70), il secondo verde centrato alle coordinate (90,200) e il terzo di nuovo rosso alle coordinate (10,10).

Esercizio 2 (6 punti)

Scrivere una funzione `list_difference(a, b)` che prende come argomenti due liste e restituisce una **nuova** lista composta dagli elementi di a che non compaiono in b . L'ordine degli elementi in a viene preservato, così come eventuali ripetizioni. Ad esempio:

```
list_difference([1, 10, 1, "a", 0], ["b", 3, 10])
```

restituisce la lista `[1, 1, "a", 0]`.

Scrivere quindi una funzione `list_difference_inplace(a, b)` che si comporta allo stesso modo di `list_difference(a, b)`, con la differenza che invece di restituire una nuova lista, modifica direttamente la lista a e non restituisce nulla.

Esercizio 3 (3 punti)

Utilizzare il framework `pytest` per collaudare la funzione `list_difference`. Si verifichino, in particolare, i seguenti casi:

- l'esempio mostrato nell'esercizio qui sopra;
- il caso in cui il secondo elemento è una lista vuota e il primo elemento è una lista casuale di 100 numeri da 1 a 20.

Programmazione e Algoritmi 1 - Prova scritta

A.A. 2024/25 — Compito del 17/9/2025

prof. Gianluca Amato

Esercizio 4 (8 punti)

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

```
1  def flatten(l):
2      return flatten_aux(l, [])
3
4  def flatten_aux(l,res):
5      if len(l) == 0:
6          return res
7      else:
8          tmp = flatten_aux(l[1:], res + l[0])
9          return tmp
10
11 l = [[1,2] , [3], ["a", "c"]]
12 print(flatten(l))
```

Esercizio 5 (4 punti)

Descrivere in dettaglio il funzionamento degli algoritmi di ricerca sequenziale e di ricerca binaria, ed esemplificarne il funzionamento con la ricerca del numero 45 nella lista [1, 5, 5, 10, 30, 31, 45, 56, 98].

Esercizio 6 (5 punti)

Scrivere la funzione `matrix_sum(a,b)` che prende come parametri due matrici (implementate come liste di liste) che hanno lo stesso numero di righe e colonne e restituisce una nuova tabella, sempre con lo stesso numero di righe e colonne, i cui elementi sono ottenuti sommando i rispettivi elementi di a e b . Ad esempio, se

$a = \begin{bmatrix} 10, & 20, & 30 \\ 40, & 50, & 60 \end{bmatrix}$	$b = \begin{bmatrix} 1, & 2, & 3 \\ 4, & 5, & 6 \end{bmatrix}$
--	--

la funzione restituirà la tabella

```
[
    [11, 22, 33],
    [44, 55, 66]
]
```