

Programmazione e Algoritmi 1 - Prova pratica

A.A. 2025/26 — Compito del 17/7/2026

prof. Gianluca Amato

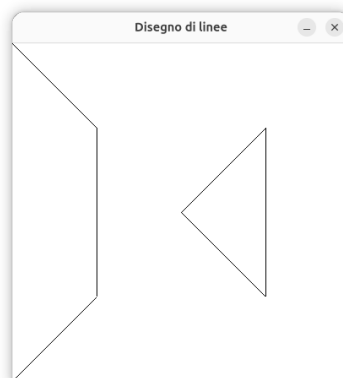
Esercizio 1 (8 punti)

Scrivere un programma che legge un file di nome `istruzioni.txt`. Ogni riga del file contiene una sequenza di numeri interi non negativi che rappresentano, alternativamente, le coordinate x e y di punti nel piano, secondo il sistema di coordinate usato da `ezgraphics`. Si può assumere che ogni riga contenga un numero pari di valori e le coordinate di almeno due punti.

Per ciascuna riga, il programma deve disegnare una linea spezzata che unisca i punti nell'ordine in cui compaiono. Ogni riga descrive una nuova spezzata, che non deve essere collegata a quella descritta dalla riga precedente. Ad esempio, se il file contiene

```
0 0 100 100 100 300 0 400
200 200 300 100 300 300 200 200
```

il programma deve disegnare due linee spezzate: la prima unisce i punti $(0,0)$, $(100,100)$, $(100,300)$ e $(0,400)$; la seconda unisce i punti $(200,200)$, $(300,100)$, $(300,300)$ e $(200,200)$. Il risultato è il seguente:



Esercizio 2 (6 punti)

Una lista rappresenta le quote raggiunte durante un'escursione, misurate ogni 15 minuti. Ad esempio,

```
[520, 548, 575, 560, 590, 615, 608, 602, 630, 605]
```

rappresenta un'escursione in cui si parte da 520 metri, si sale prima a 548 metri e poi a 575 metri, si scende a 560 metri, si sale a 590 metri e così via.

Scrivere una funzione `quote_estreme(l)` che riceve una lista di questo tipo e restituisce una nuova lista contenente soltanto le quote estreme. Il primo e l'ultimo elemento della lista sono sempre considerati estremi. Un elemento interno è invece un estremo se è strettamente maggiore di entrambi gli elementi adiacenti (massimo locale) oppure strettamente minore di entrambi (minimo locale). Per la lista dell'esempio, la funzione deve restituire `[520, 575, 560, 615, 602, 630, 605]`.

Esercizio 3 (3 punti)

Scrivere alcuni test con il framework `pytest` per verificare il corretto funzionamento della funzione `quote_estreme`. In particolare, implementare i seguenti test:

- un test basato sull'esempio dell'esercizio 2;
- un test per una lista con valori strettamente crescenti;
- un test per la lista vuota;
- un test per una lista con un solo elemento.

Programmazione e Algoritmi 1 - Prova scritta

A.A. 2025/26 — Compito del 17/7/2026

prof. Gianluca Amato

Esercizio 4 (7 punti)

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

```
1  def f(x, l):
2      if l == []:
3          return 0
4      if l[0] == x:
5          return f(x, l[1:])
6      else:
7          rest = f(x, l[1:])
8          return l[0] + rest
9
10 print(f(3, [4, 3, 2, 3]))
```

Esercizio 5 (5 punti)

Scrivere una funzione `scelta_multipla` che riceva tre liste: una lista di valori booleani e due liste di numeri interi. La funzione deve restituire una nuova lista di interi costruita nel modo seguente: per ogni posizione della lista di booleani, se il valore è `True`, si inserisce nella nuova lista l'elemento nella stessa posizione della prima lista di interi; altrimenti, si inserisce l'elemento corrispondente della seconda lista di interi. Ad esempio,

```
scelta_multipla([True, False, True], [1, 2, 3], [4, 5, 6])
```

deve restituire la lista `[1, 5, 3]`. Si può assumere che entrambe le liste di interi abbiano lunghezza almeno pari a quella della lista di booleani.

Esercizio 6 (5 punti)

Descrivere la struttura della memoria, in termini di stack e heap, al termine dell'esecuzione, rispettivamente, delle righe 3, 5 e 6 del seguente codice Python:

```
1  l = [1, 2, 3]
2  n = [1, 2, 3]
3  m = l
4  l[0] = 99
5  l.append(4)
6  l = l + [5]
```