

# Programmazione e Algoritmi 1

A.A. 2023/24 — Compito dell'8 luglio 2024

prof. Gianluca Amato

Gli esercizi di programmazione saranno valutati sulla base della correttezza, efficienza e comprensibilità della soluzione proposta. In generale, **se volete usare una funzione o un metodo che non è stato presentato a lezione, chiedete prima al docente se è consentito.**

## Esercizio 1 (5 punti)

Descrivere l'algoritmo di ordinamento per fusione (merge sort): input, output, funzionamento, vantaggi e svantaggi rispetto ad altri algoritmi che risolvono lo stesso problema. Successivamente, illustrarne il funzionamento applicandolo alla lista ["ciricio", "a", "tutti", "sono", "arale"].

## Esercizio 2 (8 punti)

Si consideri il seguente codice Python, e se ne scriva la traccia di esecuzione negli appositi moduli.

```
1 def mistero(l1, l2):
2     for i in range(len(l1)):
3         l1[i] += mistero2(l2, i)
4
5 def mistero2(l, i):
6     x = 0
7     while i >= 0:
8         x += l[i]
9         i -= 1
10    return x
11
12 data1 = [1, 2]
13 data2 = [2, 3, 4, 5]
14 mistero(data1, data2)
15 print(data1)
```

## Esercizio 3 (5 punti)

Scrivere una funzione `estrai_maschera(data, maschera)`, dove `data` è una lista di oggetti qualunque e `maschera` è una lista di booleani. La funzione, senza modificare i parametri in input, restituisce una nuova lista che contiene gli elementi di `data` che corrispondono a valori `True` in `maschera`. Ad esempio `estrai_maschera(["a", "z", 5, 7], [True, False, False, True])` restituisce `["a", 7]`.

Se la lunghezza di `maschera` è minore di quella di `data`, gli elementi mancanti vengono considerati uguali a `False`. Se la lunghezza di `maschera` è maggiore di quella di `data`, gli elementi in più vengono semplicemente ignorati.

## Esercizio 4 (5 punti)

Scrivere due test nel framework `pytest` per verificare il corretto funzionamento di `estrai_maschera`. In particolare, si implementino due test.

- un test per le seguenti chiamate di funzione:
  - `estrai_maschera(["a", "z", 5, 7], [True, False, False, True])`
  - `estrai_maschera(["a", "z", 5, 7], [True])`
  - `estrai_maschera(["a", "z"], [False, True, True])`
- un test per il caso in cui la lista `data` è generata casualmente (sia la lunghezza che il contenuto sono casuali) e `maschera` è sempre `[True]`; questo test deve essere ripetuto 100 volte.

## Esercizio 5 (5 punti)

Scrivere un programma che legge il file `input.txt` composto da due righe di testo: la prima riga contiene una sequenza di numeri interi separati da spazi, la seconda riga contiene una sequenza di 0 ed 1, dove 0 rappresenta il booleano falso e 1 il booleano vero. Il programma deve invocare la funzione `estrai_maschera` con le due liste lette dal file, e scrivere il risultato su `output.txt`, un numero per riga. Ad esempio, se il file `input.txt` contiene le seguenti righe

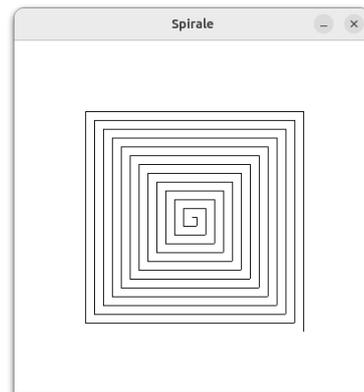
```
1 2 3 4 5
0 1 1 0 0
```

il file `output.txt` conterrà

```
2
3
```

## Esercizio 6 (5 punti)

Scrivere una programma che visualizza una spirale quadrata usando la libreria `ezgraphics`. Il programma chiede all'utente il numero di linee di cui deve essere composta la spirale, quindi la visualizza su una finestra grafica. Questo è un esempio di output per una spirale di 5 linee e una di 50.



Questo è un ingrandimento della spirale di 5 linee, per mostrarne meglio i dettagli. Notare che c'è uno spigolo bianco nell'angolo in basso a destra. È un bug di `ezgraphics`, ignoratelo.

