

# Laboratorio 7

## Elementi di Informatica e Programmazione

### Esercizio 1: Palindromi ricorsivi

Scrivere il programma `recursiveIsPalindrome.py` che verifichi se una stringa, fornita dall'utente, è un palindromo oppure no. Si ricorda che una stringa è un palindromo se è composta da una sequenza di caratteri (anche non alfabetici) che possa essere letta allo stesso identico modo anche al contrario (es. "radar", "anna", "inni", "xyz%u%zyx").

Il programma non deve utilizzare costrutti iterativi (cioè non deve avere nessun tipo di ciclo). Verificare il corretto funzionamento del programma con:

- una stringa di lunghezza pari che sia un palindromo
- una stringa di lunghezza dispari che sia un palindromo
- una stringa di lunghezza pari che non sia un palindromo
- una stringa di lunghezza dispari che non sia un palindromo
- una stringa di lunghezza unitaria (che è ovviamente un palindromo)
- una stringa di lunghezza zero (che è ragionevole ritenere sia un palindromo, dato che niente la rende "non un palindromo"...) )

### Esercizio 2: conteggio della frequenza delle lettere

Scrivere il programma `letterfreq.py` che calcoli la frequenza delle singole lettere dell'alfabeto all'interno di un file di testo (un'informazione utile nell'ambito della crittografia e della compressione):

- chiede all'utente il nome del file da elaborare
- se l'utente scrive la stringa vuota, si usi il testo American Fairy Tales by L. Frank Baum, reperibile all'indirizzo <http://www.gutenberg.org/cache/epub/4357/pg4357.txt>, che potete salvare in un file locale dal nome che preferite
- altrimenti, il file si trova nel file system locale (e la stringa può essere il solo nome, oppure un percorso relativo o assoluto)

- nel testo da elaborare, dopo averlo convertito in maiuscolo, conta le occorrenze delle singole lettere (ignorando tutti i caratteri che non sono lettere)
- visualizza una tabella di due colonne: nella prima colonna vanno riportate le lettere dell'alfabeto inglese, in ordine lessicografico (cioè, in questo caso, alfabetico), nella seconda colonna la percentuale di lettere del testo che sono uguali a quella indicata in tale riga, riportando i dati con due cifre decimali nella parte frazionaria, allineati a destra

**Esempio:**

```
A 1.32%
B 12.20%
C 0.01%
D 3.00%
...
```

Osservate come solitamente nei testi in lingua inglese la lettera che ricorre più frequentemente sia la e, mentre nei testi in lingua italiana la “competizione” sia accesa tra le lettere e e a.

### Esercizio 3: triangoli

Progettare il modulo `triangle.py` che contenga la definizione della classe `Triangle`, i cui esemplari rappresentino triangoli geometrici nel piano cartesiano, oltre a codice di collaudo.

Per prima cosa definire la classe `Point` che descriva un punto nel piano cartesiano: il costruttore riceve le coordinate `x` e `y` del punto; il metodo di esemplare `x()` restituisce la coordinata `x` e il metodo di esemplare `y()` restituisce la coordinata `y`, mentre il metodo di esemplare `xy` restituisce una tupla contenente, nell'ordine, la coordinata `x` e la coordinata `y`; il metodo di classe `delta(p1, p2)` calcola e restituisce la distanza tra il punto `p1` e il punto `p2`.

La classe `Triangle` ha un costruttore che richiede tre punti (cioè tre esemplari di `Point`) e solleva l'eccezione `ValueError` se tali punti non costituiscono un triangolo (ricordando che tre punti definiscono un triangolo se e solo se ciascuno dei tre segmenti da essi definiti è minore della somma degli altri due). Per decidere quali siano le migliori informazioni di stato per un tale oggetto, è bene analizzare la sua interfaccia pubblica, così definita:

- il metodo `area()` restituisce l'area del triangolo (può essere utile ricordare la [formula di Erone](#))
- il metodo `height()` restituisce l'altezza relativa al lato maggiore
- il metodo `isScalene()` restituisce `True` se e solo se il triangolo è scaleno
- il metodo `isIsosceles()` restituisce `True` se e solo se il triangolo è isoscele
- il metodo `isEquilateral()` restituisce `True` se e solo se il triangolo è equilatero (ovviamente un triangolo equilatero è anche isoscele)

- il metodo `isRight()` restituisce `True` se e solo se il triangolo è rettangolo (ovviamente un triangolo rettangolo può anche essere isoscele o scaleno)