

## Compito d'esame -- 21 gennaio 2019 -- Compito A

### Istruzioni (leggere attentamente)

**Nota importante:** la mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione dell'esercitazione.

**Registrazione dei dati dello studente:** PRIMA DI INIZIARE, eseguite (con un doppio click sull'icona) il programma `RegistraStudente` che si trova nella cartella `Esame`. Inserite (separatamente) *Numero di Matricola*, *Cognome* e *Nome* seguendo le istruzioni che compaiono sul terminale, e confermate i dati che avete inserito. Il programma genera il file `studente.txt` che contiene Matricola, Cognome e Nome su tre righe separate (nell'ordine indicato). Il file `studente.txt` non deve essere modificato manualmente. Verificate che i dati nel file `studente.txt` siano corretti. In caso di errore potete rieseguire il programma `RegistraStudente`.

**Svolgimento degli esercizi:** Per ogni esercizio avete una cartella `EsercN` che contiene un file dal nome `A_ExN.py` (dove **N** è il numero dell'esercizio) con lo scheletro della soluzione. Questo file incorpora un codice di test che proverà la vostra soluzione per un certo numero di possibili dati in input. Aprite il file con Spyder e modificate SOLO il contenuto della funzione. Eseguendo il file `.py` si otterrà il responso dei test sulla console. Non spostate i file dalla loro posizione e non create nuovi file. **Si noti che per la correzione verranno usati insieme di dati di test diversi.**

E' possibile consultare la documentazione ufficiale del linguaggio Python, ma **non è possibile usare libri o appunti**. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

**Al termine** dello svolgimento del compito d'esame **NON è necessario effettuare una procedura di consegna** dell'elaborato. Le vostre soluzioni saranno copiate direttamente dalla cartella `Esame` della macchina su cui avete lavorato. Quindi, per concludere la prova, è sufficiente chiudere il programma Spyder e lasciare la vostra postazione.

### Esercizi

- **A\_Ex1(s1,s2)** Scrivere una funzione che riceve in ingresso due stringhe **s1** e **s2**, e restituisce il prefisso più lungo comune alle due stringhe, cioè la sottostringa comune più lunga posta all'inizio di **s1** ed **s2**. Ad esempio, se **s1** = "amaca" e **s2** = "amato", la funzione deve restituire la stringa "ama". Se invece **s1** = "pippo" e **s2** = "minnie" la funzione deve restituire la stringa vuota "". Se almeno una delle due stringhe è vuota la funzione deve restituire "".
- **A\_Ex2 (m)** In una matrice **m** di numeri interi, una posizione di indici **i,j** si chiama posizione di sella se l'intero in quella posizione è (strettamente) più piccolo dei due numeri a lui adiacenti a sinistra e destra e (strettamente) più grande dei due numeri a lui adiacenti sopra e sotto o VICEVERSA. Deve cioè essere il numero più piccolo in una direzione (orizzontale o verticale) ed il più grande nell'altra. Per semplicità potete assumere che le posizioni sui bordi NON possono essere posizioni di sella. Scrivere una funzione che riceve in ingresso una matrice **m**, rappresentata come lista di liste, e restituisca la lista ORDINATA in modo crescente degli interi che occupano le posizioni di sella presenti in **m**. Ad esempio, se la matrice è:

3	2	1	5
2	10	<b>3</b>	4
4	5	2	4

Allora la funzione deve restituire [3], poiché solo l'elemento in grassetto è in una posizione di sella.

- **A\_Ex3(l)** Scrivere una funzione che riceve in ingresso una lista **l** contenente stringhe, e restituisce la lista ottenuta eliminando da **l** tutte le stringhe contenenti il massimo numero di lettere maiuscole. Ad esempio, se **l** = ["piPPo", "pippo", "PLuto", "Pippo"], la funzione deve restituire ["pippo", "Pippo"], essendo il massimo numero di lettere maiuscole contenuto in una stringa di **l** pari a due.
- **A\_Ex4 (file)** Scrivere una funzione che riceve in ingresso le informazioni sui quadri che vengono comprati e venduti da collezionisti. Più precisamente, le informazioni sono in un file **file** in formato csv contenente l'elenco delle transazioni, nel formato:

```
Acquirente,Venditore,Pittore
```

Cioè il primo campo contiene il nome dell'acquirente del quadro, il secondo specifica il nome del venditore ed il terzo specifica il nome del pittore. All'inizio tutti i collezionisti hanno 0 quadri e la prima volta che un quadro compare nel file ovviamente non potrà esserci un proprietario precedente e quindi nella riga nel campo del Venditore ci sarà una stringa vuota (' '). Inoltre, ovviamente un venditore non può vendere un quadro di un pittore di cui possiede 0 quadri, in questo caso le righe corrispondenti VANNO IGNORATE. La funzione deve restituire una lista, ORDINATA alfabeticamente, con il nome del collezionista (o dei collezionisti) che ha (hanno) il maggior numero di quadri. Ad esempio, se il file è:

```
Acquirente,Venditore,Pittore
Mario,,Leonardo
Paolo,,Picasso
Paolo,,Van_Gogh
Gianni,,Van_Gogh
Gianni,Paolo,Picasso
Mario,Gianni,Van_Gogh
```

Allora il risultato sarà la lista ['Mario']