

## Compito d'esame -- 19 febbraio 2019 -- Compito A

### Istruzioni (leggere attentamente)

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

**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(l)** Scrivere una funzione che prende in ingresso una lista **l** contenente stringhe e restituisce la lista ORDINATA in modo crescente e SENZA RIPETIZIONI delle lettere estratte dagli elementi contenuti in **l** seguendo la seguente regola: se *s* è una stringa che occupa in **l** una posizione pari, allora si estraggono da *s* le lettere in posizione pari, se *s* occupa una posizione dispari in **l**, si estraggono da *s* le lettere in posizione dispari. Le posizioni si contano a partire dallo 0. Ad esempio, se `l=["mamma", "sola", "casa"]` la funzione deve restituire `["a", "c", "m", "o", "s"]`.
- **A\_Ex2 (m)** Scrivere una funzione che prende in ingresso una matrice di numeri interi **m** non vuota, rappresentata come lista di liste, e restituisce una lista contenente la somma degli elementi delle varie colonne (secondo l'ordine delle colonne) se i valori della matrice sono tutti diversi fra loro, una lista vuota altrimenti. Ad esempio, se **m** è la matrice

2	7	6
9	5	1
4	10	11

allora la funzione deve restituire `[15, 22, 18]` in quanto gli elementi sono tutti diversi tra loro e le somme degli elementi sulle colonne sono in ordine 15, 22 e 18.

- **A\_Ex3(file)** Scrivere una funzione che prende in ingresso un file di testo e restituisce un dizionario con chiave un carattere alfabetico e valore la lista ORDINATA dei numeri di riga (contate a partire

da 1) in cui quel carattere è quello più frequente, NON considerando nell'analisi le vocali ("aeiou" sia maiuscole che minuscole) e gli spazi. Ad esempio, se il file contiene:

In Sicilia ad AcI Trezza,  
un paesino nei pressi di Catania, vivono i Toscano,  
soprannominati i Malavoglia.

La funzione deve restituire il dizionario `{ 'c': [1], 'z': [1], 'n': [2, 3] }`. Notate che se un carattere non è mai uno dei caratteri più frequenti in una riga allora NON deve comparire nel dizionario. Lettere maiuscole e minuscole vanno considerate distinte. Notate che per ogni riga vi possono essere più caratteri che compaiono il numero massimo di volte.

- **A\_Ex4(file,n)** Scrivere una funzione che prende in ingresso un numero intero **n** ed un file csv **file** rappresentante tutti i voli in partenza ed in arrivo dall'aeroporto della città di Anycity nel seguente formato:

**Città\_di\_partenza,Città\_di\_arrivo,Ora\_partenza,Ora\_arrivo**

Potete assumere che:

- una tra Città\_di\_partenza e Città\_di\_arrivo sia obbligatoriamente Anycity,
- Ora\_partenza e Ora\_arrivo siano numeri interi compresi tra 0 e 23 (non dovete quindi preoccuparvi dei minuti),
- per un volo Ora\_partenza sia sempre minore di Ora\_arrivo,
- tutti i voli rientrano entro al massimo le 23 (cioè non ci sono voli che partono prima ed arrivano dopo mezzanotte).

La vostra funzione deve costruire la lista ORDINATA alfabeticamente di tutte le città che è possibile visitare in giornata partendo da Anycity (cioè facendo andata e ritorno nello stesso giorno) e in cui avete a disposizione almeno **n** ore ( $\geq n$ ) per visitare la città. Ad esempio, se il file contiene:

Città\_di\_partenza,Città\_di\_arrivo,Ora\_partenza,Ora\_arrivo  
Anycity,Lisbona,9,11  
Anycity,Londra,7,9  
Lisbona,Anycity,10,12  
Londra,Anycity,12,14  
Anycity,Parigi,8,9  
Parigi,Anycity,17,18

ed **n** vale 3, allora la funzione deve restituire la lista `['Londra', 'Parigi']`, poiché è possibile visitare Londra e Parigi in giornata avendo almeno 3 ore tra l'arrivo del volo di andata e la partenza del volo di ritorno.