

Sapienza Università di Roma
Corso di Laurea in Ingegneria Informatica e Automatica
A.A. 2018-19
Compito d'esame -- 14 giugno 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 i 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 prende in ingresso due stringhe, **s1** ed **s2**, e restituisce `True` se **s1** è una stringa *bifronte* di **s2**, cioè è uguale all'inverso di **s2**, altrimenti restituisce `False`. Ad esempio, se **s1**='acetone' ed **s2**='enoteca' la funzione deve restituire `True`, se invece **s1**='aceto' ed **s2**='teca' la funzione deve restituire `False`.
- **A_Ex2 (M,I1,I2)** Scrivere una funzione che prende in ingresso una matrice **M** (rappresentata come lista di liste) di dimensione $m \times n$, tale che $m > 1$ ed $n > 1$ (cioè si può assumere che **M** abbia almeno due righe ed almeno due colonne), una lista **I1** (anche vuota) di interi compresi fra 1 ed m , ed una lista **I2** (anche vuota) di interi compresi fra 1 ed n . La funzione restituisce una matrice ottenuta da **M** eliminando le righe indicate in **I1** e le colonne indicate in **I2**. Si assuma che **I1** contenga al massimo $m-1$ elementi, and **I2** contenga al massimo $n-1$ elementi (in tal modo la matrice restituita non sarà mai vuota, perché non si può chiedere di eliminare tutte le righe e/o tutte le colonne). Ad esempio, se **M** è la matrice

2	7	6
9	5	1
4	10	11
3	4	4

I1=[1,3] ed **I2**=[2], allora la funzione deve restituire la matrice

9	1
3	4

in quanto la lista **l1** richiede di eliminare la prima e la terza riga, mentre la lista **l2** richiede di eliminare la seconda colonna.

Nota: Si ricorda che in una lista di liste Python che rappresenta una matrice le righe e le colonne sono indicizzate a partire dallo 0. Nelle liste **l1** ed **l2**, invece, righe e colonne sono indicate nel modo classico, e cioè vengono contate a partire da 1.

- **A_Ex3(file,n)** Scrivere una funzione che prende in ingresso un file di testo ed un numero *n* e restituisce un dizionario con chiave un carattere alfabetico ('a'-'z') e valore la lista ORDINATA dei numeri di riga (contate a partire da 1) in cui quel carattere compare almeno *n* volte (va contato sia maiuscolo che minuscolo). Ad esempio, se *n* vale 3 ed 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 {'a': [1, 2, 3], 'i': [1, 2, 3], 'e': [2], 'n': [2, 3], 'o': [2, 3], 's': [2]}.

- **A_Ex4(file)** Scrivere una funzione che prende in ingresso il nome di un file csv **file** contenente i tempi fatti nelle varie maratone corse nell'anno da un gruppo di *n* (non noto a priori) maratoneti nel seguente formato:

Nome1,Nome2,.....,Nomen
tempo1,tempo2,.....,tempon

Dove tempo_i è il tempo in minuti (intero) per completare la maratona dell'atleta Nome_i. Il numero 0 significa che il maratoneta non ha corso quella maratona. La vostra funzione deve costruire un dizionario con chiave in nome del maratoneta e valore la lista di 4 valori [miglior tempo, numero maratone disputate, numero maratone vinte, numero maratone arrivato ultimo].

Potete assumere che:

1. Ogni maratoneta ha corso almeno una maratona nell'anno.
2. Per ogni maratona ci sono almeno 2 concorrenti ed i tempi dei partecipanti sono tutti diversi (non ci sono arrivi ex-aequo)

Ad esempio, se il file contiene:

Mario,Paolo,Gianna,Giulia
150,132,0,160
135,141,130,0
130,0,0,121

allora la funzione deve restituire il dizionario {'Mario': [130, 3, 0, 1], 'Paolo': [132, 2, 1, 1], 'Gianna': [130, 1, 1, 0], 'Giulia': [121, 2, 1, 1]}