

Compito d'esame -- 18 febbraio 2021 -- Compito A

Istruzioni (leggere attentamente)

Nota importante: la mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione della prova d'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`.

Tempo a disposizione: 1 ora e 30 minuti

Per risolvere gli esercizi in modo che possano essere successivamente corretti è **necessario scrivere la soluzione di ogni esercizio nel file .py relativo**, che trovate nella cartella dell'esame (ad esempio, per l'esercizio 1 scrivete il vostro programma nel file `Ex1.py`, per l'esercizio 2, nel file `Ex2.py`, e così via). Notate che ogni file incorpora del codice python per eseguire alcuni test sulla funzione. **NON** modificate questo codice, ma **SCRIVETE SOLO il contenuto della funzione**. 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 (<file:///Library/Frameworks/Python.framework/Versions/3.9/Resources/English.lproj/Documentation/index.html>),

ma **non è possibile usare libri o appunti**. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

In ogni esercizio, se non diversamente richiesto, potete sempre assumere che gli input forniti siano coerenti con la traccia (ad esempio, se l'esercizio chiede di dare in input alla funzione una lista non vuota di stringhe, potete sempre assumere l'input sia in tale forma e non è necessario nel codice effettuare controlli per gestire casi diversi da questo, considerando, ad esempio, il caso di lista vuota).

Per gli esercizi relativi a lettura da file, la stringa in input che identifica il file è sempre comprensiva anche della sua estensione e il file risiede sempre nella stessa directory dell'esercizio.

Esercizi

- **Ex1(I)** Una stringa si dice palindroma se leggendola da destra a sinistra o viceversa non cambia. Chiamiamo una stringa palindroma di ordine n se i primi n caratteri letti da sinistra a destra sono uguali agli ultimi n caratteri letti da destra a sinistra. Ad esempio, la stringa 'assosa' è una palindroma di ordine 2, mentre 'anna' è una palindroma di ordine 4. Scrivere una funzione che riceve in ingresso una lista di stringhe l e restituisce l'insieme delle stringhe di l che sono palindrome di ordine massimo. Ad esempio, se la lista l vale `['avola', 'ava', 'plutoulp', 'corda']`, la funzione deve restituire l'insieme `{ 'ava', 'plutoulp' }` (come al solito negli insiemi l'ordine non è rilevante).

- **Ex2 (I)** Scrivere una funzione che riceve in ingresso una lista **I** contenente tuple. Le tuple hanno lunghezza variabile ma sempre pari almeno a 2, e tutte hanno la seguente struttura: il primo elemento è una stringa che corrisponde al nome di un atleta, mentre gli elementi successivi della tupla sono indicatori di performance (interi positivi, negativi o nulli). Ogni tupla si riferisce ad un atleta diverso (quindi non ci saranno mai due tuple con il primo elemento uguale). La funzione deve restituire un insieme contenente i nomi degli atleti che compaiono in **I** e che hanno il massimo indicatore di performance complessivo, che corrisponde, per ciascun atleta, alla somma di tutti i suoi indicatori di performance. Ad esempio, se `l=[('marco',-3,0,1),('aldo',2,3),('maria',-1,1,2,1,2)]` allora la funzione deve restituire `{'aldo','maria'}` in quanto 'marco' ha indicatore di performance complessivo pari a -2, 'aldo' e 'maria' hanno indicatore di performance complessivo pari a 5 (che è il massimo conseguito da un atleta in **I**). Se la lista **I** in input è vuota, la funzione deve restituire un insieme vuoto.

- **Ex3(file1,file2)** Scrivere una funzione che prende in ingresso un file di testo **file1** in formato csv che contiene le informazioni sui computer portatili disponibili in un negozio ed un secondo file **file2** che contiene, in ordine di arrivo, le richieste di acquisto. Il file dei portatili disponibili **file1** ha il seguente formato:

`NomePortatile,Disponibilità,Costo,caratteristica_1,...,carateristica_n`

dove `NomePortatile` è il nome del modello, `Disponibilità` denota quante unità sono disponibili, `Costo` è il costo in euro (intero) e `caratteristica_1,...,carateristica_n` sono caratteristiche del prodotto, come 'schermogrande', 'cpuveloce', 'discosd', eccetera. Il file **file2** delle richieste di acquisto ha il seguente formato:

`NomeCliente,prezzoMassimo,caratteristicaR_1,..., caratteristicaR_n`

Cioè per ogni riga contiene il nome (univoco) del cliente, seguito dal prezzo massimo che è disposto a spendere, seguito dalle caratteristiche richieste dal cliente. La funzione deve trovare il più economico portatile che costi meno del `prezzoMassimo` e che possiede tutte le caratteristiche richieste, e restituire un dizionario in cui le chiavi sono i nomi dei clienti che hanno completato un acquisto (potete assumere che ogni cliente effettui uno ed un solo ordine) e per ogni chiave il valore associato sia una tupla con il `NomePortatile` acquistato ed il costo pagato. Ovviamente, quando un portatile viene acquistato la sua disponibilità scende di 1. Potete assumere che non ci siano mai due portatili con lo stesso prezzo, ma notate che le caratteristiche NON sono ordinate e quindi possono essere ogni volta in ordine diverso. Ad esempio, se il file **file1** contiene:

```
Apple,1,549,leggero,schermogrande,cpuveloce
Acer,2,349,leggero,discosd
Samsung,1,399,leggero,cpuveloce
Dell,1,369,leggero,discosd
```

Ed il **file2** contiene:

```
Giorgio,500,cpuveloce
Paola,400,leggero,discosd
Francesca,450,leggero
```

Allora la funzione deve restituire `{'Giorgio': ('Samsung', 399), 'Paola': ('Acer', 349), 'Francesca': ('Acer', 349)}`.