

Compito d'esame -- 20 gennaio 2023 -- Compito B

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 45 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'esercitazione (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 di 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)** Scrivere una funzione che riceve in ingresso una lista `l` contenente stringhe di lunghezza 2 e caratteri minuscoli e restituisce la lunghezza della più lunga sequenza di stringhe con la proprietà che il secondo numero carattere della stringa a sinistra è uguale al primo carattere della stringa a destra. Ad esempio, se `l=["ca", "ab", "dd", "df", "fa", "ad", "kl", "no", "aa"]`, la funzione deve restituire 4 perché quattro è la lunghezza della più lunga sequenza in `l` avente la caratteristica richiesta (in grassetto) e non c'è in `l` una sequenza di questo tipo che abbia lunghezza 5 o superiore. Se la lista è vuota, la funzione deve restituire 0, se contiene una sola stringa deve restituire 1.

- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **file**, e analizzi gli indirizzi e i telefoni presenti in esso. Si assuma che gli indirizzi e telefoni corretti abbiano il seguente formato:
Nomi.Cognome.Telefono(Città) e che siano separati da spazi bianchi e ritorni a capo

Dove Nomi è una sequenza di uno o più nomi (sequenza di caratteri alfabetici) separati da virgole, Cognome è una sequenza di caratteri alfabetici, telefono una sequenza di caratteri numerici e Città è una sequenza di caratteri alfabetici. Più precisamente, bisogna costruire un dizionario con chiave i vari Nomi di Città presenti negli indirizzi corretti del file e valore una lista di 3 numeri che indicano il numero di indirizzi corretti di quella città che sono composti da 1, 2 o più di 2 nomi. Se ad esempio il file è il seguente

Mario.Rossi.355975(Roma) Maria,Francesca.Neri.44473(Roma) Giovanni(Roma)
Marco.Bianchi.555575(Milano)

la funzione deve restituire {'Roma': [1, 1, 0], 'Milano': [1, 0, 0]}.

- **Ex3(file,l)** Scrivere una funzione che prende in ingresso un file di testo in formato csv **file** ed una lista **l**. Il file **file** contenente le informazioni sui dei pacchi di grosso peso che devono essere spediti e e la lista **l** i servizi offerti dalla ditta attuale. Il file csv **file** ha il seguente formato:

IdPacco, SpedizioneRichiesta, Peso

dove IdPacco è il codice del pacco da spedire, SpedizioneRichiesta è il tipo di spedizione e Peso è il peso del pacco.

La lista **l** ha il seguente formato: SpedizioneOfferta, NumeroCorrieri, PesoDisponibile

Dove SpedizioneOfferta è l'unica tipologia di spedizione offerta dalla ditta, NumeroCorrieri sono il numero di corrieri che la ditta ha a disposizione e PesoDisponibile è peso di carico disponibile complessivamente per la ditta. Ogni pacco deve essere associato a un corriere secondo la seguente regola: I pacchi vanno assegnati al corriere seguendo l'ordine in cui si trovano nel file, devono essere assegnati ad un corriere solo se il tipo di spedizione è adatta ed un peso di carico sufficiente (ovviamente dopo che un pacco è stato assegnato a un corriere il peso di carico disponibile della ditta scende). Se nessun camion soddisfa i criteri vuol dire che il pacco non può essere spedito.

La funzione deve restituire un dizionario con chiave l'ID di un pacco e valore il numero del corriere (da 1 a NumeroCorrieri) a cui è stato assegnato. Se nessun corriere è disponibile per quel pacco allora il valore sarà 'Nessuno'. Ad esempio, se il file **file** contiene:

pacco1, Europea, 500
pacco2, Nazionale, 200
pacco3, Nazionale, 500
pacco4, Mondiale, 200
pacco5, Nazionale, 100

e la lista **l** vale [Nazionale, 2, 800], allora la funzione deve restituire {'pacco1': 'Nessuno', 'pacco2': 1, 'pacco3': 2, 'pacco4': 'Nessuno', 'pacco5': 'Nessuno'}, poiché pacco1 e pacco4 hanno bisogno di una spedizione non disponibile per la ditta e pacco5 non ha più corrieri disponibili per essere consegnato.

Domande veroFalso

Ex4 Il file Ex4.py contiene la funzione veroFalso() che stampa 8 domande sullo schermo. La funzione deve essere modificata cambiando il valore del return, elencando le lettere delle domande che ritenete essere vere. Ad esempio, se ritenete che le domande B e C sono vere il return deve essere modificato in

```
return 'BC'
```