

Esercitazione Python n. 8 -- 26 Novembre 2024

Obiettivo dell'esercitazione è esercitarsi con l'accesso ai file.

In laboratorio, per avviare la macchina virtuale, dovete selezionarla all'interno della schermata di VirtualBox (schermata visualizzata all'accensione del PC), cliccando sull'opzione **LXLE-BIAR-4.6**. Una volta avviata la macchina virtuale, PRIMA DI INIZIARE, eseguite il programma `REGISTRastudente.py` che si trova nella cartella `Lab1`. 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.py`. Proseguite usando l'ambiente Spyder per svolgere gli esercizi. Ovviamente è possibile consultare il materiale didattico disponibile sulla pagina web del corso (<https://classroom.google.com/u/1/c/NzEwMzM0OTA4MDE4>). Si ricorda che le note relative alle lezioni Python possono essere lette con l'applicazione JupyterLab.

La consegna deve essere effettuata entro l'orario di fine dell'esercitazione.

LE ESERCITAZIONI SVOLTE CONSEGNATE OLTRE QUESTO TERMINE, O CHE NON RISPETTANO IL FORMATO INDICATO PER LA CONSEGNA, NON VERRANNO CONSIDERATE. In particolare, vi chiediamo di NON caricare un esercizio svolto per volta, di NON usare formati di compressione diversi da .zip, di NON rinominare i file o metterli in sottocartelle.

Fate attenzione che gli input siano richiesti all'utente UNO PER VOLTA e NELL'ORDINE RIPORTATO nell'esercizio e che le vostre stampe riportino a video i messaggi ESATTAMENTE nel formato atteso.

Ogni esercizio richiede che sia completata una funzione all'interno del file predisposto con lo stesso nome e, come visto a lezione, può essere testata eseguendo il file stesso, con i casi di test forniti. Notate infatti 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, come in sede di esame, per la correzione verranno usati insieme di dati di test diversi.**

Esercizi

- **A_Ex1(fileName):** Scrivere una funzione che prende in ingresso una stringa **fileName**, nome di un file di testo, e restituisce il numero totale di caratteri alfabetici contenuti nel file.
- **A_Ex2(fileName):** Scrivere una funzione che prende in ingresso una stringa **fileName**, nome di un file csv, contenente le informazioni sugli esami superati dagli studenti (nel formato *Matricola, Voto, Materia*), e restituisce una lista contenente tutte e sole le coppie (*Matricola, Materia*) per gli esami superati, tali cioè che lo studente con quella matricola abbia preso un voto maggiore o uguale a 18. Notare che una coppia è semplicemente una tupla di lunghezza due. Le coppie devono comparire nella lista nello stesso ordine in cui le informazioni si trovano nel file. Ad esempio, se il file contiene

```
Matricola,Voto,Materia
1345,29,Fisica
1987,17,Fondamenti
1346,27,Analisi
1896,30,Geometria
1753,30,Fisica
```

La funzione deve restituire

```
[('1345','Fisica'),('1346','Analisi'),('1896','Geometria'),('1753','Fisica')]
```

- **A_Ex3(fileName):** Scrivere una funzione che prende in ingresso una stringa **fileName**, nome di un file csv che ha lo stesso formato del file descritto nell'esercizio precedente (**A_Ex2**), e restituisce un insieme contenente tutte e sole le **Materie** per cui ci sono almeno due studenti che hanno preso almeno 29 (potete assumere che nel file non ci siano mai due righe uguali, aventi stessa matricola e stessa materia – in altri termini, il file contiene un solo voto per un certo studente ed una certa materia).
- **A_Ex4(fileName,anno):** Scrivere una funzione che prende in ingresso una stringa **fileName** che denota il nome di un file csv che contiene le informazioni sul numero e tipo di oggetti venduti in anni consecutivi da un negozio, ed un intero **anno** che denota un anno (che potete assumere siano tra quelli presenti nel file) e restituisce il nome dell'oggetto più venduto in quell'anno. Il file è nel seguente formato (come esempio, ma il numero di anni potrebbe variare):

```
Anno,2010,2011,2012
Zaino,27,21,11
Maglione,11,13,16
Giubbotto,13,15,17
```

Se ci sono più oggetti con lo stesso numero di vendite, si restituisca quello alfabeticamente più grande. Ad esempio, se il file è quello di sopra e l'anno è il 2012 allora il risultato restituito dalla funzione deve essere 'Giubbotto'.

- **A_Ex5(fileName,oggetto):** Scrivere una funzione che prende in ingresso una stringa **fileName** che denota il nome di un file csv nel formato di sopra ed il nome di un **oggetto** e restituisce il numero intero dell'anno in cui c'è stata la maggiore crescita assoluta nelle vendite rispetto all'anno precedente. Se ci sono più anni con la stessa crescita assoluta, restituire quello più grande. Se non c'è mai stata una crescita, la funzione restituisce il primo anno presente nel file. Ad esempio, se il file è quello di sopra ed **oggetto** vale 'Giubbotto' allora deve restituire 2012, se invece **oggetto** vale 'Zaino' allora deve restituire 2010 (primo anno presente nel file).
- **A_Ex6(fileName):** Scrivere una funzione che prende in ingresso una stringa **fileName** che denota il nome di un file csv nel formato di sopra e calcola in quale anno (numero intero) le vendite complessive sono state massime. Se ci sono 2 anni con le stesse vendite, si prenda l'anno più grande. Ad esempio, se il file è quello di sopra allora deve restituire 2010, anno in cui ci sono stati venduti in tutto 51 oggetti.
- **A_Ex7(fileName):** Scrivere una funzione che prende in ingresso un file di testo contenente solo lettere alfabetiche, spazi, cifre e caratteri newline ('\\n') e restituisce la somma dei numeri presenti nel file. Ad esempio, se il file contiene:

```
Paolo è andato a casa 3 volte
ed ha mangiato 11 ciambelle ed una TORTA
```

Allora il risultato è 14 (3+11) visto che questi sono i soli numeri presenti nel file.

- **A_Ex8(fileName):** Scrivere una funzione che prende in ingresso un file di testo contenente solo lettere alfabetiche, spazi, cifre e caratteri newline ('\\n') e restituisce il numero della riga che contiene più caratteri alfabetiche maiuscoli. Se ci sono più righe con lo stesso numero, restituisca quella di indice più grande. Si assuma che la prima riga abbia indice 1. Ad esempio, se il file è quello di sopra, allora il risultato è 2, poiché la seconda riga ha più caratteri alfabetiche maiuscoli della 1.
- **A_Ex9(fileName,squadra)** Scrivere una funzione che prende in ingresso una stringa **fileName**, nome di un file csv, contenente tutte le partite giocate in un torneo di calcio ed il nome di una squadra e calcola il numero di punti ottenuti da quella squadra. Il file ha il seguente formato:

Prima riga:

```
Nome_Squadra1, Nome_Squadra2, Numero_gol_Squadra1, Numero_gol_Squadra2
```

Altre righe:

```
Squadra1, Squadra2, golSquadra1, golSquadra2
```

La funzione deve restituire i punti in classifica della squadra **squadra**. Assumete che la squadra che vince ottiene 3 punti, la perdente ottiene 0 punti, e, in caso di pareggio, entrambe ottengono 1 punto.

Ad esempio se file contiene

```
Nome_Squadra1,Nome_Squadra2,Numero_gol_Squadra1,Numero_gol_Squadra2
Chelsea,Everton,2,0
Arsenal,Tottenham,0,0
Chelsea,Arsenal,0,1
Tottenham,Everton,1,2
```

E la squadra è Chelsea la funzione deve restituire 3.