

## Esercitazione Python n. 9 -- 30 Novembre 2021

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 A\_Ex1.py, per l'esercizio 2, nel file A\_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**. In fase di correzione verranno eseguiti dei test diversi da quelli che trovate attualmente.

Per **consegnare l'esercitazione svolta**, comprimate la cartella LabPython09 (**FATE ATTENZIONE**, la cartella che comprimate deve contenere gli esercizi e **NON** una ulteriore cartella con dentro gli esercizi) **in un file .zip** e caricatela sulla pagina del corso di <https://classroom.google.com/u/0/w/MTY5MzU2NjY1MTg2/t/all> (dalla sezione 'Esercitazioni su Python', selezionate 'Esercitazione 9' e successivamente 'Visualizza Compito'; poi cliccate su 'Aggiungi o crea' e scegliete il file da caricare). **NON** è necessario rinominare il file .zip. E' **NECESSARIO NON** rinominare i singoli file. Al termine dell'operazione cliccate su 'Contrassegna come completato'. La consegna deve avvenire in maniera inderogabile entro le **23:59 di Mercoledì 1 dicembre**.

A valle dell'esercitazione provvederemo ad effettuare una correzione automatica delle soluzioni consegnate, il cui esito verrà comunicato ad ogni studente tramite un messaggio di posta privato.

Si rammenta che **le esercitazioni consegnate in ritardo o che non rispettano le indicazioni per la consegna saranno automaticamente valutate con punteggio 0**. In particolare, si ricorda di **comprimere e riconsegnare l'intera cartella dell'esercitazione**, e non singolarmente i file degli esercizi, di **NON usare formati di compressione diversi da .zip**, di **NON** rinominare i file o **metterli in sottocartelle**.

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 un intero positivo, potete assumere che l'input sia sempre un numero intero maggiore di zero, e non è necessario nel codice effettuare controlli per gestire casi diversi da questo).

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

- **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 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(l):** Scrivere una funzione che riceve in ingresso una lista **l** di stringhe e restituisce il dizionario in cui le stringhe di **l** sono le chiavi ed il valore associato a ciascuna chiave **k** è il numero di volte in cui la stringa **k** compare nella lista. Se la lista **l** in ingresso è vuota, la funzione deve restituire il dizionario vuoto {}. Ad esempio, se **l** = ['casa', 'orso', 'cane', 'casa', 'orso', 'casa'] allora il dizionario restituito sarà {'casa': 3, 'orso': 2, 'cane': 1}. Notate che l'ordine per i dizionari non è rilevante.
- **A\_Ex5(s,n):** Scrivere una funzione che riceve in ingresso una stringa **s**, composta da soli caratteri alfabetici minuscoli e spazi, ed un numero **n** e calcoli il dizionario avente come chiavi lettere minuscole e come valore associato a ciascuna chiave **k** il numero di parole di **s** che cominciano per **k** e sono lunghe almeno **n** caratteri. Il dizionario NON deve contenere come chiavi lettere che non sono iniziali di almeno una parola lunga almeno **n** caratteri. Se la stringa in ingresso **s** è vuota la funzione deve restituire il dizionario vuoto {}. Ad esempio, se **s** vale 'tanto va la gatta al lardo che ci lascia lo zampino' ed **n** vale 3 la funzione deve restituire: {'t': 1, 'g': 1, 'c': 1, 'l': 2, 'z': 1}
- **A\_Ex6(fileName)** Scrivere una funzione che riceve in input una stringa **fileName**, nome di un file di testo, contenente in ogni riga una serie di numeri interi separati da virgole, e restituisce un dizionario che ha come chiavi i numeri (in formato intero) che appaiono nel file e come valore associato a ciascuna chiave **k** un insieme contenente i numeri di riga in cui appare **k**. Assumete che il numero identificativo delle righe parta dal valore 1. Ad esempio se il file contiene

```
10,-5,-5,0
10,-5,8,-3
```

la funzione deve restituire:

```
{10: {1,2}, -5: {1,2}, 0: {1}, 8: {2}, -3: {2}}
```

- **A\_Ex7(fileName)** 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. 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 il dizionario che ha come chiavi i nomi delle squadre che compaiono nel file csv e come valore associato a ciascuna chiave **k** i punti in classifica della squadra **k**. 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
```

la funzione deve restituire:

```
{'Chelsea': 3, 'Everton': 3, 'Arsenal': 4, 'Tottenham': 1}
```