

Esercitazione Python n. 11 -- 12 Dicembre 2023

Obiettivo dell'esercitazione è di prendere confidenza con le espressioni regolari.

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-4.5**. Una volta avviata la macchina virtuale, svolgete gli esercizi così come indicato nel testo. Usate 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/NjIwOTY0ODk3MDAx>). 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

Ex1(file): Scrivere una funzione Python che prende in input un numero intero n e una lista di interi l e restituisce la somma dei risultati della divisione intera di n per ognuno degli interi nella lista, gestendo l'eccezione eventualmente generata dalla divisione per 0 in modo tale che l'esecuzione non sia interrotta e la divisione errata non abbia impatto sul risultato della somma. Per esempio, se $n=5$ e $l=[1,5,0,2]$, la funzione deve restituire 8.

Ex2(file): Scrivere una funzione Python che prende in input il nome di un file csv, che contiene una tabella di interi con almeno una riga e due colonne, e restituisce due liste, la prima che contiene, per tutte le righe, in ordine, la somma dei valori contenuti nella riga, e la seconda che contiene, per tutte le colonne, in ordine, la somma dei valori contenuti nella colonna. Per fare questo, la funzione tenta di creare un array usando il modulo Numpy, e, se ottiene un'eccezione perché il file non esiste, restituisce due liste vuote, mentre se ottiene un'eccezione perché ci sono alcuni dati mancanti nel file, crea un dataframe usando il modulo Pandas.

N.B. Assumiamo che le colonne non abbiano intestazione. In questo caso, la funzione `load_csv` del modulo Pandas deve essere usata specificando il parametro `header=None`.

Ex3(m): Scrivere una funzione Python che, ricevendo in ingresso un array numpy quadrato $n \times n$ m , con $n \geq 0$, contenente gli interi 0 e 1 restituisca:

- ‘Principale’ se la diagonale principale (dall'angolo in alto a sinistra a quello in basso a destra) contiene più valori a 1 della diagonale secondaria (dall'angolo in alto a destra a quello in basso a sinistra)
- ‘Secondaria’ se la diagonale secondaria contiene più 1 di quella principale
- ‘Uguali’ se il numero di 1 è uguale.

Per esempio, ricevendo la matrice `[[1,0], [1,1]]` la funzione deve restituire ‘Principale’.

Ex4(m) Definizione: un elemento di una matrice si dice **sopraffatto verticale** se entrambi gli elementi sopra e sotto di lui sono strettamente maggiori del suo valore o **sopraffatto orizzontale** se entrambi gli elementi a destra e sinistra di lui sono strettamente maggiori del suo valore. Scrivere una funzione Python che, ricevendo in ingresso un array numpy restituisce una tupla con due valori, pari al numero dei sopraffatti verticali e orizzontali.

Per esempio, ricevendo la matrice `[[1,1,1], [1,0,0], [1,1,1]]` la funzione deve restituire (2,0).

Ex5(m): Definizione: una riga di una matrice si dice **crescente** se tutti i suoi elementi sono ordinati in modo strettamente crescente (da sinistra a destra). Scrivere una funzione Python che, ricevendo in ingresso un array numpy restituisce il numero delle righe crescenti.

Per esempio, ricevendo la matrice `[[1,2,4], [1,0,0], [1,1,2]]` la funzione deve restituire 1.

Ex6(m): Scrivere una funzione Python che, ricevendo in ingresso un array numpy restituisce il numero delle righe ottenibili come somma di altre due righe **adiacenti**.

Per esempio, ricevendo la matrice `[[1,2,4], [1,0,0], [2,2,4]]` la funzione deve restituire 1.

Ex7(anno,file): Scrivere una funzione Python che prende in input un anno, ed un nome di file (excel in formato xlsx) e legge il file scritto in lingua inglese che contiene informazioni su degli artisti con le seguenti colonne:

Artist ID	(Codice Artista)
Name	(Nome)
Nationality	(Nazionalità)
Gender	(Sesso)
Birth Year	(Anno di nascita)
Death Year	(Anno di morte, se manca l'artista è ancora in vita)

Conta quanti autori presenti nel file sono vivi nell'anno fornito ed hanno almeno 50 anni. **NOTA BENE:** alcune celle potrebbero essere vuote. Per verificare se un valore è NaN o meno si possono usare le funzioni Pandas `isna()` e `notna()`

Ex8(anno,area,file): Scrivere una funzione Python che prende in input un anno, un'area (in cm²) ed un nome di file (excel in formato xlsx) e legge il file scritto in lingua inglese che contiene informazioni su delle opere artistiche con le seguenti colonne:

Artwork ID	(Codice opera)
Title	(Titolo opera)
Artist ID	(Codice Artista)
Name	(Nome)
Date	(Data di creazione opera)
Height (cm)	(Altezza in centimetri)
Width (cm)	(Larghezza in centimetri)

Conta quante opere presenti nel file sono state realizzate **STRETTAMENTE PRIMA** dell'anno fornito in input ed hanno un'area (calcolata moltiplicando Height e Width) strettamente minore dell'area fornita. **NOTA BENE:** alcune celle potrebbero essere vuote. Per verificare se un valore è NaN o meno si possono usare le funzioni Pandas `isna()` e `notna()`

Ex9(file1,file2): Scrivere una funzione Python che, prende in input due files, uno nel formato dell'esercizio 7 ed uno nel formato dell'esercizio 8 e calcola quante opere presenti nel file 2 sono state realizzate da autori presenti nel file1 NOTA BENE: Gli autori sono identificati dal Artist ID (Codice Artista). Per verificare se un valore è NaN o meno si possono usare le funzioni Pandas isna() e notna()