

Esercitazione Python n. 6 -- 7 Novembre 2023

Obiettivo dell'esercitazione è iniziare a prendere confidenza con liste, tuple e insiemi.

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 insiemi di dati di test diversi.**

Esercizi

- **A_Ex1(l1, l2)** Scrivere una funzione che, ricevendo in ingresso due liste **l1** e **l2** contenenti numeri interi, e tali che $\text{len}(l2) \geq \text{len}(l1)$, restituisca una nuova lista composta dalla somma degli interi che si trovano nella stessa posizione in **l1** e **l2**. Se **l1** è più corta si assuma che gli interi mancanti siano uguali a 0. Ad esempio, se **l1** vale [3,6] e **l2** vale [3,4,9] la funzione deve restituire [6, 10, 9]. Se **l1** ed **l2** sono entrambe vuote (cioè pari a []), la funzione deve restituire una lista vuota.
- **A_Ex2 (start, n)** Scrivere una funzione che riceve in ingresso due interi non negativi **start** e **n** e restituisce la lista dei primi **n** numeri dispari $\geq \text{start}$. Se **n** vale zero, la funzione deve restituire una lista vuota. Ad esempio, se **start** vale 4 e **n** vale 3 la funzione deve restituire [5, 7, 9].
- **A_Ex3(l,n)** Scrivere una funzione che prende in ingresso una lista **l** di stringhe e un intero non negativo **n** e restituisce la stringa composta, in ordine, da tutti i caratteri con indice **n** di tutte le stringhe della lista. Se in una stringa in **l** non c'è il carattere di indice **n** (perché la stringa è troppo corta), la funzione deve inserire il carattere '!'. Se la lista in ingresso è vuota, la funzione deve restituire la stringa vuota. Ad esempio, se **l** vale ['tanto', 'va', 'la', 'gatta', 'al', 'lardo'] e **n** vale 3 la funzione deve restituire la stringa 't!!t!d'

- **A_Ex4(l)** Scrivere una funzione che prende in ingresso una lista **l** di interi e restituisce la stringa composta da soli caratteri 'P' e 'D' che indicano se l'intero nella lista è pari o dispari. Più precisamente, se il numero in posizione *i* in **l** è pari, la stringa restituita dalla funzione contiene 'P' in posizione *i*, 'D' altrimenti. Se la lista è vuota la funzione deve restituire la stringa vuota. Ad esempio, se **l** vale [3,7,8,9] la funzione deve restituire la stringa 'DDPD'.
- **A_Ex5(l)** Scrivere una funzione che prende in ingresso una lista **l** di stringhe e restituisce una lista di interi, tale che ciascun intero *x* nella lista restituita è ottenuto dalla somma dei codici Unicode dei caratteri della stringa che in **l** occupa la stessa posizione che occupa *x* nella lista restituita. Ad esempio, se **l** è ['ama', 'ma', 'amaca'] la funzione dovrebbe restituire la lista [303,206,499], in quanto il codice Unicode di 'a' è 97, quello di 'm' è 109 e quello di 'c' è 99 (e quindi ad esempio ad 'ama' corrisponde l'intero 303). Se la lista in input è vuota, la funzione deve restituire una lista vuota.
- **A_Ex6(l1,l2)** Scrivere una funzione che prende in ingresso due liste **l1** e **l2** contenenti numeri interi e restituisce una lista contenente gli elementi di **l1** che NON compaiono in **l2**. Nella lista restituita gli elementi devono comparire in **ordine crescente**. Si noti che eventuali ripetizioni di un elemento *x* in **l1** sono tutte non presenti nel risultato se *x* compare anche solo una volta in **l2**, mentre sono tutte presenti nel risultato se *x* non compare in **l2**. Ovviamente, se la lista **l1** è vuota la funzione deve restituire la lista vuota, invece se **l2** è vuota, la funzione deve restituire una lista uguale a **l1**. Ad esempio, se **l1** = [1, 3, 7, 2, 1,-5, 7] e **l2** = [1, 3], la funzione deve restituire la lista [-5, 2,7,7].
- **A_Ex7(l)** Scrivere una funzione che prende in ingresso una lista **l** contenente stringhe non vuote e restituisce (la tupla costituita da) i caratteri più frequenti di ciascuna stringa. Se una stringa ha più di un carattere più frequente, la tupla deve contenere il più piccolo carattere nell'ordine Unicode. Ad esempio, se **l** = ['amaca', 'amaranto', 'rosso'] allora la funzione deve restituire ('a', 'a', 'o').
- **A_Ex8(l)** Scrivere una funzione che prende in ingresso una lista **l** *non vuota* di stringhe e calcola quale carattere alfabetico minuscolo ('a'-'z') compare in più stringhe. Se ci sono più caratteri che compaiono lo stesso numero di volte si scelga quello alfabeticamente più grande. Ad esempio, se **l** = ['casa', 'senape', 'ketchup', 'pasta'], allora il carattere da restituire è 's' che compare in 3 stringhe ed è più grande di 'a' e 'p', che anche compaiono in 3 stringhe.
- **A_Ex9(l)** Scrivere una funzione che prende in ingresso una lista **l** di stringhe e restituisce un insieme contenente tutti e soli i caratteri che appaiono almeno due volte in una delle stringhe. Ad esempio, se **l** = ['casa', 'albero', 'bello'], allora l'insieme da restituire sarà {'a', 'l'}. Se la lista **l** in ingresso è vuota, oppure non ci sono caratteri che appaiono almeno due volte in una stringa di **l**, la funzione deve restituire l'insieme vuoto.
- **A_Ex10(l)** Scrivere una funzione che prende in ingresso una lista **l** di stringhe e restituisce un insieme di coppie (x1, x2) tali che x1 e x2 sono stringhe di **l** della stessa lunghezza e x1 è diverso da x2. Notare che una coppia è semplicemente una tupla di lunghezza due. In altri termini, l'insieme deve contenere tutte e sole le coppie formate combinando a due a due, in tutti i possibili modi, le stringhe di **l** che hanno la stessa lunghezza, escludendo i casi in cui una stringa si combina con sé stessa. Ad esempio, se **l** = ['tre', 'h', 'plqa', 'a', 'due'], allora l'insieme di coppie da restituire sarà {'(tre','due)', ('h','a'), ('a','h'), ('due','tre')}. Se la lista **l** in ingresso è vuota, la funzione deve restituire l'insieme vuoto.