

Esercitazione Python n. 7 -- 16 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 anche dei test diversi da quelli che trovate attualmente.

Per **consegnare l'esercitazione svolta**, comprimate la cartella LabPython07 **in un file .zip** e caricatela sulla pagina del corso <https://classroom.google.com/u/0/w/MzkwNTM3Njg2Njc3/t/all> (dalla sezione 'Esercitazioni su Python in Laboratorio', selezionate 'Esercitazione 7' 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 17 novembre**.

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).

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 3030). 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(s)** Scrivere una funzione che prende in ingresso una stringa **s** e restituisce una lista composta da tutti e soli i caratteri alfabetici maiuscoli contenuti in **s**, senza ripetizioni. La lista restituita deve essere **ordinata in ordine alfabetico** (cioè secondo l'ordine UNICODE crescente). Se la stringa in ingresso è vuota, la funzione deve restituire una lista vuota. Ad esempio, se **s**='cIAo MAMma', la funzione deve restituire la lista ['A','I','M'].
- **A_Ex8(s1,s2)** Scrivere una funzione che prende in ingresso due stringhe **s1** e **s2** e restituisce il prefisso più lungo comune alle due stringhe. Ad esempio, se **s1** è 'amaca' e **s2** è 'amaranto', la funzione deve restituire la stringa 'ama'. Se invece, **s1** è 'amaca' e **s2** è 'stringa' la funzione deve restituire la stringa vuota ''. Se (almeno) una delle due stringhe è vuota la funzione deve restituire ''.
- **A_Ex9(t)** Scrivere una funzione che prende in ingresso una tupla **t** contenente stringhe non vuote e restituisce l'insieme dei caratteri più frequenti di ciascuna stringa. Se una stringa ha più di un carattere più frequente, l'insieme deve contenere il più piccolo carattere nell'ordine Unicode. Ad esempio, se **t** = ('amaca', 'amaranto', 'rosso') allora la funzione deve restituire {'a','o'}.
- **A_Ex10(i)** Scrivere una funzione che prende in ingresso un insieme **i** di interi e restituisce la tupla contenente i soli elementi divisibili per 5. La tupla deve essere ordinata in ordine crescente. Ad esempio, se **i**={3, 345,15,24,25, -10}, la funzione deve restituire (-10,15,25,345).