Sapienza Università di Roma
Corso di Laurea in Ingegneria Informatica e Automatica A.A. 2021-22

Esercitazione Python n. 6 -- 9 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**, comprimete la cartella LabPython06 **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', selezionate 'Esercitazione 6' 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 10 novembre**.

Si rammenta che **le esercitazioni consegnate in ritardo e/o che non rispettano le indicazioni per la consegna non saranno prese in considerazione per l'assegnazione del punto bonus**. 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(s)** Completare la funzione Python A_Ex1(s) (contenuta nel file A_Ex1.py) realizzando una funzione che, ricevendo in ingresso una stringa **s**, restituisca una nuova stringa composta dai caratteri più frequenti nella stringa nell'ordine in cui compaiono. Ad esempio se **s** vale `'caso palese'` allora la funzione deve restituire `'ase'` in quanto i tre caratteri compaiono tutti due volte nella stringa e due è la frequenza massima (si noti che nella stringa restituita ogni carattere compare una sola volta). Nel caso in cui la **s** in input sia la stringa vuota la funzione deve restituire una stringa vuota.

- **A_Ex2(s1,s2)** Completare la funzione Python A_Ex2(s1,s2) (contenuta nel file A_Ex2.py) realizzando una funzione che, ricevendo in ingresso due stringhe, **s1** ed **s2**, restituisca il numero di caratteri che compaiono lo stesso numero di volte (diverso da 0) nelle due stringhe. Ad esempio se **s1** vale `'caso palese'` e **s2** vale `'casa blue'` allora la funzione deve restituire `4`, poiché i caratteri `'c'`,`'a'`, `' '` (spazio) e `'l'` compaiono lo stesso numero di volte in **s1** e **s2**. Ovviamente se una delle due stringhe in input è vuota la funzione deve restituire `0`. Si noti che la funzione **deve** restituire sempre un valore di tipo intero.

- **A_Ex3(s)** Completare la funzione Python A_Ex3(s) (contenuta nel file A_Ex3.py) realizzando una funzione che, prendendo in ingresso una stringa **s** composta di sole cifre e di lunghezza massima 10, restituisce `True` se almeno un carattere di indice *i* è uguale alla cifra corrispondente. Ad esempio, se la stringa **s** vale `'999379'` allora la funzione deve restituire `True` poiché il carattere di indice 3 è la cifra `'3'`. Se invece la stringa **s** è `'234567890'` allora la funzione deve restituire `False`, perché in nessun carattere di indice *i* è uguale alla cifra corrispondente. Ovviamente, se la stringa in input è vuota, la funzione deve restituire `False`. Si noti che il valore restituito è un booleano.

- **A_Ex4(s,n)** Completare la funzione Python A_Ex4(s,n) (contenuta nel file A_Ex4.py) realizzando una funzione che, prendendo in ingresso una stringa **s** ed un numero intero positivo **n** (quindi maggiore di 0), restituisce la

stringa che contiene, una sola volta, i caratteri di **s** il cui codice Unicode sia esattamente divisibile per **n**. Ad esempio, se **s** è `'stringa di prova'` ed **n** vale `3`, allora la funzione deve restituire `'rio'` perché solo questi caratteri hanno la proprietà desiderata. Notate che sia la `'r'` che la `'i'` compaiono una sola volta nella stringa restituita, anche se compaiono due volte in **s**. Inoltre, nella stringa restituita i caratteri devono comparire nello stesso ordine in cui compaiono in **s**.

- **A_Ex5(s)** Completare la funzione Python A_Ex5(s) (contenuta nel file A_Ex5.py) realizzando una funzione che, prendendo in input una stringa **non vuota s** contenente numeri interi positivi separati dal carattere `'-'` (trattino), restituisce il massimo intero contenuto in **s**. Ad esempio, se **s** vale `'12-123-45-6-78'` la funzione deve restituire `123`. Si assuma che `'-'` non compaia mai in prima o ultima posizione (quindi se la stringa contiene un solo numero, non contiene alcun trattino). Si noti che la funzione **deve** restituire sempre un valore di tipo intero.

- **A_Ex6(s)** Completare la funzione Python A_Ex6(s) (contenuta nel file A_Ex6.py) realizzando una funzione che riceve in ingresso una stringa **s**, e restituisce la frequenza (un numero intero) del carattere alfabetico maiuscolo che appare più volte in **s**. Se **s** non contiene alcun carattere alfabetico maiuscolo, la funzione deve restituire il numero intero 0. Ad esempio, se **s** vale `'aHa^^&^HH'`, la funzione deve restituire il numero intero `3`. Se invece **s** vale `'CIAO'`, la funzione deve restituire il numero intero `1`. Si noti che la funzione **deve** restituire sempre un valore di tipo intero.

- **A_Ex7(b1,b2)** Completare la funzione Python A_Ex7(b1,b2) (contenuta nel file A_Ex7.py) realizzando una funzione che prende in input due stringhe **b1** e **b2** contenenti solo caratteri `'0'` ed `'1'` (interpretati come bit) e restituisce una stringa di `'0'` ed `'1'` che corrisponde all'AND bit a bit di **b1** e **b2** letti da sinistra a destra. Si noti che l'AND bit a bit effettua l'AND solo dei bit che occupano la stessa posizione nelle stringhe **b1** e **b2**, e per ogni posizione *i* produce 1 solo se entrambi i bit in posizione *i* in **b1** e **b2** sono pari ad 1. Per i bit di **b1** che non hanno un corrispondente bit in **b2** (in questo caso **b1** è più lunga di **b2**) la funzione deve calcolare l'AND con lo `'0'`. Analogamente nel caso in cui **b2** sia più lunga di **b1**. Ad esempio, se **b1** vale `'100'` e **b2** vale `'1011'`, la funzione deve restituire la stringa `'1000'`. Se invece **b1** vale `''` (stringa vuota) e **b2** vale `'111'` la funzione deve restituire '000'. Se entrambe **b1** e **b2** sono vuote, la funzione deve restituire la stringa vuota.