

Compito d'esonero -- 5 Novembre 2019 -- Compito D

Istruzioni (leggere attentamente)

Nota importante: la mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione dell'esercitazione.

Registrazione dei dati dello studente: PRIMA DI INIZIARE, eseguite (con un doppio click sull'icona) il programma `RegistraStudente` che si trova nella cartella `Esame`. Inserite (separatamente) *Numero di Matricola*, *Cognome* e *Nome* seguendo le istruzioni che compaiono sul terminale, e confermate i dati che avete inserito. Il programma genera il file `studente.txt` che contiene Matricola, Cognome e Nome su tre righe separate (nell'ordine indicato). Il file `studente.txt` non deve essere modificato manualmente. Verificate che i dati nel file `studente.txt` siano corretti. In caso di errore potete rieseguire il programma `RegistraStudente`.

Svolgimento degli esercizi: Per ogni esercizio avete una cartella `EsercN` che contiene un file dal nome `A_ExN.py` (dove **N** è il numero dell'esercizio) con lo scheletro della soluzione. Questo file incorpora un codice di test che proverà la vostra soluzione per un certo numero di possibili dati in input. Aprite il file con IDLE e modificate SOLO il contenuto della funzione. Eseguendo il file `.py` si otterrà il responso dei test sulla console. Non spostate i file dalla loro posizione e non create nuovi file. **Si noti che per la correzione verranno usati insieme di dati di test diversi.**

E' possibile consultare la documentazione ufficiale del linguaggio Python, ma **non è possibile usare libri o appunti**. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

Esercizi

- **D_Ex1(s):** Scrivere una funzione che riceve in ingresso una stringa **s**, e restituisce la frequenza del carattere di **s** che appare meno volte in **s**. Se non appare nessun carattere in **s** (cioè **s** è la stringa vuota), la funzione deve restituire il numero intero 0. Ad esempio, se **s** = "&&a&aa^h^hh&", la funzione deve restituire il numero intero 2. Se invece **s** = "hhhh", la funzione deve restituire il numero intero 4.
- **D_Ex2(s):** Scrivere una funzione che prende in input due stringhe **b1** e **b2** fatte da caratteri '0' ed '1' (interpretati come bit) e restituisce una stringa di '0' e '1' che corrisponde all'OR bit a bit di **b1** e **b2**, letti da sinistra a destra. 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'OR con lo '0'. Analogamente nel caso in cui **b2** sia più lunga di **b1**. Ad esempio, se **b1**='100' e **b2**='0011', la funzione deve restituire la stringa '1011'. Se invece **b1**='101' e **b2**='' la funzione deve restituire '101'. Se entrambe **b1** e **b2** sono vuote, la funzione deve restituire la stringa vuota.
- **D_Ex3(l)** Scrivere una funzione che riceve in ingresso una lista **l** di numeri interi compresi nell'intervallo 0-100 (estremi esclusi) e restituisce il più piccolo numero che è (strettamente) maggiore sia del numero precedente che del numero successivo nella lista. Notate che il primo ed ultimo numero della lista NON vanno mai considerati perché non hanno sia un elemento precedente che uno successivo. Ad esempio, se la lista **l** = [5,7,6,8,4,9] la funzione deve restituire il valore 7, poiché i soli valori 7 ed 8 hanno la proprietà di essere maggiori dei numeri precedenti e successivi e 7<8. Se nessun numero della lista soddisfa la proprietà, la funzione deve restituire il valore 100