

Compito d'esame -- 21 gennaio 2019 -- Compito B

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 `B_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 Spyder 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.

Al termine dello svolgimento del compito d'esame **NON è necessario effettuare una procedura di consegna** dell'elaborato. Le vostre soluzioni saranno copiate direttamente dalla cartella **Esame** della macchina su cui avete lavorato. Quindi, per concludere la prova, è sufficiente chiudere il programma Spyder e lasciare la vostra postazione.

Esercizi

- **B_Ex1(s1,s2)** Scrivere una funzione che prende in ingresso due stringhe **s1** e **s2**, e restituisce la lunghezza del suffisso più lungo comune alle due stringhe, cioè la lunghezza della sottostringa comune più lunga posta alla fine di **s1** ed **s2**. Ad esempio, se **s1** = "arciere" e **s2** = "pompieri", la funzione deve restituire 4. Se invece **s1** = "arciere" e **s2** = "pippo" la funzione deve restituire 0. Se almeno una delle due stringhe è vuota la funzione deve restituire 0.
- **B_Ex2 (m)** Scrivere una funzione che prende in ingresso una matrice di numeri interi **m**, rappresentata come lista di liste, e restituisce la lista ORDINATA in modo crescente degli interi in **m** che sono negativi e sono preceduti e seguiti da un numero positivo sia sulla riga che sulla colonna a cui appartengono, o viceversa (cioè sono positivi e sono preceduti e seguiti sia sulla riga che sulla colonna da interi negativi). Per semplicità potete assumere che gli elementi sui bordi di **m** NON siano da controllare (perché di fatto non possono soddisfare la condizione voluta). Ad esempio, se la matrice è:

3	2	-1	5
2	-10	3	-4
4	5	-2	4
2	10	3	4

allora la funzione deve restituire [-10,-2,3].

- **B_Ex3(l)** Scrivere una funzione che prende in ingresso una lista **l** contenente stringhe, e restituisce la lista ottenuta da **l** replicando nella posizione successiva ogni stringa che contiene il maggior numero di lettere minuscole. Ad esempio, se **l** = ["piPPo", "PIppO", "PLuto", "PiPPo"], la funzione deve restituire ["piPPo", "piPPo", "PIppO", "PLuto", "PLuto", "PiPPo"], essendo il maggior numero di lettere minuscole contenuto in una stringa pari a tre.
- **B_Ex4(file)** Scrivere una funzione che prende in ingresso un file di testo **file** in formato csv che contiene tutte le partenze ed arrivi di treni in un giorno dalle varie stazioni nel seguente formato:

CodiceTreno,Stazione,Arrivo/Partenza

dove **CodiceTreno** è il codice (univoco) del treno, **Stazione** è il nome (univoco) della stazione in cui il treno arriva o da cui parte, mentre **Arrivo/Partenza** è una stringa tra 'A' o 'P' ed indica se il treno è arrivato o partito. Dato un file in questo formato, la funzione deve restituire un dizionario con chiave il nome della stazione e valore la lista, ORDINATA E SENZA RIPETIZIONI, dei codici dei treni che hanno iniziato il viaggio da quella stazione, cioè che sono partiti da quella stazione ma non ci sono mai arrivati. Ad esempio, se il file contiene:

```
T81,Termini,P
T72,Tiburtina,P
T72,Tiburtina,A
T81,Tiburtina,A
T72,Termini,P
T61,Tiburtina,P
T61,Termini,A
T72,Trastevere,A
```

allora la funzione deve restituire il dizionario:

```
{'Termini': ['T72', 'T81'],'Tiburtina': ['T61'],' Trastevere': []}.
```