

Compito d'esame -- 24 giugno 2021

Istruzioni (leggere attentamente)

Nota importante: la mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione della prova d'esame.

Registrazione dei dati dello studente: PRIMA DI INIZIARE, aprite il programma Python `RegistraStudente.py` che si trova nella cartella `Esame` nell'ambiente IDLE. 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 la procedura.

Tempo a disposizione: 1 ora e 30 minuti

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'esame (ad esempio, per l'esercizio 1 scrivete il vostro programma nel file `Ex1.py`, per l'esercizio 2, nel file `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**. 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 (<file:///Library/Frameworks/Python.framework/Versions/3.9/Resources/English.lproj/Documentation/index.html>),
ma **non è possibile usare libri o appunti**. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

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 una lista non vuota di stringhe, potete sempre assumere l'input sia in tale forma e non è necessario nel codice effettuare controlli per gestire casi diversi da questo, considerando, ad esempio, il caso di lista vuota).

Per gli esercizi relativi a lettura da file, la stringa in input che identifica il file è sempre comprensiva anche della sua estensione e il file risiede sempre nella stessa directory dell'esercizio.

Esercizi

- **Ex1(I)** Scrivere una funzione che prende in input una lista di numeri interi `I` e restituisce una lista ottenuta eliminando da `I` ogni elemento che è maggiore o uguale di tutti quelli che lo seguono. L'elemento in ultima posizione non deve quindi essere eliminato. Se la lista in input è vuota, la funzione deve ovviamente restituire una lista vuota. Ad esempio, se `I=[10,3,4,7,5,6]`, la funzione deve restituire `[3,4,5,6]`; se `I=[12]`, la funzione deve restituire `[12]` (essendoci un unico elemento, questo è anche l'ultimo e non deve essere eliminato).

- **Ex2(file)** Scrivere una funzione che prende in ingresso un file di testo **file** contenente solo lettere minuscole, spazi e andate a capo e restituisce l'insieme di parole presenti in **file** che iniziano e finiscono con la stessa lettera e contengono almeno una *sottostringa interna* costituita da due lettere consecutive nell'ordine alfabetico (cioè, 'ab', 'bc', 'cd', ...). Per sottostringa interna di una stringa *s* si intende una sequenza di lettere non sovrapposta ai caratteri iniziale e finale di *s*. Ad esempio, se il file contiene

il nemico era ostico ed opulento
ma lei disse astuta abracadabra
trasformando la sua spada in ghirlanda

la funzione deve restituire {'ostico', 'astuta', 'abracadabra'}, in quanto queste parole iniziano e finiscono con la stessa lettera ('o' per 'ostico' ed 'a' per le altre due) e tutte contengono internamente una sequenza di due lettere consecutive nell'ordine alfabetico, e cioè, 'ostico', 'abracad**ab**ra' ed 'ast**ut**a' (si noti che in 'astuta' c'è anche la sottostringa 'tu', ma che comunque è sufficiente individuare una sola di tali sottostringhe per selezionare la parola). La parola 'opulento' invece non è nell'insieme restituito, anche se inizia e finisce con la stessa lettera e contiene la sottostringa 'op', in quanto 'op' non è interna.

- **Ex3 (diz,file)** Scrivere una funzione che simula una partita di un gioco da tavolo tipo Risiko, in cui ci sono dei giocatori e dei territori, ed in ogni territorio è presente un certo numero di armate di uno dei giocatori (non ci sono territori vuoti ed in un territorio ci sono armate di un solo giocatore). La funzione riceve in input due informazioni:
 - il dizionario **diz** che descrive la situazione attuale, dove le chiavi sono i giocatori ed i valori sono a loro volta dei dizionari con chiave il nome dei territori di proprietà del giocatore e valore il numero di armate presenti in quel territorio.
 - il nome di un file **file** csv contenente le mosse che ogni giocatore intende effettuare nel formato:

nome giocatore, territorio di attacco, territorio attaccato

La vostra funzione deve calcolare e restituire il dizionario con la situazione dopo aver eseguito tutte le mosse presenti nel file, seguendo queste regole:

- Le mosse vanno eseguite nell'ordine in cui si trovano nel file.
- Per attaccare un territorio bisogna che il giocatore sia proprietario del territorio da cui si attacca e NON sia proprietario del territorio attaccato. Se non è questo il caso, la mossa va ignorata e si passa alla prossima. Assumete che tutti i territori confinino tra di loro, cioè da ogni territorio si può attaccare qualunque altro.
- Un attacco è vincente se il numero di armate presenti nel territorio da cui si attacca è strettamente maggiore del numero di armate presenti nel territorio attaccato. Se il numero è minore o uguale, l'attaccante perde tutte le sue armate meno una che rimane nel territorio da cui si è attaccato.
- Se un attacco è vincente, l'attaccante occupa il territorio attaccato con un numero di armate uguale alla differenza tra le sue armate e quelle del difensore. Nel territorio da cui attaccava viene aggiunta 1 nuova armata.

Ad esempio, se il dizionario vale: {'Paolo': {'Italia': 5, 'Francia': 3}, 'Anna': {'Germania': 6}, 'Giorgio': {'Spagna': 2, 'Austria': 1}},

ed il file contiene:

Paolo,Italia,Spagna
Anna,Spagna,Francia
Giorgio,Spagna,Francia
Anna,Germania,Francia
Giorgio,Spagna,Germania

la funzione deve restituire {'Paolo': {'Italia': 1, 'Spagna': 3}, 'Anna': {'Germania': 1, 'Francia': 3}, 'Giorgio': {'Austria': 1}}.