

Compito d'esame -- 23 giugno 2020 -- Compito A

Istruzioni (leggere attentamente)

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

Svolgimento degli esercizi: Per ogni esercizio avete una voce EsercizioN nel menu di Exam.net (dove **N** è il numero dell'esercizio). Ognuna di queste voci un file `.py` differente con lo scheletro della soluzione (notate che il nome delle funzioni è sempre lo stesso in tutti i file, ma si tratta di funzioni diverse, una per ciascun esercizio). Questo file incorpora un codice di test che proverà la vostra soluzione per un certo numero di possibili dati in input. **Per OGNI esercizio eseguite prima la connessione (tasto 'Connect').** Una volta connessi verrà indicata l'avvenuta connessione e sarà mostrato nome, cognome e matricola dello studente (quelli inseriti su Exam.net per effettuare l'accesso all'esame). Potrete a questo punto svolgere l'esercizio scrivendo la vostra soluzione Python nell'area di testo 'Soluzione da svolgere'. Per eseguire il codice, cliccate sul bottone 'Save and Run' posto sotto l'area di testo. Solo in questo modo potrete salvare il codice scritto. **Se scrivete codice e non lo eseguite, questo non sarà salvato per la correzione.** Eseguendo il codice otterrete il responso dei test sull'area di output. Per visionare i file in input agli esercizi 3 e 4 cliccate sul link relativo in fondo alla pagina corrispondente all'esercizio. **Si noti che per la correzione verranno usati insieme i dati di test diversi.**

E' possibile consultare la documentazione ufficiale del linguaggio Python, ma **non è possibile usare libri o appunti.**

Esercizi

- **A_Ex(s,n)** Scrivere una funzione che riceve in ingresso una stringa **s** ed un intero positivo **n**. La funzione deve restituire una nuova stringa ottenuta eliminando da **s** la più lunga sequenza iniziale di caratteri s_I tale che la somma dei codici UNICODE dei caratteri in s_I sia minore o uguale a **n**. Ad esempio, se **s**='abc' ed **n**=200, la funzione deve restituire 'c', in quanto il codice UNICODE di 'a' è 97, il codice UNICODE di 'b' è 98, il codice UNICODE di 'c' è 99, e pertanto 'ab' è la più lunga sequenza iniziale di **s** che si sta cercando (infatti $97+98 < 200$ ma $97+98+99 > 200$). Si ricorda che per ottenere il codice UNICODE di un carattere *c* si può usare la funzione `ord(c)`.
- **A_Ex(m)** Scrivere una funzione che riceve in ingresso una matrice **m** di interi positivi rappresentata per righe come lista di liste e restituisce una matrice che ha come righe le colonne di **m** per le quali la somma di tutti gli elementi è pari. L'ordine delle righe nel risultato deve corrispondere all'ordine delle colonne selezionate di **m**. Ad esempio se l'input **m** rappresenta la matrice

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 3 \\ 1 & 0 & 2 \end{pmatrix}$$

la funzione deve restituire la rappresentazione per righe come lista di liste della matrice

$$\begin{pmatrix} 1 & 4 & 1 \\ 3 & 3 & 2 \end{pmatrix}$$

In quanto la somma degli elementi della colonna 1 di **m** è $1+4+1=6$, che è pari, e quindi la colonna 1 viene selezionata nel risultato, la somma degli elementi della colonna 2 di **m** è $2+1+0=3$, che è dispari, e quindi la colonna 2 di **m** non viene inclusa nel risultato, la somma degli elementi della colonna 3 di **m** è $3+3+2=8$, che è pari, e quindi anche la colonna 3 di **m** viene selezionata nel risultato. Si noti che nel risultato le colonne 1 e la colonna 3 di **m** diventano righe, e che la riga corrispondente alla colonna 1 precede la riga corrispondente alla colonna 3 (cioè viene riprodotto nelle righe del risultato lo stesso ordine che le colonne selezionate hanno in **m**).

- **A_Ex(file)** Si considerino solo file di testo in cui sono presenti solo:
 1. caratteri alfanumerici (senza lettere accentate),
 2. spazi bianchi,
 3. caratteri di fine riga ('`\n`'),
 4. espressioni di tagging che hanno la forma `<x attributo='dato'>`, in cui *x* è una lettera minuscola, *attributo* è una stringa di sole lettere minuscole e *dato* è una stringa di caratteri alfanumerici e spazi.
 5. Eventuali espressioni di tagging scorrette, cioè che non rispettano la forma descritta al punto 4 (ad esempio, manca la *x*, oppure un apice, oppure è presente più di un attributo).

Scrivere una funzione che prende in ingresso un file di tale forma e restituisce un dizionario le cui chiavi sono gli attributi presenti nei tagging corretti (che cioè rispettano la forma descritta al punto 4), ed il valore associato a ciascuna chiave è l'insieme dei dati assegnati a quell'attributo nei tagging corretti (si memorizzino tutti i dati come valori di tipo stringa). Si può assumere che un tagging non è mai suddiviso su più righe. Ad esempio, se il file contiene il testo

```
Ogni domenica <p nome='Marco Rossi'> marco incontra
<p nome='Mario Bianchi'> mario <p> e vanno allo
<l indirizzo='viale dei Gladiatori'>stadio a vedere la
<p casa='roma' ospite='inter'>partita<p>
```

La funzione deve restituire

```
{'nome': {'Marco Rossi', 'Mario Bianchi'}, 'indirizzo': {'viale dei Gladiatori'}}
```

Si noti in particolare che l'ultimo tagging è scorretto perché ci sono due attributi (e quindi non rispetta le indicazioni date al punto 4 precedente). Per risolvere l'esercizio è conveniente usare le espressioni regolari, ma è possibile anche usare metodi diversi.

- **A_Ex(file1,file2)** Scrivere una funzione che riceve in ingresso le informazioni sulle vendite di un negozio divise in due files, in formato csv, uno che riporta i prezzi unitari dei prodotti (**file1**), ed uno che riporta per ogni ricevuta quali prodotti sono stati acquistati ed in che quantità (**file2**). La funzione deve calcolare per ogni ricevuta il totale dei costi. Il primo file contiene su ciascuna riga il nome del prodotto ed il prezzo per una unità di prodotto, secondo lo schema:

Prodotto,CostoUnitario

Il secondo file contiene su ciascuna riga il codice della ricevuta, un prodotto acquistato con la spesa a cui la ricevuta si riferisce, e quante unità di quel prodotto sono state acquistate in quella ricevuta, secondo lo schema:

Ricevuta,Prodotto,Quantita

La funzione deve leggere i dati e restituire un dizionario con chiave la ricevuta e valore il totale complessivo della ricevuta. Si noti però che **file2** potrebbe contenere più righe riportanti lo stesso codice di ricevuta e lo stesso prodotto. In questo caso, nel calcolare il totale **bisogna considerare solo la prima riga che contiene quel codice di ricevuta e quel prodotto**.

Ad esempio, se il file1 è:

```
Prodotto,CostoUnitario
Latte,2
Pane,3
Mela,1
Zucchero,5
Sale,4
Acqua,1
Biscotti,4
```

Ed il file2 è:

```
Ricevuta,Prodotto,Quantita
R1,Latte,2
R2,Mela,4
R1,Sale,1
R2,Biscotti,2
R1,Sale,5
R3,Acqua,6
R3,Acqua,2
R1,Zucchero,1
```

allora il risultato sarà {'R1': 13, 'R2': 12, 'R3': 6}. (Si noti che le righe R1, Sale, 5 e R3, Acqua, 2 sono scartate durante il calcolo del costo totale per ricevuta).