

Compito d'esame -- 14 febbraio 2022 -- Compito B

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, 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`.

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'esercitazione (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**.

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(l,x,y)** Scrivere una funzione che riceve in ingresso una lista **l** contenente numeri interi e due numeri interi **x** e **y** e restituisce una lista contenente solo gli elementi che si trovano nelle posizioni contenute fra la posizione di una **x** e la posizione della prima **y** che segue la **x** (escludendo gli estremi, cioè la lista restituita non contiene la **x** iniziale e la **y**). Notate che ogni **x** che segue un'altra **x** prima di incontrare una **y** è contenuta nel risultato, e che se una **x** non è seguita da alcuna **y**, gli elementi dopo la **x** **NON** sono contenuti nel risultato. Gli elementi del risultato devono comparire nello stesso ordine in cui compaiono in **l**. Ad esempio, se **l**=[1, 2, 2, 0, 3, 1, 3, 4, 2, 1, 4, 2, 3], **x**=2 e **y**=4, la funzione deve restituire [2, 0, 3, 1, 3, 1], deve cioè inserire nel risultato gli elementi che di seguito evidenziamo in rosso [1, 2, **2**, **0**, **3**, **1**, **3**, 4, 2, **1**, 4, 2, 3].
- **Ex2(m)** Scrivere una funzione che riceve in ingresso una matrice quadrata **m** di numeri interi (rappresentata come lista di liste) e restituisce il valore massimo della somma dei quadratini 2x2 contenuti in **m**. Ad esempio, se la matrice **m** vale:

```
[[3, 2, 3, 4],  
 [2, 2, 1, 4],  
 [6, 1, 2, 3],  
 [1, 1, 1, 1]]
```

allora la funzione deve restituire 12 perché la somma degli elementi del blocchetto 2x2 evidenziato in grassetto e con vertice in alto a sinistra (0,2) vale 12 e non ci sono altri blocchetti 2x2 con somma maggiore. Se la matrice ha dimensione 0 (è cioè rappresentata da una lista vuota) la funzione deve restituire 0, invece se ha dimensione 1 allora la funzione deve restituire il suo unico elemento.

- **Ex3(file,l)** Scrivere una funzione che aiuti una software house a partecipare ad una gara e che prende in ingresso un file di testo **file** (in formato csv) ed una lista **l**. Il file **file** contiene le informazioni sul personale a disposizione, le sue disponibilità di tempo (in giornate), le sue competenze, gli anni di esperienza ed il costo orario. Se un dipendente ha più di una competenza comparirà in più righe del file. Il file csv **file** ha il seguente formato:

```
NomeDipendente,Competenza,GiorniDisponibili,CostoGiornaliero
```

dove NomeDipendente è il nome univoco del dipendente, Competenza è la competenza posseduta dal dipendente, GiorniDisponibili è il numero di giornate che il dipendente ha disponibili per quella competenza e CostoGiornaliero è il costo di una giornata di lavoro del dipendente (potete assumere che il costo di una giornata di lavoro non supererà mai 1000).

La lista **l** contiene le specifiche di un'offerta economica per una gara di appalto a cui la software house ha intenzione di partecipare ed ha il seguente formato:

```
[Competenza1,NumeroGiornate1,..., NumeroGiornatem,Competenzam]
```

Dove, per ogni valore di j tra 1 ed m , Competenza _{j} è la competenza richiesta e NumeroGiornate _{j} indica il numero di giornate necessarie di un dipendente che abbia la competenza richiesta. La funzione deve calcolare (e restituire) l'offerta migliore che la software house può presentare, cioè quella di costo minimo e che rispetta quante più specifiche possibili. Più precisamente, la funzione deve restituire un dizionario con chiavi i nomi delle competenze richieste nella specifica della gara, e valore (per ciascuna chiave) il nome del dipendente selezionato per quella competenza (cioè quello con quella competenza, con disponibilità di giornate almeno uguale a quella richiesta e di costo minore). Se per una competenza richiesta non c'è alcun dipendente che soddisfa la richiesta (nessuno ha quella competenza oppure nessuno ha la disponibilità) allora il valore associato a quella competenza nel dizionario deve essere 'Nessuno'. Ad esempio, se il file **file** contiene:

```
Carlo,Java,100,500  
Giulia,DataBases,35,420  
Flavia,Python,50,330  
Flavia,Java,50,400  
Michele,DataBases,25,450
```

e la lista **l** vale ['Java', 60, 'Python', 10, 'DataBases', 40], allora la funzione deve restituire {'Java': 'Carlo', 'Python': 'Flavia', 'DataBases': 'Nessuno'}.

Nota: Potete assumere che le giornate di lavoro ed i costi nei file **file1** e **file2** siano sempre convertibili in numeri interi.