

06.07.2016

Esame di Fondamenti Informatica per Ingegneria Gestionale - A.A. 2015/2016

Durata 2h - Compito B

Istruzioni (leggere attentamente)

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

Registrazione dei dati dello studente

Prima di iniziare il compito, eseguire attivaesame.bat che si trova sul Desktop. Aprire poi la cartella T:\Esame, sempre sul Desktop. Eseguire il programma registrazione.pyc, inserire i dati personali fornendo (separatamente) Numero di Matricola, Cognome e Nome. Il programma genera un file studente.txt che non deve essere modificato manualmente. Verificare che i dati nel file studente.txt siano corretti; in caso di errore potete rieseguire il programma registrazione.pyc.

Svolgimento degli esercizi

Leggere attentamente il testo e risolvere gli esercizi proposti.

Per ogni esercizio avete una cartella EsercN che contiene un file dal nome B_S_ExN.py (dove N è il numero dell'esercizio) con lo scheletro della soluzione. Non create nuovi file.

Per verificare la correttezza di un esercizio **DOVETE** usare il programma TestEx.pyc (basta cliccarci sopra 2 volte) che proverà la vostra soluzione con un certo numero di casi di test. **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

- **B_Ex1(stringa) (8 punti)** Scrivere una funzione che prende in ingresso una stringa e restituisca una lista di liste ove ciascuna sottolista è composta da due elementi, il primo elemento è il carattere della stringa mentre il secondo elemento è il numero di volte, in formato stringa, che il carattere compare nella stringa. Ad esempio se la stringa data è "abaco" la funzione deve restituire [['a','2'], ['b','1'], ['c','1'], ['o','1']] Se la stringa in ingresso è vuota allora la funzione deve restituire la lista vuota. Si noti, come mostrato nell'esempio, che il carattere 'a' compare due volte ma nella lista di uscita vi è una sola sottolista associata al carattere il cui numero di occorrenze però è '2'.
- **B_Ex2(nomeFile) (8 punti)** Scrivere una funzione che restituisca una matrice i cui valori sono memorizzati in un file. La matrice è rappresentata nel file nel seguente modo:
 - ogni riga del file è composta da una stringa del tipo (indice_riga,indice_colonna)=valore;
 - gli indici di riga e colonna partono da 1 (non da zero);
 - il file contiene solo dati relativi agli elementi della matrice che sono diversi da zero.

Esempio:

Se il file contiene le seguenti righe

(1,1)=2

(1,4)=1

(2,3)=3

(2,5)=1

(3,2)=1

(3,4)=1

(3,5)=2

allora la funzione deve restituire la seguente matrice $[[2,0,0,1,0],[0,3,0,1],[0,1,0,1,2]]$

Si noti, come mostrato nell'esempio, che per le coppie di indici non presenti nel file i valori della matrice devono essere posti uguali a zero.

- **B_Ex3(file) (8 punti)** Scrivere una funzione python che riceve in ingresso il nome di un file csv contenente i viaggi effettuati dalle linee metropolitane di Roma nel formato:

Linea,Persone,Fermata

La funzione deve restituire un dizionario contenente per ogni linea della metro una lista con la progressione delle sue fermate e il totale dei passeggeri trasportati fino a quel momento. Il numero di persone potrà essere specificato come un intero positivo (persone che salgono) o negativo (persone che scendono). Considerare come capienza massima della metro il numero di 150 Persone. Ad esempio, se l'input è:

Linea,Persone,Fermata
metro A,20,1
metro A,50,2
metro B,100,1
metro B,40,2
metro B,-40,3

La funzione deve restituire il dizionario:

`{'metro B': ['1', 100, '2', 140, '3', 100], 'metro A': ['1', 20, '2', 70]}`

- **B_Ex4(g, k) (8 punti)** Implementare la funzione **B_Ex4(g, k)** che riceve in ingresso un dizionario *g* che rappresenta un grafo *diretto* i cui nodi sono rappresentati da etichette *intere* e un intero *k*. La funzione deve restituire la lista ordinata dei nodi che hanno *almeno* *k* archi entranti. Ad esempio, se il grafo fosse $g = \{1: [4, 2, 3], 2: [3, 4], 5: [1, 2, 3, 4], 4: [3], 3: [4]\}$ e $k = 3$, la funzione dovrebbe restituire la lista $[3, 4]$, perché soltanto questi nodi hanno almeno 3 archi entranti (entrambi hanno esattamente 4 archi entranti).

Si noti che un nodo privo di archi uscenti non è presente come chiave nel dizionario.