

10.07.2017

Esame di Fondamenti Informatica per Ingegneria Gestionale - A.A. 2016/2017

Durata 1h45' - Compito A

Istruzioni (leggere attentamente)

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

Cartella di esame e registrazione dei dati dello studente

La cartella Esame contenente il compito da svolgere si trova sul Desktop. Entrare nella cartella e, prima di iniziare il compito, 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 A_ExN.py (dove N è il numero dell'esercizio). Questo è il file che dovrete modificare con la vostra 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

- **A_Ex1(s) (8 punti)** Completare la funzione **A_Ex1(s)** che prende come input una stringa *s* e restituisce la posizione del massimo carattere all'interno della stringa *secondo l'ordinamento definito da Python*. Se *s* contiene più volte il carattere massimo allora la funzione restituisce la posizione minore. Se *s* è la stringa vuota allora la funzione deve restituire -1.

Esempio: Se *s*='ABCabc' allora la funzione deve restituire 5, poiché il carattere massimo in questo caso è 'c' e compare una sola volta, in posizione 5.

- **A_Ex2(l,s)** Scrivere una funzione che riceve in input una lista *l* di stringhe ed una stringa *s* e restituisce una lista fatta nel modo seguente: per ogni stringa *st* in *l* la funzione calcola il numero *n* di caratteri di *s* che compaiono in *st* e si scrive nella nuova lista prima *st* e poi *n*. Ad esempio: se *l* = ["fff", "vvf", "merc"] e la stringa *s* = "fvf" allora la funzione dovrà restituire la lista ["fff", 2, "vvf", 3, "merc", 0]. Se la lista *l* è vuota la funzione dovrà restituire una lista vuota. Se la stringa *s* in ingresso è vuota la funzione dovrà restituire la nuova lista ove dopo ogni stringa di *st* in *l* sarà seguita da uno zero.
- **A_Ex3(file1, file2) (8 punti)** Implementare la funzione **A_Ex3(file1, file2)** che prende in ingresso i nomi di due file, ciascuno contenente testo privo di punteggiatura. La funzione deve restituire la *lista ordinata* delle parole (in lettere minuscole) che compaiono sia in **file1** che in **file2**. Eventuali maiuscole devono essere ignorate, per cui ad esempio "Casa" e "casa" vanno considerate la stessa parola. Ogni parola in comune deve essere considerata *una volta sola*.

Esempio: si supponga che **file1** contenga il testo:

“Sopra la panca la capra campa sotto la panca la capra muore”

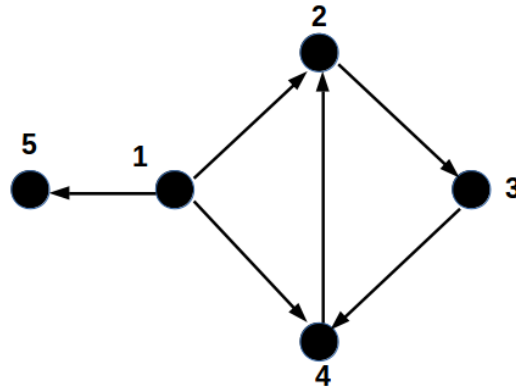
e che **file2** contenga il testo:

“Sotto la panca dorme un gatto”

La funzione deve restituire la lista [“la”, “panca”, “sotto”].

Nota: si ricordi che, data una stringa, il metodo `lower` restituisce una copia della stringa con caratteri minuscoli.

- **A_Ex4(g, u) (8 punti)** Implementare la funzione **A_Ex4(g, u)** che (tra gli altri parametri) riceve in ingresso un dizionario **g** che rappresenta un grafo *diretto*, i cui nodi hanno etichette *interi*. Il parametro **u** è un nodo del grafo. La funzione deve restituire la somma delle etichette dei nodi che hanno **u** come vicino.



Ad esempio, se il grafo fosse quello in figura, avremmo $g = \{1: [4, 2, 5], 2: [3], 3: [4], 4: [2], 5: []\}$. Se **u** fosse il nodo 1 la funzione dovrebbe restituire 0, mentre se **u** fosse il nodo 2, la funzione dovrebbe restituire il valore 5.