

Fondamenti di Informatica per Ing. Gestionale

Esercitazione del 17 Maggio 2018

Istruzioni

Collegarsi al sito del corso <https://piazza.com/uniroma1.it/spring2018/1017401/home> e scaricare i file necessari all'esercitazione dalla sezione risorse. I file sono compressi in un file `.zip`, contenente una cartella con i file dell'esercitazione.

Svolgimento degli esercizi - Leggere attentamente

Leggere attentamente il testo e risolvere gli esercizi proposti.

Per ogni esercizio avete unacartella EsercN che contiene un file dal nome A_ExN.py (dove N è il numero dell'esercizio) con lo scheletro della soluzione. Il file A_S_ExN.py contiene la soluzione dell'esercizio proposta dal docente. *E' nel vostro diretto interesse non aprire quest'ultimo file prima di aver dedicato tutto il tempo disponibile a risolvere autonomamente l'esercizio.* Non create nuovi file.

Per verificare la correttezza di un esercizio **DOVETE** usare il programma TestEx.py (che non va assolutamente modificato), che proverà la vostra soluzione su un certo numero di casi di test, mostrandovi il confronto tra i risultati ottenuti con la vostra soluzione e usando la soluzione proposta dal docente. Ciò ha lo scopo di facilitarvi nella ricerca e correzione di eventuale errori

1. Scrivere una funzione `A_Ex1(s)` che, ricevuta in ingresso una stringa s, calcoli la lunghezza della più lunga sequenza di caratteri successivi in ordine (strettamente) crescente. Ad esempio se la s vale 'abclmnop' allora la funzione deve restituire 5, poiché la sequenza 'lmnop' in s è di lunghezza 5 ed è la più lunga ordinata in modo (strettamente) crescente.
2. Scrivere una funzione `A_Ex2(file)` che, ricevuto in ingresso il nome di un file che contiene n interi per ogni riga separati da spazio, restituisca l'intero più piccolo e l'intero più grande. Ad esempio, se il contenuto del file fosse "3 7\n55 17\n8 23\n99 12\n6 21\n" la funzione dovrebbe restituire (3,99).
3. Completare la funzione Python `A_Ex3(file, c)` che riceve in ingresso una matrice di stringhe m e un carattere c e restituisce l'indice della riga i in cui il carattere c occorre con maggior frequenza. Se il carattere c non è presente in alcuna riga di m, allora la funzione deve restituire il valore -1. Se il carattere c appare su più righe con la stessa frequenza massima, allora la funzione deve restituire l'indice più piccolo tra questi.

Ad esempio, se la matrice fosse:

$$\begin{pmatrix} Ciao & Bye \\ Casa & Canale \end{pmatrix}$$

e il carattere c fosse 'a' la funzione dovrebbe restituire 1, mentre se c fosse il carattere 'z' la funzione dovrebbe restituire -1.

4. Scrivere una funzione `B_Ex1(l, n)` che, ricevuti in ingresso una lista di interi `l` e un intero `n`, restituisca la lunghezza della sequenza più lunga di interi consecutivi di `l` che sono maggiori o uguali ad `n`. Ad esempio, se `l` vale `[15, 15, 80, 21, 6, 5, 6, 21]` e `n` vale 15 la funzione deve restituire 4.
5. Completare la funzione `B_Ex2(file_in)` che, ricevuto in ingresso il nome di un contenente `n` interi per ogni riga separati da `;` restituisca una lista i cui elementi corrispondono alla somma dei numeri presenti su ogni riga del file. Ad esempio, se le righe del file fossero `"3;4;5"` e `"9;2;6"`, la funzione dovrebbe restituire la lista `[12, 17]`.
6. Completare la funzione Python `B_Ex3(m, c)` che, ricevuta in ingresso una matrice di stringhe `m` e un carattere `c`, restituisca l'indice della colonna `j` in cui il carattere `c` occorre con la frequenza minima (maggiore di 0). Se il carattere `c` non presente in alcuna colonna di `m`, allora la funzione deve restituire il valore -1. Se il carattere `c` appare su più colonne con la stessa frequenza minima, allora la funzione deve restituire l'indice più grande tra questi.

Ad esempio, se la matrice fosse:

$$\begin{pmatrix} Ciao & Bye \\ Casa & Canale \end{pmatrix}$$

e il carattere `c` fosse 'b' la funzione dovrebbe restituire 1, mentre se `c` fosse il carattere 'z' la funzione dovrebbe restituire -1.