

Sapienza Università di Roma  
Corso di Laurea in Ingegneria Informatica e Automatica  
A.A. 2018-19  
**Compito d'esame -- 11 settembre 2019 -- Compito A**

## 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, 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`.

**Svolgimento degli esercizi:** Per ogni esercizio avete una cartella `EsercN` che contiene un file dal nome `A_ExN.py` (dove **N** è il numero dell'esercizio) con lo scheletro della soluzione. Questo file incorpora un codice di test che proverà la vostra soluzione per un certo numero di possibili dati in input. Aprite il file con Spyder e modificate SOLO il contenuto della funzione. Eseguendo il file `.py` si otterrà il responso dei test sulla console. 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.**

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.

**Al termine** dello svolgimento del compito d'esame **NON è necessario effettuare una procedura di consegna** dell'elaborato. Le vostre soluzioni saranno copiate direttamente dalla cartella `Esame` della macchina su cui avete lavorato. Quindi, per concludere la prova, è sufficiente chiudere il programma Spyder e lasciare la vostra postazione.

## Esercizi

- **A\_Ex1(s,n)** Scrivere una funzione che prende in ingresso una stringa `s` di caratteri alfabetici ed un intero positivo `n` e restituisce il numero di caratteri che hanno esattamente `n` occorrenze in `s`. Non si faccia distinzione fra maiuscole e minuscole. Ad esempio, se `s='ossessivo'` ed `n=4` la funzione deve restituire 1, se `s='Anna'` ed `n=2` la funzione deve restituire 2, se `s='amico'` ed `n=1` la funzione deve restituire 5.
- **A\_Ex2(M)** Scrivere una funzione che prende in ingresso una matrice `M` (rappresentata come lista di liste) contenente numeri interi e restituisce una matrice delle stesse dimensioni (sempre rappresentata come lista di liste) contenente gli stessi elementi di `M` ordinati in ordine crescente da sinistra a destra e dall'alto in basso. Si assuma che `M` contenga almeno un elemento. Ad esempio, se `M` è la matrice

3	4	3
2	1	2

la funzione deve restituire la matrice

1	2	2
3	3	4

- **A\_Ex3(file)** Scrivere una funzione che prende in ingresso un file di testo contenente solo righe che contengono parole (composte di sole lettere minuscole) e restituisce la parola più lunga che compare in almeno 3 righe diverse. Se ci sono più parole della stessa lunghezza massima, restituite quella alfabeticamente più piccola. Se nessuna parola compare in almeno 3 righe, restituite la stringa vuota. Ad esempio, il file contiene:

```
palla casa giovane albero
casolare casata albero
```

cielo palla casa  
gomma albero gomma palo  
mare palla casa

La funzione deve restituire la stringa 'albero'.

- **A\_Ex4(file)** Scrivere una funzione che prende in ingresso il nome di un **file** di testo contenente (una per riga) le scommesse raccolte da una società di scommesse sportive su un insieme di partite. Ogni riga ha il seguente formato

**CodicePartita,Scommessa,Cifra,Quota**

dove **CodicePartita** è un numero intero, **Scommessa** è una tra 1X2, **Cifra** è un numero intero in euro, mentre **Quota** è la quota che l'agenzia deve pagare PER OGNI EURO SCOMMESSO se lo scommettitore vince la scommessa. Per semplicità, assumete che la quota sia un numero intero.

La vostra funzione deve costruire un dizionario con chiave il **CodicePartita** e valore una lista di 4 valori, cioè il totale scommesso su quella partita, la quota che l'agenzia deve complessivamente pagare se si verifica il risultato 1, la quota che l'agenzia deve complessivamente pagare se si verifica il risultato X, la quota che l'agenzia deve complessivamente pagare se si verifica il risultato 2,

Ad esempio, se il file contiene:

12,1,10,2  
12,X,5,2  
12,2,12,3

allora la funzione deve restituire il dizionario {12: [27, 20, 10, 36]}