

## Compito d'esame -- 20 gennaio 2021 -- 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 i dati di test diversi**.

E' possibile consultare la documentazione ufficiale del linguaggio Python (file:///Library/Frameworks/Python.framework/Versions/3.9/Resources/English.lproj/Documentation/index.html),

ma **non è possibile usare libri o appunti**. In caso di problemi tecnici chiedere ai docenti o ai tecnici del laboratorio.

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(I)** Scrivere una funzione che riceve in ingresso una lista `I` contenente liste di stringhe (che sono quindi sottoliste di `I`) e restituisce una stringa composta dai caratteri che compaiono complessivamente il maggior numero di volte in ogni sottolista. Si assuma che `I`, ogni sottolista di `I`, ed ogni stringa contenuta in queste sottoliste siano non vuote. Si assuma anche che in ogni sottolista ci sia sempre un solo carattere che compare il maggior numero di volte. Nella stringa restituita i caratteri devono seguire lo stesso ordine che hanno in `I` le sottoliste da cui vengono estratti. Ad esempio, se  

```
l=[['amare','mare','fermata'], ['accordare','fare','dire']]
```

la funzione deve restituire 'ar', in quanto la a compare cinque volte nella prima sottolista, e non c'è un carattere che compare più di cinque volte nella prima sottolista, e la r compare quattro volte nella seconda sottolista, e non c'è un carattere che compare più di quattro volte nella seconda sottolista.

- **Ex2 (file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **file** contenente al suo interno del testo che include anche degli indirizzi email che hanno la seguente forma (semplificata rispetto alla realtà):

mailto:pippo@pluto.it

cioè iniziano sempre per mailto: seguito dal *nome*, che è una sequenza non vuota di caratteri alfanumerici o il carattere '\_' (underscore), seguito dal carattere @, seguito dall'*organizzazione*, che è una sequenza non vuota di caratteri alfanumerici o underscore, seguito dal punto, seguito da una *sigla*, che è una sequenza di soli caratteri alfabetici avente come valore uno fra org, com, net, oppure l'abbreviazione del nome di una nazione (ad esempio it si riferisce all'Italia).

La funzione deve leggere il file di testo e restituire un dizionario che ha come chiavi le varie nazioni (la sigla finale dell'email) e come valore la lista, *ordinata alfabeticamente*, dei nomi che compaiono in indirizzi email corretti (che quindi rispettano la forma indicata in precedenza) relativi a quella nazione. Ad esempio, se il file contiene il seguente testo:

Ieri ho scritto una email a mailto:domenico@mionet.it ed un'altra a mailto:pippo@topolinia.com  
Il giorno prima invece avevo scritto a mailto:maria@maria.it ed a mailto:marco@onair.es  
mentre l'email anna@mionet.it ha un formato sbagliato e non dovete considerarla

la funzione deve restituire {'it':['domenico','maria'],'es':['marco']}. Notate che mailto:pippo@topolinia.com non va considerato nel risultato perchè termina con com, mentre anna@mionet.it non va considerato perchè ha un formato errato.

- **Ex3(l,file)** Scrivere una funzione che prende in ingresso una lista contenente i punteggi di un gioco di carte dal più piccolo al più grande ed un file di testo **file** in formato csv che contiene i risultati di partite con due giocatori, secondo lo schema:

Giocatore1,Giocatore2,PuntoGiocatore1,PuntoGiocatore2

la funzione deve restituire la lista contenente i nomi dei giocatori, *ordinati alfabeticamente*, che hanno vinto il massimo numero di partite. Ad esempio, se

**l**=[coppia,doppiacoppia,tris,scala,full,poker,colore,scalareale]  
e **file** contiene:

Giocatore1,Giocatore2,PuntoGiocatore1,PuntoGiocatore2  
Marco,Aldo,coppia,tris  
Antonio,Marco,tris,scala  
Anna,Marco,scala,tris  
Aldo,Anna,poker,scala  
Antonio,Marco,coppia,scala

allora la funzione deve restituire ['Aldo','Marco']