

Compito d'esame -- 20 gennaio 2023 -- Compito C

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 il programma `REGISTRAs studente.py` 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 `REGISTRAs studente.py`.

Tempo a disposizione: 1 ora e 45 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 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.

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(s)** Scrivere una funzione che riceve in ingresso una stringa `s` e calcola la lunghezza della più lunga sottostringa di `s` lunga almeno due caratteri e in cui solo il primo e l'ultimo carattere sono una vocale italiana ('aeiou'). Ad esempio, se la stringa `s` vale `'cavalloaltrimenti'`, la funzione deve restituire 5 poiché `'altri'` è la sottostringa più lunga in cui solo il primo e l'ultimo carattere sono una vocale. N.B. La funzione non deve distinguere tra caratteri maiuscoli e minuscoli.
- **Ex2(file,s)** Scrivere una funzione che riceve in ingresso il nome di un file di testo `file`, che potete assumere contenga solo caratteri alfabetici, spazi e `'\n'`, e una stringa `s` contenente solo caratteri alfabetici (cioè una parola) e restituisce un dizionario con chiavi i numeri di riga (contate a partire da 1) e valore associato a ciascuna chiave `k` la lista delle parole della riga `k` che contengono tutte e sole le vocali italiane ('aeiou') di

s (incluse ripetizioni). Non dovete fare differenza tra lettere maiuscole e minuscole, quindi 'Anna' e 'casa' contengono le stesse vocali. Ad esempio, se il file **file** contiene:

Tanto va la gatta
Al lardo lorda
Datato rase

Ed s vale 'lasso' allora la funzione deve restituire {1: ['Tanto'], 2: ['lardo', 'lorda'], 3: ['Datato']}.

- **Ex3(file)** Scrivere una funzione che prende in ingresso un file di testo in formato csv, **file**. Il file **file** contiene le informazioni su partite:

G_1, G_2, \dots, G_N

dove G_i identifica un generico giocatore, $3 < N \leq 10$ e ogni riga corrisponde a una partita a dadi, riportando il punteggio di ciascun giocatore ottenuto tirando due dadi. Poiché ogni tiro di dado può dare come risultato un numero tra 2 e 12, ogni riga in cui ci sia anche un solo numero che non è compreso in questo intervallo va considerata non corretta e quindi ignorata.

La funzione deve restituire un insieme che contiene i nomi dei 3 giocatori che hanno vinto più partite corrette (dove una partita è corretta se contiene solo punteggi tra 2 e 12 ed è vinta quando un giocatore ottiene un punteggio strettamente più grande di quello di tutti gli altri). Nel caso in cui per arrivare a 3 sia necessario effettuare una scelta tra 2 o più giocatori a pari merito, si sceglierà il giocatore(i) con indice(i) più alto(i). Ad esempio, se il file **file1** contiene:

G_1, G_2, G_3, G_4
9, 5, 3, 8
7, 6, 2, 6
2, **3**, 2, 2
3, 7, 4, 1
2, 4, **5**, 2

allora la funzione deve restituire { G_1, G_2, G_3 } . Infatti, la partita 4 non è corretta poiché uno dei punteggi è 1 e va quindi ignorata, G_1 ha vinto 2 partite, G_2 una partita, G_3 una partita e G_4 nessuna .

Domande veroFalso

Ex4 Il file Ex4.py contiene la funzione veroFalso() che stampa 8 domande sullo schermo. La funzione deve essere modificata cambiando il valore del return, elencando le lettere delle domande che ritenete essere vere. Ad esempio, se ritenete che le domande B e C sono vere il return deve essere modificato in

```
return 'BC'
```