

Compito d'esame -- 8 febbraio 2023 -- 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 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(v,i1,i2)** Scrivere una funzione che riceve in ingresso un vettore di interi **v** e due interi **i1** e **i2** e calcola la lunghezza della più lunga sequenza di interi consecutivi di **v** che contiene lo stesso numero (maggiore di zero) di **i1** e **i2**. Ad esempio, se **v** vale `[9,5,7,33,9,5,5,5,8,5,33,33,6,15,8,5,6]` e **i1** = 33 e **i2** = 5, la funzione deve restituire 9 perché esiste una sequenza di lunghezza 9 avente la caratteristica richiesta (quella in grassetto) e non c'è in **v** una sequenza di questo tipo che abbia lunghezza superiore.
- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **file**, e conta in quante righe è presente una sequenza di 3 parole consecutive in cui la parola di mezzo comincia con l'ultimo carattere di quella che la precede e finisce con il primo carattere di quella che la segue. Se ad esempio il file è il seguente:

Sono onde esagerate e fredde
Metto leso
Pelle rotta
Pesce eletto ora e panna verde

Allora la funzione deve restituire 2 poiché nella prima e nell'ultima riga ci sono 3 parole consecutive con la caratteristica richiesta. N.B.: l'esercizio va risolto usando le espressioni regolari.

- **Ex3(file)** Scrivere una funzione che prende in ingresso un file di testo in formato csv **file** che contiene le informazioni (non realistiche) sul cambio dell'euro con alcune valute straniere (raggruppate per giornata dell'anno, indicata con un intero) e calcola un dizionario con chiave il nome della valuta e valore il giorno in cui la valuta ha avuto l'aumento maggiore rispetto al suo valore precedente. Più precisamente, il file ha il seguente formato:

```
giorno, valuta1, cambio1, ..., valutan, cambion
```

dove `giorno` è un numero tra 1 e 365 (potete assumere che i giorni siano presenti nel file in ordine), `valuta1` è il nome di una valuta straniera e `cambio1` è il valore del cambio di quella valuta in quella giornata; il numero `n` di valute non è fissato e può cambiare di giorno in giorno. Se una valuta, apparsa in precedenza, non compare per qualche giorno, questo vuol dire che il suo cambio è rimasto invariato in quei giorni. Se ci sono più giorni in cui si verifica l'aumento massimo, selezionate il primo presente nel file. Se una valuta non subisce mai un aumento allora dovete restituire 'stabile'. Ad esempio, se il file **file** contiene:

```
10, dollaro, 2.0, sterlina, 1.0  
20, rublo, 0.5, sterlina, 1.1  
30, sterlina, 1.2, dollaro, 2.3
```

allora la funzione deve restituire `{dollaro: 30, sterlina: 20, rublo: 'stabile'}`, poiché il dollaro ha il prezzo di 2.0 il 10 e il 20 (prezzo stabile perché non compare) e poi aumenta di 0.3, la sterlina ha il prezzo massimo il 30 ma l'aumento 0.1 è lo stesso il 20 e il 30 e quindi il valore corretto è il 20, mentre il rublo viene quotato il 20 e non subisce mai un aumento.

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'
```