

Compito d'esame -- 27 gennaio 2022 -- Compito A

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 `l` contenente numeri interi e restituisce la lunghezza della più lunga sequenza di interi in ordine *crescente* in `l`. Ad esempio, se `l=[1, 2, 3, 3, 1, 4, 5, 1, 7]`, la funzione deve restituire 3 perchè tre è la lunghezza della più lunga sequenza in `l` avente la caratteristica richiesta (in questo esempio ce ne sono due di lunghezza massima, `1, 2, 3` e `1, 4, 5`) e non c'è in `l` una sequenza di questo tipo che abbia lunghezza 4 o superiore.

- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **file**, che può contenere solo caratteri alfanumerici spazi ed andate a capo `'\n'`. Si individuino in **file** le sequenze di sole cifre (0,1,2,3,4,5,6,7,8,9), e che quindi possono essere trasformate in numeri interi positivi, e si restituisca un dizionario con chiavi tali sequenze (trasformate in interi) e valore per ciascuna chiave la lista *ordinata in ordine decrescente* degli indici di riga (contate a partire dall'indice 1) in cui compare la sequenza corrispondente alla chiave. Si noti che nella trasformazione di una stringa in intero gli eventuali 0 iniziali non modificano il numero ottenuto: ad esempio 0039 e 39 sono due stringhe che rappresentano lo stesso numero. Se ad esempio il file è il seguente

Marco ha 11 anni e possiede 3 palloni
ognuno dei 3 palloni ha disegnato il simbolo 1
mettendo i 3 palloni vicini si ottiene il numero 111

la funzione deve restituire `{11:[1], 3:[3,2,1], 1:[2], 111:[3]}`

- **Ex3(file1,file2)** Scrivere una funzione che prende in ingresso due file di testo (in formato csv), il file **file1** contiene le informazioni sugli oggetti disponibili in un negozio online all'inizio dell'anno. Il file csv **file1** ha il seguente formato:

Oggetto,Quantità,MinQuant,Prezzo

dove **Oggetto** è il nome di un oggetto del negozio, **Quantità** è la quantità inizialmente disponibile all'inizio dell'anno, **MinQuant** è la quantità minima che si vorrebbe avere in magazzino e **Prezzo** è il prezzo dell'oggetto all'inizio dell'anno.

Il file **file2** contiene gli acquisti e le vendite effettuate dal negozio durante l'anno ed ha il seguente formato:

Oggetto,Quantità

Dove **Oggetto** è il nome dell'oggetto acquistato o venduto e **Quantità** è un valore (positivo o negativo) che indica la quantità acquistata (se positivo) o venduta (se negativo). Se il numero di oggetti venduti è maggiore di quelli disponibili, la quantità disponibile diventa 0 e bisogna ricordarsi quanti oggetti non sono stati venduti (numero venduti – numero disponibili). Gli acquisti in questo file sono nell'ordine in cui sono avvenuti. Dopo ogni transazione, se la quantità disponibile di un oggetto *diventa* strettamente inferiore alla quantità minima prevista per quell'oggetto (nel senso che prima della transazione **Quantità** è maggiore o uguale a **MinQuant** ma dopo diventa minore), il suo prezzo viene aumentato di 5, se invece la quantità disponibile *torna ad essere* superiore della quantità minima prevista, il suo prezzo viene diminuito di 5. La funzione deve restituire un dizionario con chiave gli oggetti e valore una lista contenente la quantità disponibile, il numero di pezzi che non si è potuto vendere per mancanza di disponibilità ed il prezzo alla fine di tutti gli acquisti. Ad esempio, se il file **file1** contiene:

Iphone,10,5,500
Ipad,6,3,700
Ps5,5,0,500

e il file **file2** contiene:

Iphone,6
Ipad,-5
Ipad,-3

allora la funzione deve restituire `{'Iphone': [16, 0, 500], 'Ipad': [0, 2, 705], 'Ps5': [5, 0, 500]}`.