Sapienza Università di Roma
Corso di Laurea in Ingegneria Informatica e Automatica A.A. 2022-23

Compito d'esame -- 20 gennaio 2023 -- 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 il programma `REGISTRAstudente.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 `REGISTRAstudente.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 insiemi 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(l)** Scrivere una funzione che riceve in ingresso una lista **l** contenente coppie (tuple) di numeri interi e restituisce la lunghezza della più lunga sequenza di coppie con la proprietà che il secondo numero della coppia a sinistra è uguale al primo numero della coppia a destra. Ad esempio, se **l**=`[(4,1),(1,2),(3,3),(3,5),(5,1),(1,4),(5,2),(2,1),(1,7)]`, la funzione deve restituire `4` perché quattro è la lunghezza della più lunga sequenza in **l** avente la caratteristica richiesta (in grassetto) e non c'è in **l** una sequenza di questo tipo che abbia lunghezza 5 o superiore. Se la lista è vuota, la funzione deve restituire 0, se contiene una sola tupla 1.

- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **file**, e analizzi gli indirizzi email presenti in esso. Si assuma (semplificando un poco) che gli indirizzi email corretti abbiano il seguente formato:

  partesinistra@dominio

  Dove la partesinistra è composta da 1 o più sequenza NON VUOTE di caratteri alfanumerici (lettere o cifre) separate dal carattere punto (`'.'`) ed il dominio contiene solo caratteri alfanumerici ed il carattere punto (`'.'`). Più precisamente, bisogna costruire un dizionario con chiave i vari domini presenti negli indirizzi email corretti del file e valore una lista di 3 numeri che indicano il numero di indirizzi email corretti di quel dominio che sono composti da 1, 2 o più di 2 sequenze. Se ad esempio il file è il seguente

  Ho scritto una mail a johnny@gmail.com ed una a paolo..carmo@uniroma1.it ed altre
  a johnny.carson@gmail.com e rossi.2222222@studenti.uniroma1.it
  Dopo ne ho ricevuta una da bianchi.11111.@studenti.uniroma1.it ed una da johnny.carson.jr@gmail.com

  la funzione deve restituire {'gmail.com': [1, 1, 1], 'studenti.uniroma1.it': [0, 1, 0]} poiché le mail paolo..carmo@uniroma1.it e bianchi.11111.@studenti.uniroma1.it NON sono corrette poiché hanno delle sequenze VUOTE.

- **Ex3(file,l)** Scrivere una funzione che prende in ingresso un file di testo in formato csv **file** ed una lista **l**. Il file **file** contiene le informazioni sui programmi che devono essere eseguiti da un centro di calcolo e la lista **l** le informazioni sul computer disponibile per farli girare. Il file csv **file** ha il seguente formato:

  `NomeProgramma,SistemaOperativo,QuantitàRAM`

  dove `NomeProgramma` è il nome del programma da eseguire, `SistemaOperativo` è il nome del sistema operativo su cui questo programma deve girare e `QuantitàRAM` è la quantità minima di RAM (intera e misurata in GB) necessaria per far girare il programma.

  La lista **l** ha il seguente formato `[SistemaOperativo,NumeroCores,QuantitàRAMdisponibile]`

  Dove `SistemaOperativo` è il sistema operativo del computer, `NumeroCores` è il numero di Cores presenti nel computer e `QuantitàRAMdisponibile` è il numero (intero) di GB di RAM complessivamente disponibili su quel computer. Ogni programma deve essere eseguito su un core del computer secondo la seguente regola: I programmi vanno assegnati ai core seguendo l'ordine in cui si trovano nel file, devono essere assegnati solo se il computer ha il sistema operativo adatto ed un numero di GB disponibili sufficienti (ovviamente dopo che un programma è stato assegnato ad un core, quel core non è più disponibile ed i GB disponibili complessivamente scendono). Se nessun core soddisfa i criteri vuol dire che il programma non si può eseguire.
  La funzione deve restituire un dizionario con chiave il nome del programma e valore il numero del core (da 1 a `NumeroCores`) su cui viene fatto girare. Se nessun core è disponibile per quel programma allora il valore sarà `'Nessuno'`. Ad esempio, se il file **file** contiene:

  ```
  prog1,MacOS,5
  prog2,Windows,2
  prog3,Windows,5
  prog4,Linux,2
  prog5,Windows,1
  ```

  e la lista **l** vale `[Windows,2,8]`, allora la funzione deve restituire `{'prog1': 'Nessuno', 'prog2': 1, 'prog3': 2, 'prog4': 'Nessuno', 'prog5': 'Nessuno'}`, poiché `prog1` e `prog4` hanno bisogno di un sistema operativo non disponibile sul computer e `prog5` non ha più core disponibili per essere eseguito.

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'
```