

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 `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(I)** 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.222222@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'
```

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'
```

Compito d'esame -- 13 luglio 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 `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 i 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

N.B.: Gli studenti con DSA non devono svolgere il secondo esercizio

- **Ex1(s,c)** Scrivere una funzione che riceve in ingresso una stringa `s` e il carattere `c` e restituisce il numero di volte in cui carattere `c` è seguito da un carattere maggiore in ordine alfabetico. Ad esempio, se `s='mattinata tremenda'` e `c='t'` la funzione deve restituire 1. N.B. se il carattere `c` non compare nella stringa la funzione deve restituire -1.
- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo `file`, contenente solo caratteri alfanumerici minuscoli non accentati e caratteri di spaziatura. La funzione deve restituire un

dizionario in cui le chiavi sono i caratteri minuscoli dell'**ultima** parola del file e il valore il numero di occorrenze del carattere nel file, escludendo l'ultima parola e tutte le parole uguali all'ultima parola. Se ad esempio il **file** è il seguente

```
abbiamo trovato il pallone
```

la funzione deve restituire {'p': 0, 'a': 3, 'l': 1, 'o': 3, 'n': 0, 'e': 0}.

- **Ex3(file,nome)** Scrivere una funzione che prende in ingresso un file di testo in formato csv **file** che contiene i risultati di alcune partite tra giocatori ed il **nome** di un giocatore. Il file csv **file** ha il seguente formato:

```
giocatore1,giocatore2,risultato
```

dove il risultato è uno tra **1X2**, dove **1** significa che ha vinto il primo giocatore, **X** che è finita in pareggio e **2** che ha vinto il secondo giocatore.

La funzione deve restituire l'insieme di tutti i giocatori che hanno vinto il massimo numero di partite contro il giocatore **nome**. Ad esempio, se il **file** contiene:

```
Paolo,Gianni,1  
Piero,Gianni,1  
Gianni,Paolo,X  
Gianni,Piero,2
```

e il nome è Gianni allora la funzione deve restituire l'insieme {Piero}

Domande veroFalso

- **Ex4** Il file Ex4.py contiene la funzione veroFalso() che stampa 8 domande sullo schermo. La funzione deve essere modificata **cambiando solo 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'
```

Compito d'esame -- 24 gennaio 2024 -- Compito A
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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)** In una lista `l` contenente stringhe si definisce *coppia di gemelle* una coppia di stringhe con indici a distanza `k`, `<l[i], l[i+k]>` che hanno entrambe la stessa lunghezza `k`, dove `k>0`. La stringa `l[i]` viene indicata come gemella minore e quella `l[i+k]` gemella maggiore. Scrivere una funzione che riceve in ingresso una lista `l` contenente stringhe e restituisce il numero di coppie gemelle, escludendo dal conteggio le coppie in cui l'indice della gemella minore coincide con l'indice di una gemella maggiore di un'altra coppia. Per esempio, se `l=['io', 'mio', 'tu', 'mio', 'il', 'mio']` la funzione deve restituire 1 perché la lista contiene due coppie gemelle `<l[0],l[2]>` e `<l[2],l[4]>` ma la seconda coppia non va contata perché l'indice della gemella minore coincide con quello della gemella maggiore della prima. Se la lista è vuota, la funzione deve restituire -1.

- **Ex2(fileName)** In una stringa *s* si definisce *distanza massima tra 'a'* la lunghezza della massima sottostringa di *s* (non vuota) compresa tra due 'a'. Per esempio se *s*='sa**ss**ata' la distanza massima tra 'a' vale 4. Scrivere una funzione che riceve in ingresso il nome di un file di testo **fileName** e restituisce l'insieme di tutte le parole che condividono la maggiore distanza massima (maggiore di 0) tra 'a'. Per esempio, se il file contiene le parole 'sa**ss**ata', 'ca**ss**a' e 'ca**sc**ata', la funzione deve restituire {'sa**ss**ata', 'ca**sc**ata'}.
- **Ex3(file,l)** Su una matrice si definisce cammino zig-zag un cammino che parte da un elemento di una riga (esclusa la prima e l'ultima) in prima colonna, e si muove sempre verso destra nel seguente modo: scende sulla riga sottostante in diagonale, risale in diagonale sulla riga di partenza, sale in diagonale sulla riga soprastante e così via sino a raggiungere l'ultima colonna (non necessariamente sulla riga di partenza). Per esempio, si mostra un cammino zig-zag che parte dalla riga con indice 1, evidenziandone in grassetto gli elementi (si noti che, escludendo la prima e ultima riga nella matrice esiste un solo cammino zig-zag):

1	2	4	5	8
50	7	9	2	7
4	8	2	1	3

Scrivere una funzione che prende in ingresso un nome di un file in formato csv **nomeFile** che contiene i valori di una matrice di interi positivi (senza la riga intestazione), legge il file (si consiglia di usare pandas) e inizializza con questi valori una matrice (si consiglia di usare numpy); restituire un insieme che contiene gli indici delle righe di partenza dei cammini zig-zag che hanno somma massima.
Per esempio, se il file contiene la matrice

1	2	4	5	8
50	7	9	2	7
4	8	2	1	3
1	1	1	1	1

la funzione deve restituire {1}.

Ex4 Il file Ex4.py contiene la funzione `veroFalso()` che stampa 8 domande sullo schermo. La funzione deve essere modificata **cambiando solo 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'
```

Compito d'esame -- 24 gennaio 2024 -- Compito B
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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(1)** In una lista `l` contenente stringhe di lunghezza maggiore di zero si definisce *coppia di gemelle* una coppia di stringhe con indici a distanza `k`, `<l[i], l[i+k]>` che hanno lo stesso numero `k` di occorrenze del primo carattere, dove `k > 1`. La stringa `l[i]` viene indicata come *gemella minore* e quella `l[i+k]` *gemella maggiore*. Scrivere una funzione che riceve in ingresso una lista `l` contenente stringhe non vuote e restituisce il numero di coppie gemelle, escludendo dal conteggio le coppie in cui l'indice della gemella minore coincide con l'indice di una gemella maggiore di un'altra coppia. Per esempio, se `l=['oro','mio','alla','mio','ara','mio']` la funzione deve restituire 1 perché la lista contiene due coppie gemelle `<l[0],l[2]>` e `<l[2],l[4]>` ma la seconda coppia non va contata perché l'indice della gemella minore coincide con quello della gemella maggiore della prima. Se la lista è vuota, la funzione deve restituire -1.

- **Ex2(fileName,coppia)** In una stringa **s** si definisce *numero di coppie di s1* il numero di coppie di **s1** che sono sottostringhe di lunghezza 2 di **s**, si trovano sulla stessa riga e sono separate da caratteri tutti differenti sia dal primo che dal secondo carattere di **s1**. Un singolo elemento **s1** può far parte di una sola coppia. Per esempio se **s='altalena altalenante'** e **s1='al'**, il *numero di coppie di s1* vale 2. Scrivere una funzione che riceve in ingresso il nome di un file di testo **fileName** e una stringa di lunghezza 2 **s1** e restituisce il numero di coppie di **s1** presenti nel file. Per esempio, se **s1='al'** e il file contiene le righe **'altalene alte\n'**, **'altalena altalenante\n'**, la funzione deve restituire 3.
- **Ex3(fileName,l)** Su una matrice si definisce *pattern crociato di una riga* (esclusa la prima e l'ultima) il pattern costituito da tutti gli elementi della riga e dagli elementi della riga sovrastante per la prima colonna, terza colonna, etc., e da quelli della riga sottostante per la seconda colonna, quarta colonna, etc. Per esempio, sulla matrice che segue il pattern crociato della seconda riga è costituito dai seguenti elementi: 5-1-70-8-9-4-2-1-7-8 (si noti che, escludendo la prima e ultima riga, nella matrice esiste un solo il pattern crociato della seconda riga):

```

      1   2   4   5   8
Pattern crociato → 5   70  9   2   7
                  4   8   2   1   3

```

Scrivere una funzione che prende in ingresso un nome di un file in formato csv **fileName** che contiene i valori di una matrice di interi positivi (senza la riga intestazione), legge il file (si consiglia di usare pandas) e inizializza con questi valori una matrice (si consiglia di usare numpy); restituire un insieme che contiene gli indici delle righe dei pattern crociati in cui la somma degli elementi è massima.

Per esempio, se il file contiene la matrice

```

      1   2   4   5   8
      5   70  9   2   7
      4   8   2   1   3
      1   1   1   1   1

```

la funzione deve restituire {1}, in quanto la seconda riga ha un pattern crociato che da somma massima.

Ex4 Il file Ex4.py contiene la funzione `veroFalso()` che stampa 8 domande sullo schermo. La funzione deve essere modificata **cambiando solo 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'
```

Compito d'esame -- 24 gennaio 2024 -- Compito C
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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,l)** Scrivere una funzione che riceve in ingresso una stringa `s` ed una lista `l` di stringhe e calcola quante stringhe di `l` sono anagrammi di `s`. Ad esempio, se la stringa `s` vale `'canto'` e la lista `l` vale `['canto', 'conta', 'cotta', 'tacco']` la funzione deve restituire 2 poiché `'canto'` e `'conta'` sono anagrammi di `'canto'`.
- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo `file` e calcola quante date (corrette e scorrette) sono presenti nel file. Le date sono scritte in questo formato:
 - 1 o 2 cifre
 - Il carattere separatore `'/'` oppure `'-'`
 - 1 o 2 cifre
 - Lo stesso carattere separatore di cui sopra
 - 2 o 4 cifre

Una data è corretta se, oltre a rispettare il formato di sopra corrisponde anche ad un giorno corretto. Ad esempio, 1/1/2001 è corretta, mentre 33/11/1998 non è corretta (assumete che il mese di Febbraio abbia sempre 28 giorni). La funzione deve restituire due numeri interi, cioè il numero di date che rispettano il formato di sopra e sono corrette ed il numero di date che rispettano il formato di sopra ma NON sono corrette. Ad esempio, se il file **file** contiene:

Fin dal 1/1/2001 mi sono attivato, poi il 33/01/2001 ho lavorato, fino al 1-13-19.
Il 12-12/2001 ho deciso che dal 1/1/11 al 31/9/13 mi riposerò

Allora la funzione deve restituire 2,3 poiché 1/1/2001 e 1/1/11 sono date corrette, 33/01/2001, 1-13-19 e 31/9/13 sono date scorrette, mentre 12-12/2001 non rispetta il formato di sopra (usa 2 separatori diversi, non ammesso). NB: per la soluzione si consiglia (ma non è obbligatorio) l'uso delle espressioni regolari.

- **Ex3(file,n)** Un gruppo di amici, i cui nomi compaiono nella prima riga di un file in formato csv **file**, sta giocando ad un gioco tipo gioco dell'oca, dove i giocatori si spostano a turno tirando un dado su un tabellone fatto di **n** caselle (numerate da 1 ad **n**) con le seguenti regole:
 - *Tutti i giocatori partono dalla casella speciale 0 e muovono nell'ordine in cui il loro nome compare nella prima riga del file.*
 - *Se un giocatore, che si trova nella posizione i , tira il dado e fa il numero k (compreso tra 1 e 6) allora si sposta nella posizione $i+k$. Se in quella posizione c'è un altro giocatore allora l'altro giocatore viene messo nella casella 0. Se $i+k$ è maggiore di **n** allora la mossa non si può effettuare ed il giocatore rimane fermo. La casella speciale 0 è la sola dove possono essere presenti contemporaneamente più giocatori.*
 - *Un giocatore vince se arriva per primo nella casella **n**. Se alla fine della partita nessuno arriva nella casella **n** allora non c'è alcun vincitore. Poiché i giocatori si muovono in un ordine fissato, non ci può essere un pari merito.*

Le informazioni sulle mosse sono nelle righe successive alla prima del file. Più precisamente, il file **file** ha il seguente formato:

```
NomeGiocatore_1,...,NomeGiocatore_m  
Tiro_1,...,Tiro_m
```

dove NomeGiocatore_*i* identifica il giocatore *i*, mentre le righe successive alla prima contengono i tiri che hanno fatto i giocatori nei vari turni. Il numero dei giocatori non è fissato e nemmeno il numero di tiri, ma potete assumere che nel file tutti i giocatori fanno sempre lo stesso numero di tiri, cioè tutte le righe contengono lo stesso numero di valori, e che tutti i tiri siano corretti (cioè contengano valori tra 1 e 6). La funzione deve restituire il nome del vincitore (se ce n'è uno) altrimenti deve restituire "Nessuno". Ad esempio, se **n** vale 10 ed il file **file** contiene:

```
Paolo,Miriam,Carla,Giulia,Flavio  
2,1,2,3,5  
5,6,3,4,4  
3,2,2,4,5  
5,6,3,1,2
```

allora la funzione deve restituire "Carla", poiché "Paolo" al primo tiro viene raggiunto da "Carla" e riportato all'inizio nella casella 0.

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'
```

Compito d'esame -- 24 gennaio 2024 -- Compito D
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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,l)** Scrivere una funzione che riceve in ingresso una stringa `s` ed una lista `l` di stringhe e calcola quante stringhe di `l` contengono tutti e soli i caratteri di `s` (anche in numero diverso dalle occorrenze in `s`). Ad esempio, se la stringa `s` vale `'canto'` e la lista `l` vale `['canto', 'conto', 'cotanto', 'taccone']` la funzione deve restituire 2 poiché `'canto'` e `'cotanto'` contengono tutti e soli i caratteri di `'canto'`, mentre `'conto'` non contiene la `'a'` e `'taccone'` contiene anche la `'e'`.
- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo `file` e calcola quante ore (corrette e scorrette) sono presenti nel file. Le ore sono scritte in questo formato:
 - 1 o 2 cifre
 - Il carattere separatore `':'` oppure `','`
 - 1 o 2 cifre

- **Lo stesso carattere separatore di cui sopra**
- **1 o 2 cifre**

Un'ora è corretta se, oltre a rispettare il formato di sopra corrisponde anche ad un'ora corretta. Ad esempio, 1:1:01 è corretta, mentre 25.11.19 non è corretta (assumete che le ore possano andare da 0 a 24 inclusi, mentre minuti e secondi da 0 a 59). La funzione deve restituire due numeri interi, cioè il numero di ore che rispettano il formato di sopra sono corrette ed il numero di ore che rispettano il formato di sopra ma NON sono corrette. Ad esempio, se il file **file** contiene:

Fin dalle 12:00:00 mi sono attivato, poi alle 25.3.5 ho lavorato, fino alle 11:13:19.
Alle 12:12.01 ho deciso che dalle 1:1:11 alle 31:9:13 mi riposerò

Allora la funzione deve restituire (3,2) poiché 12:00:00, 11:13:19 e 1:1:11 sono ore corrette, 25.3.5 e 31:9:13 sono ore scorrette, mentre 12:12.01 non è un'ora (usa 2 separatori diversi, non ammesso). NB: per la soluzione si consiglia (ma non è obbligatorio) l'uso delle espressioni regolari.

- **Ex3(file,n)** Un gruppo di amici sta giocando a un gioco tipo gioco dell'oca, dove i giocatori si spostano a turno tirando un dado su un tabellone fatto di **n** caselle (numerate da 1 ad **n**) con le seguenti regole:
 - *Tutti i giocatori partono dalla casella speciale 0 e muovono nell'ordine in cui sono nel file.*
 - *Se un giocatore, che si trova nella posizione i , tira il dado e fa il numero k (compreso tra 1 e 6) allora si sposta nella posizione $i+k$. Se in quella posizione c'è un altro giocatore allora il giocatore che ha mosso rimane fermo. Se $i+k$ è maggiore di **n** allora la mossa non si può effettuare ed il giocatore rimane fermo. La casella speciale 0 è la sola dove possono essere presenti contemporaneamente più giocatori.*
 - *Un giocatore vince se arriva per primo nella casella **n**. Se alla fine della partita nessuno arriva nella casella **n** allora non c'è alcun vincitore. Poiché i giocatori muovono in un ordine fissato, non ci può essere un pari merito.*

Le informazioni sulle mosse sono nel file di testo in formato csv, **file** che ha il seguente formato:

```
NomeGiocatore_1,...,NomeGiocatore_m  
Tiro_1,...,Tiro_m
```

dove NomeGiocatore_*i* identifica il giocatore *i*, mentre le righe successive contengono i tiri che hanno fatto i giocatori nei vari turni. Il numero dei giocatori non è fissato e nemmeno il numero di tiri, ma potete assumere che nel file tutti i giocatori fanno sempre lo stesso numero di tiri, cioè tutte le righe contengono lo stesso numero di valori, e che tutti i tiri siano corretti (cioè contengano valori tra 1 e 6).

La funzione deve restituire il nome del vincitore (se ce n'è uno) altrimenti deve restituire "Nessuno". Ad esempio, se **n** vale 10 ed il file **file** contiene:

```
Paolo,Miriam,Carla,Giulia,Flavio  
2,1,2,3,5  
5,6,3,4,4  
3,2,2,4,5  
5,6,3,1,2
```

allora la funzione deve restituire "Paolo", poiché "Carla" al primo tiro trova "Paolo" nella posizione 2 e quindi rimane ferma nella casella 0.

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'
```

Compito d'esame -- 22 febbraio 2024 -- Compito A
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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 Python che, ricevendo una lista `l` contenente insiemi di interi, restituisca il numero di volte in cui un insieme `l[i]` è uguale a `l[i+1]` oppure a `l[i+1] ∪ l[i+2]`. Se la lista è vuota, la funzione deve restituire `-1`.

Per esempio, se `l = [{5, 7, 2}, {5, 7}, {5, 2}, {5}, {5}, {2}, {5}, {7}]` la funzione deve restituire `2` perché:

- `l[0] == l[1] ∪ l[2]`
- `l[3] == l[4]`.

- **Ex2(fileName)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **fileName** e conta in quante righe è presente una sequenza di 3 parole consecutive in cui la parola centrale comincia con il carattere che succede immediatamente, in ordine alfabetico, all'ultimo carattere della prima e finisce con un carattere che, sempre in ordine alfabetico, precede immediatamente il primo carattere della terza. Se, per esempio il file è il seguente:

```
Sono patate fresche tre anni
Poco palese
Anche questo
Mangerei pesce fritto palesemente fresco
```

Allora la funzione deve restituire 2 poiché nella prima e nell'ultima riga ci sono 3 parole consecutive con la caratteristica richiesta. N.B. Nel testo, per ordine alfabetico, si intende l'ordine dei codici UNICODE.

- **Ex3(nomeFile)** Su una matrice si definisce 'somma riga partendo dal minimo' la somma degli elementi di una riga a partire dall'elemento che ha il valore minimo (compreso); se la riga ha più elementi con valore minimo, la somma viene fatta partendo dall'elemento che ha indice di colonna più basso. Per esempio, nella matrice sottostante, le somme dei cammini partendo dal minimo delle tre righe sono calcolati sommando i numeri in grassetto e valgono 19, 27 e 4):

```
9    2    4    5    8
2    7    9    2    7
4    8    2    1    3
```

Scrivere una funzione che prende in ingresso il nome di un file in formato csv che contiene i valori di una matrice di interi positivi (senza la riga intestazione), legge il file (si consiglia di usare pandas), inizializza con questi valori una matrice (si consiglia di usare numpy) e restituisce un insieme che contiene gli indici delle righe che hanno il valore massimo della 'somma riga partendo dal minimo'.

Per esempio, se il file contiene la matrice

```
1    2    2    3    2
50   7    9    2    7
20   8    4    3    7
1    1    1    1    1
```

la funzione deve restituire {0,2}.

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

```
return 'BC'
```

Compito d'esame -- 22 febbraio 2024 -- Compito B
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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 Python che, ricevendo una lista `l` contenente degli interi, restituisca il numero di volte in cui un intero `l[i]` è uguale a `l[i+1]` oppure a `l[i+1] + l[i+2]`. Se la lista è vuota, la funzione deve restituire -1.

Per esempio, se `l=[352, 300, 52, 49, 1, 2, 2]` la funzione deve restituire 2 perché:

- `l[0] = l[1] + l[2]` ;
- `l[5] = l[6]` .

- **Ex2(fileName)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **fileName** e conta in quante righe è presente almeno una sequenza di 3 parole consecutive in cui la parola centrale comincia con il carattere che precede immediatamente, in ordine alfabetico, l'ultimo carattere della prima e finisce con un carattere che, sempre in ordine alfabetico, segue immediatamente il primo carattere della terza. Se, per esempio il file è il seguente:

```
Ci sono nove dune da superare
Poco palese
Anche questo
Mangerei tonno novello non cotto
```

Allora la funzione deve restituire 2 poiché nella prima e nell'ultima riga ci sono 3 parole consecutive con la caratteristica richiesta. N.B. Nel testo, per ordine alfabetico, si intende l'ordine dei codici UNICODE.

- **Ex3(nomeFile)** Su una matrice si definisce, per una data colonna, 'somma colonna partendo dal massimo' la somma degli elementi della colonna a partire dall'elemento che ha il valore massimo (compreso); se la colonna ha più elementi con valore massimo, la somma viene fatta partendo dall'elemento che ha indice di riga più basso. Per esempio, nella matrice sottostante, le somme colonne partendo dal massimo sono calcolate sommando i numeri in grassetto e valgono, rispettivamente, 15, 8, 6 e 9:

```
9  2  2  5
2  7  3  3
4  8  3  1
```

Scrivere una funzione che prende in ingresso il nome di un file in formato csv che contiene i valori di una matrice di interi non negativi (senza la riga intestazione), legge il file (si consiglia di usare pandas), inizializza con questi valori una matrice (si consiglia di usare numpy) e restituisce un insieme che contiene gli indici delle colonne che hanno il valore massimo della 'somma colonna partendo dal massimo'. Per esempio, se il file contiene la matrice

```
1    2    10  3    2
10  7    5    2    7
9   8    4    3    7
1   1    1    1    1
```

la funzione deve restituire {0,2}.

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

```
return 'BC'
```

Compito d'esame -- 22 febbraio 2024 -- Compito C
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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,l)** Una stringa si dice “cammellata dalle parole p_1, \dots, p_n ”, se è la concatenazione senza spazi di p_1, \dots, p_n e se tutte le lettere sono minuscole tranne la prima di ogni parola esclusa la prima parola, ovvero devono essere maiuscole tutte e sole le prime lettere di p_i , per $i > 1$. Ad esempio, “massimoValorePossibile” è una stringa “cammellata dalle parole “massimo”, “valore” e “possibile”. Scrivere una funzione che riceve in ingresso una stringa s (che inizia con una lettera minuscola) ed una lista l di parole (tutte minuscole, non vuote e diverse tra loro) e calcola la lista degli indici delle parole di l che “cammellano s ”. Se l non contiene le parole che cammellano l , allora dovrà restituire il valore “Impossibile”. Se s è una stringa vuota dovrà restituire la lista vuota. Ad esempio, se la stringa s vale `'cantoPianoPiano'` e la lista l vale `['canto', 'conta', 'piano', 'tacco']` la funzione deve restituire `[0,2,2]` poiché la stringa cammellata è composta dalla parola in posizione 0, seguita da quella in posizione 2 per due volte.

- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **file** e calcola quante righe del file contengono una doppia che si ripete ESATTAMENTE 3 volte nella riga. Ad esempio, se il file contiene:

pallone bello e giallo
cattedrale brutta, fatta male e stretta

allora la funzione deve restituire 1, perché la prima riga contiene la doppia 'll' esattamente 3 volte, mentre nella riga successiva la doppia 'tt' si ripete 4 volte. NB: per la soluzione è possibile (ma non richiesto) l'uso delle espressioni regolari.

- **Ex3(file,l)** Una ditta mantiene le informazioni sui componenti che ha in magazzino in modo che quando riceve le specifiche dei componenti di nuovi oggetti da produrre, possa calcolare quante copie di ognuno di questi oggetti è possibile costruire. Più precisamente, tali informazioni sono contenute nel file **file**, che ha il seguente formato:

NomeComponente, quantità

dove NomeComponente indica il nome univoco di una componente e quantità è il numero di copie di questa componente presente nel magazzino. La lista **l** contiene dei dizionari, ognuno dei quali specifica, per un dato oggetto, di quali componenti ha bisogno. I dizionari sono composti da coppie della forma NomeComponente:numero. La funzione deve restituire una lista con il numero massimo di copie che possiamo produrre di ogni oggetto. Gli oggetti vanno analizzati nell'ordine in cui sono presenti in **l**, quindi dobbiamo calcolare il massimo numero di copie dell'oggetto in posizione 0, poi il massimo numero dell'oggetto 1 (con le componenti rimaste dopo aver costruito le copie dell'oggetto 0) e così via. Ad esempio, se **l** vale [{ 'gambeCorte':2, 'gambeLunghe':2, 'viti':8, 'seduta':1 }, { 'gambeCorte':4, 'viti':8, 'piano':1 }] ed il file **file** contiene:

NomeComponente, quantità
gambeCorte, 33
gambeLunghe, 12
seduta, 4
piano, 5
viti, 50

allora la funzione deve restituire la lista [4, 2], poiché possiamo costruire 4 copie del primo oggetto e, con le componenti rimaste, 2 copie del secondo.

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'
```

Compito d'esame -- 22 febbraio 2024 -- Compito D
Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

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`.

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**.

È 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.

Per risolvere l'esercizio si possono importare solo le seguenti librerie: numpy, math, pandas e re.

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,l)** Una stringa si dice “tratteggiata dalle parole p_1, \dots, p_n ”, se è la concatenazione, separate dal carattere ‘-’, di p_1, \dots, p_n e se tutte le lettere sono minuscole tranne la prima di ogni parola, ovvero devono essere maiuscole tutte e sole le prime lettere di p_i , per $i \geq 1$. Ad esempio, “Massimo-Valore-Possibile” è una stringa “tratteggiata dalle parole “massimo”, “valore” e “possibile”. Scrivere una funzione che riceve in ingresso una stringa s (che inizia con una lettera minuscola) ed una lista l di parole (tutte minuscole, non vuote e diverse tra loro) e calcola la lista degli indici delle parole di l che “tratteggiano s ”. Se l non contiene le parole che tratteggiano s , allora dovrà restituire il valore “Impossibile”. Se s è una stringa vuota dovrà restituire la lista vuota. Ad esempio, se la stringa s vale ‘Canto-Piano-Piano’ e la lista l vale `['canto', 'conta', 'piano', 'tacco']` la funzione deve restituire `[0,2,2]` poiché la stringa tratteggiata è composta dalla parola in posizione 0, seguita da quella in posizione 2 per due volte.

- **Ex2(file)** Scrivere una funzione che riceve in ingresso il nome di un file di testo **file** e calcola quante righe del file contengono una coppia di vocali consecutive che si ripete ESATTAMENTE 3 volte nella riga. Ad esempio, se il file contiene:

mia pia e giallo
pietà, ovvietà, miele e fiele

allora la funzione deve restituire 1, perché la prima riga contiene la coppia 'ia' esattamente 3 volte, mentre nella riga successiva la coppia 'ie' si ripete 4 volte. NB: per la soluzione è possibile (ma non richiesto) l'uso delle espressioni regolari.

- **Ex3(file,d)** Una ditta mantiene le informazioni sui componenti che ha in magazzino in modo che quando riceve le specifiche dei componenti di nuovi oggetti da produrre, possa calcolare quante copie di ognuno di questi oggetti è possibile costruire ed a che costo. Più precisamente, tali informazioni sono contenute nel file **file**, che ha il seguente formato:

NomeComponente, quantità, prezzo

dove NomeComponente indica il nome univoco di una componente, quantità è il numero di copie di questa componente presenti nel magazzino e prezzo è il costo di un'unità di quella componente. Il dizionario **d** specifica, per un dato oggetto, di quali componenti ha bisogno. E' composto da coppie della forma NomeComponente:numero. La funzione deve restituire una coppia (tupla) con il numero massimo di copie che possiamo produrre dell'oggetto ed il costo di una copia. Ad esempio, se **d** vale {'gambeCorte':2, 'gambeLunghe':2, 'viti':8, 'seduta':1} ed il file **file** contiene:

NomeComponente, quantità, prezzo
gambeCorte, 33, 10
gambeLunghe, 12, 5
seduta, 4, 6
piano, 5, 10
viti, 50, 1

allora la funzione deve restituire la coppia (4, 44), poiché possiamo costruire 4 copie dell'oggetto al costo di 48 (=2*10+2*5+8*1+1*6) ciascuna.

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'
```