

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'
```