

**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'
```