

Compito d'esame -- 29 gennaio 2025 -- Compito D

Nota: gli studenti DSA devono sostenere gli esercizi 1, 2 e 4

Gli studenti che hanno superato la prova intermedia devono sostenere gli esercizi 2 e 3 e le ultime 4 domande dell'esercizio 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 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,n)** Data una lista **l** di parole e un intero positivo o nullo **n**, scrivere una funzione che restituisce la posizione iniziale e finale della più lunga sottosequenza di **l**, lunga almeno due parole, in cui la differenza tra la lunghezza minima e massima delle parole è minore o uguale a **n**. Se ci sono più sottosequenze della stessa lunghezza con queste proprietà, si deve restituire quella più a sinistra. Ad esempio, se **l** vale ['tanto', 'va', 'la', 'gatta', 'al', 'lardo', 'che', 'ci', 'lascia', 'lo', 'zampino'] ed **n** vale 2, entrambe le sottosequenze ['va', 'la'] e ['che', 'ci'] soddisfano i requisiti e ['va', 'la'] è quella più a sinistra. La funzione deve, quindi, restituire i valori 1 e 2. Se la lista non ha almeno 2 parole o la sottosequenza non esiste la funzione deve restituire -1 e -1.
- **Ex2(nomeFile)** Dato un file di testo contenente stringhe contenenti o tutti caratteri alfabetici (parole) o tutti numerici (numeri), spazi ' ', tab '\t' e newline '\n', calcolare un dizionario <intero> <insieme di parole>

che ha come chiavi le lunghezze delle parole che sono in una riga che contiene almeno un'altra parola con la stessa lunghezza. I valori del dizionario sono l'insieme di tutte le parole che sono lunghe come la chiave e che godono della proprietà indicata. Ad esempio, se il file contiene:

```
tanto vanno i gatti al lardo che ci lasciano gli zampini
alcuni superlativi cavalli pascolavano nei prati
```

allora la funzione deve restituire il dizionario

```
5: {'vanno', 'lardo', 'tanto', 'gatti'}, 2: {'ci', 'al'}, 3: {'che', 'gli'}, 11: {'pascolavano', 'superlativi'}
```

Se il file è vuoto o non contiene parole con la proprietà indicata la funzione deve restituire un dizionario vuoto.

- **Ex3(nomeFile, anno)** Sia dato un file csv che contiene le informazioni su eventi a cui alcune persone hanno partecipato, nel formato:

```
nomePersona,evento,anno
```

Scrivere una funzione che prende in ingresso il nome di un file **nomeFile** nel formato di cui sopra e un **anno** e restituisce l'insieme degli eventi a cui hanno partecipato tutte le persone che hanno partecipato ad un evento in quell'anno. Per esempio, se il file contiene:

```
Alice,Concerto,2022
```

```
Alice,Teatro,2022
```

```
Alice,Concerto,2021
```

```
Bob,Sport,2022
```

```
Bob,Concerto,2022
```

```
Bob,Mostra,2022
```

```
Bob,Sport,2020
```

e anno vale 2022, la funzione deve restituire {Concerto} perché tutte le persone che hanno partecipato a uno o più eventi nel 2022 hanno partecipato all'evento Concerto.

- **Ex4** Il file Ex4.py contiene la funzione veroFalso() che stampa 8 domande sullo schermo. La funzione deve essere modificata **cambiando solo il valore di ris nella riga 10**, elencando le lettere delle domande che ritenete essere vere. Ad esempio, se ritenete che le domande B e C sono vere la riga 10 deve essere modificata in

```
ris = 'BC'
```