

Compito d'esame -- 21 gennaio 2019 -- Compito D

Istruzioni (leggere attentamente)

Nota importante: la mancata osservanza delle seguenti regole può comportare la perdita di informazioni necessarie alla valutazione dell'esercitazione.

Registrazione dei dati dello studente: PRIMA DI INIZIARE, eseguite (con un doppio click sull'icona) il programma `RegistraStudente` 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`.

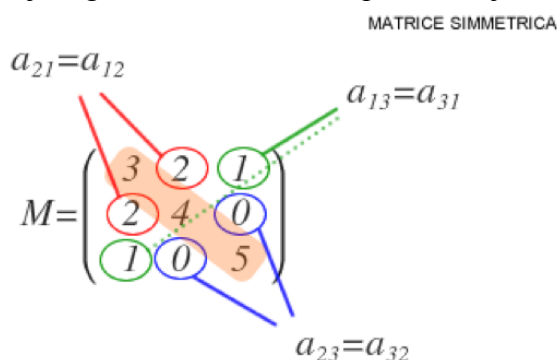
Svolgimento degli esercizi: Per ogni esercizio avete una cartella `EsercN` che contiene un file dal nome `D_ExN.py` (dove **N** è il numero dell'esercizio) con lo scheletro della soluzione. Questo file incorpora un codice di test che proverà la vostra soluzione per un certo numero di possibili dati in input. Aprite il file con Spyder e modificate SOLO il contenuto della funzione. Eseguendo il file `.py` si otterrà il responso dei test sulla console. 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.

Al termine dello svolgimento del compito d'esame **NON è necessario effettuare una procedura di consegna** dell'elaborato. Le vostre soluzioni saranno copiate direttamente dalla cartella **Esame** della macchina su cui avete lavorato. Quindi, per concludere la prova, è sufficiente chiudere il programma Spyder e lasciare la vostra postazione.

Esercizi

- **D_Ex1 (s,c,n)** Scrivere una funzione che riceve in input una stringa NON vuota **s**, un carattere **c** ed un intero non negativo **n** MINORE della lunghezza di **s**, e restituisce la stringa ottenuta da **s** inserendo il carattere **c** ogni **n** posizioni, in modo che **c** sia sempre preceduto e seguito da un carattere (cioè **c** non è inserito in prima o in ultima posizione). Se **n** è 0 la funzione deve restituire la stringa **s**. Ad esempio, se **s**="abcdefghi", **c**="X" e **n**=3, la funzione deve restituire "abcXdefXghi". Se invece **s**="abcdefghi", **c**="X" e **n**=0, la funzione deve restituire "abcdefghi".
- **D_Ex2(m):** Scrivere una funzione che prende in ingresso una matrice QUADRATA di numeri interi **m**, rappresentata come lista di liste, e restituisce la somma di tutti i valori POSITIVI presenti AL DI FUORI DELLA DIAGONALE PRINCIPALE (la diagonale principale è quella evidenziata in arancione in figura) SE **m** è una matrice SIMMETRICA, -1 altrimenti. In una matrice simmetrica ogni elemento in posizione **i,j** è uguale all'elemento in posizione **j,i**, come mostrato in figura



Ad esempio, data la matrice:

```
0  -1  3
-1  5  5
3   5  0
```

La funzione deve restituire 16.

Invece, data la matrice:

```
0  -1  3
-1  5  5
3  -5  0
```

La funzione deve restituire -1.

- **D_Ex3(I1,I2)** Scrivere una funzione che riceve in input le liste **I1** ed **I2**, tali che **I1** ed **I2** abbiano lo stesso numero di elementi, che questi siano esclusivamente stringhe NON vuote e che le stringhe nella stessa posizione in **I1** ed **I2** abbiano la stessa lunghezza. La funzione deve restituire la lista, ORDINATA alfabeticamente, che contiene tutte e sole le stringhe di **I1** aventi la seguente proprietà: ciascun carattere della stringa è uguale al carattere nella posizione successiva della corrispondente stringa in **I2**, assumendo che l'ultima posizione abbia come successiva la prima posizione (bisogna inoltre distinguere fra lettere minuscole e maiuscole). Ad esempio, se **I1** = ["mamma", "asso", "re"], ed **I2** = ["amamm", "lago", "er"], la funzione deve restituire ["mamma", "re"].
- **D_Ex4(file, ore, minuti)** Si consideri un **file** di testo in formato csv contenente gli orari di arrivo di treni in una giornata e gli eventuali ritardi e/o anticipi, nella forma:

CodiceTreno,OraArrivo,MinutiArrivo

oppure

CodiceTreno,+MinutiRitardo

oppure

CodiceTreno,-MinutiAnticipo

dove **CodiceTreno** è il codice identificativo di un treno, **OraArrivo** è l'ora prevista di arrivo (numero intero tra 0 e 23), **MinutiArrivo** è il minuto previsto di arrivo (numero intero tra 0 e 59), **+MinutiRitardo** e **-MinutiAnticipo** sono rispettivamente i minuti di ritardo accumulati o i minuti di anticipo guadagnati, sempre preceduti dal rispettivo segno + o -. Si può assumere che il file csv contenga sempre un input valido, che vi sia una sola riga contenente l'orario di arrivo previsto di un treno, che questa compaia sempre prima di eventuali righe di anticipo o ritardo e che tutti i treni arrivino entro le 23 e 59.

Scrivere una funzione Python che preso in ingresso un **file** csv nel precedente formato, un'ora e **minuti**, restituisca la lista ORDINATA E SENZA RIPETIZIONI dei **CodiceTreno** (**in formato stringa**) relativi ai treni che arrivano a destinazione entro **ora** e **minuti**.

Ad esempio, dato il file contenente

```
54200,9,23
79001,12,30
6310,13,00
54200,+180
6310,-60
54200,-60
```

ora pari a 12 e minuti pari a 20, la funzione dovrà restituire la lista ["54200", "6310"]