

**Corso di Laurea Triennale in Ingegneria Gestionale**  
**Fondamenti di Informatica**  
**Esercitazione 11 del 5/5/2017**  
**Canale A-L**

### **Istruzioni**

Per ogni esercizio avete un file dal nome Esercit11ProgX.py (dove X è il numero dell'esercizio) con lo schema della soluzione, un file ExX.pyc ed uno ProvaExX.py.

1. Scaricate il file Esercitazione11.zip
2. Estraete tutti i file in una cartella Esercitazione11
3. Partite dai files Esercit11ProgX.py scrivendo lì la vostra soluzione all'esercizio.
4. Per testare un esercizio **DOVETE** usare i programmi ProvaExX.py

I files ExX.pyc non sono leggibili, ma contengono la nostra soluzione al problema e sono usati da ProvaExX.py per verificare se la vostra soluzione è corretta.

**NOTA:** I file ExX.pyc contenuti nel file Esercitazione10.zip funzionano con la versione 3.5 di Python (versione utilizzata in laboratorio). Se sul pc su cui risolvete gli esercizi avete installato una versione diversa di Python non riuscirete a eseguire i test. Si consiglia di installare sempre la versione 3.5.

### **Esercizio 1**

Completare la funzione ricorsiva sequenzaCarInStringhe(s1, s2) che prende in ingresso due stringhe e restituisce una stringa che contiene tutti i caratteri (senza ripetizioni) che appaiono in tutte e due le stringhe.

Esempio: se s1 = 'palla' e s2 = 'pesce cane' allora deve stampare la stringa 'pa'

### **Esercizio 2**

Completare la funzione ricorsiva massimoRiga(m) che prende come input una matrice m e restituisce un lista l il cui l'elemento i-esimo è il massimo della riga i di m.

Esempio: se  $m = \begin{pmatrix} 1 & -4 & 10 \\ 9 & 15 & 7 \\ -10 & -1 & -3 \end{pmatrix}$  la funzione dovrà restituire la lista  $l=[10, 15, -1]$

### **Esercizio 3**

Completare la funzione ricorsiva minimoColonna(m) che prende come input una matrice m e restituisce un lista l il cui l'elemento i-esimo è il minimo della colonna i di m.

Esempio: se  $m = \begin{pmatrix} 1 & -4 & 10 \\ 9 & 15 & 7 \\ -10 & -1 & -3 \end{pmatrix}$  la funzione dovrà restituire la lista  $l=[-10, -4, -3]$

### **Esercizio 4**

Completare la funzione ricorsiva `listaStringheDueCar(l, c)` che riceve in ingresso una lista di stringhe `l` ed un carattere `c` e restituisce una lista composta dalle sole stringhe di `l` in cui `c` compare esattamente 2 volte.

Esempio: se `l = ['casa', 'palla', 'pallone', 'casata', 'villa']` e `c = 'a'` allora deve restituire la lista `['casa', 'palla']`.

## Esercizio 5

Il numero 6174 è conosciuto come la costante di Kaprekar in onore del matematico indiano Dattatreya Ramachandra Kaprekar che la scoprì. Tale numero si ottiene nel modo seguente:

1. Prendere qualsiasi numero di quattro cifre, usandone almeno due differenti. (Si possono anche inserire degli zeri anche all'inizio.)
2. Posizionare le cifre in ordine decrescente e poi in ordine crescente così da ottenere due numeri di quattro cifre  $c_1$  e  $c_2$ , aggiungendo degli zero iniziali se necessario.
3. Calcolare il numero  $c_1 - c_2$  sottraendo il numero più piccolo da quello più grande.
4. Ripetere il processo partendo dal punto 2.

Il processo sopra descritto, conosciuto come l'operazione di Kaprekar, convergerà sempre al valore 6174 (il suo punto fisso), in al più 7 iterazioni. Una volta raggiunto il valore 6174, il processo continua a dare il valore  $7641 - 1467 = 6174$ . Per esempio, se consideriamo il numero 3524, occorrono tre applicazioni dell'operazione di Kaprekar per giungere alla costante di Kaprekar:

1. **5432 - 2345 = 3087**
2. **8730 - 0378 = 8352**
3. **8532 - 2358 = 6174**

Completare la funzione ricorsiva `kaprekar(s)` che prende in input una *stringa* che rappresenta un numero  $n$  a 4 cifre e restituisce il numero di applicazioni dell'operazione di Kaprekar necessarie per raggiungere la costante di Kaprekar a partire da  $n$ .