

Fondamenti di Informatica
Corso di Laurea in Ingegneria Gestionale
A.A. 2017-18:
Esercitazione #5

Federico Scafoglieri
scafoglieri@diag.uniroma1.it

Domenico Lembo
lembo@diag.uniroma1.it

Obiettivo dell'esercitazione: Comprendere funzioni e liste.

Come testare il codice

Modalità tramite Test

Scaricare il file "File_Utili_Esercitazione_05", scompattarla, all'interno troverete una cartella per esercizio contenente:

- Un file *TestEsN.py* che sarà il modulo da avviare per testare la propria soluzione
- Un file *EsN.py* da completare implementando la funzione richiesta dall'esercizio
- Un file *SolN.py* contenente la soluzione ed utile per la funzione di test.

Per portare a termine l'esercitazione aprite i file *EsN.py* e implementate la funzione seguendo la segnatura proposta nel testo dell'esercizio *N* di riferimento

Modalità tramite Shell

Scrivere la funzione richiesta, Run → Run Module e nella shell inserire i parametri.

Per esempio data la seguente funzione:

```
1 def isOdd(number):  
2     if number%2==0:  
3         return False  
4     else:  
5         return True
```

Run → Run Module

```
1 >>> isOdd(1)  
2 True  
3 >>> isOdd(2)  
4 False
```

Suggerimenti

Fare ATTENZIONE al tipo restituito dalla funzione. Se la funzione richiede di restituire un booleano *True/False*, restituire la stringa 'True'/'False' è un errore. Se si richiede di restituire la stringa 'error', restituire la stringa 'Error' è un errore.

Esercitazione Laboratorio

Esercizio 1

Difficoltà: ★

Scrivere una funzione Python *restutiscitupla*(*l*, *i*) che prende come parametri una lista *l* e un intero *i* e restituisce una **tupla** composta da tutti gli elementi dall'indice *i* fino alla fine della lista *l*.

Esempio:

- Se $l = [a, f, r, d, g, e, q]$ e $i = 4$ la funzione restituisce (g, e, q)

Testare la propria soluzione nella shell verificando di aver restituito una **tupla**.

Nota: Una tupla è una lista non modificabile

Esercizio 2

Difficoltà: ★★

Scrivere una funzione Python *validamatrice*(*l*) che prende come parametri una lista di liste di interi *l* e restituisce il **booleano** *True* se *l* è una matrice altrimenti restituisce *False*

Esempio:

- Se $l = [[1, 2], [3, 4]]$, la funzione restituisce *True*
- Se $l = [[1, 2], [3, 4, 5]]$ la funzione restituisce *False*

Nota: Una matrice è rappresentata da una lista di liste in cui le sottoliste hanno tutte lo stesso numero di elementi.

Esercizio 3

Difficoltà: ★★

Scrivere una funzione Python *scambiarighe*(*m*, *i1*, *i2*) che prende come parametri una matrice *m* e due interi *i1* ed *i2* che rappresentano gli indici di due righe e restituisce una nuova **matrice** in cui le righe *i1* e *i2* sono state scambiate tra loro. Se gli indici non sono validi o se *m* non è una matrice, la funzione restituisce una **copia** della lista di liste iniziale.

Esempio:

- Se $m = \begin{bmatrix} 1 & 5 \\ 7 & 12 \\ -1 & 3 \\ 5 & -1 \end{bmatrix}$ e $i1 = 1$ e $i2 = 3$, restituisce: $m = \begin{bmatrix} 1 & 5 \\ 5 & -1 \\ -1 & 3 \\ 7 & 12 \end{bmatrix}$

Nota: Per controllare se *m* è una matrice, riutilizzare la soluzione dell'esercizio 2.

Esercizio 4

Difficoltà: ★★

Scrivere una funzione Python *restituiscicolonna*(*m*, *n*) che prende come parametri una matrice *m* e un intero *i* che rappresenta l'indice di una riga e restituisce una **lista** contenente gli elementi della colonna *n* –esima di *m*

Esempio:

- Se $m = \begin{bmatrix} 1 & 3 & 1 & 5 \\ 0 & 5 & 7 & 12 \\ 2 & 5 & -1 & 3 \\ 3 & 5 & 5 & -1 \end{bmatrix}$ e $n = 1$ la funzione restituirà $[1, 0, 2, 3,]$

Esercizio 5

Difficoltà: ★★

Scrivere una funzione Python *validatriangolareinferiore(m)* che prende come parametro una matrice m e restituisce *True* se m è triangolare inferiore, *False* altrimenti.

Esempio:

- Se m rappresenta la matrice $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 7 & 1 & 0 & 0 \\ -1 & 8 & 3 & 0 \\ 5 & 6 & -1 & 4 \end{bmatrix}$ la funzione restituisce: *True*
- Se m rappresenta la matrice $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 2 & 5 & 0 & 1 \\ 1 & -8 & 3 & 0 \\ 9 & 16 & -12 & 3 \end{bmatrix}$ la funzione restituisce: *False*

Esercizio 6

Difficoltà: ★★

Scrivere una funzione Python *restituiscistringa(m, i)* che prende come parametro una matrice m e un intero i (che indica il numero di riga), restituisce una stringa s composta concatenando tutti gli elementi delle righe multiple di i (i incluso) separando le varie concatenazioni delle righe con uno spazio.

Esempio:

- Se m rappresenta la matrice $\begin{bmatrix} f & e & c & r \\ c & i & a & o \\ 5 & 6 & -1 & 4 \\ p & i & n & o \\ u & id & a & 0 \end{bmatrix}$ e $i = 2$ restituisce: *ciao pino*
- Se m rappresenta la matrice $\begin{bmatrix} c & i & a & o \\ p & g & h & T \\ c & a & s & a \\ r & e & m & o \end{bmatrix}$ e $i = 7$ restituisce: (stringa vuota)

Esercizio 7

Difficoltà: ★★★

Scrivere una funzione Python *verificalista(l, i)* che prende come parametro una lista di liste l contenente interi e un intero i e restituisce *True* se in ogni sottomatrica sl con indice pari, la somma degli elementi di sl è pari a i , *False* altrimenti.

Esempio:

- Se $l = [[1, 0, 8, 0, 0, 2], [7, 1, 0, 0], [-1, 8, 2, 0, 1, 1], [5, 6, -1, 4, 9]]$ e $i = 11$ la funzione restituisce: *True* poichè la somma degli elementi alla riga 0 è uguale a 11, e la somma degli elementi della sottolista con indice 2 è uguale a 11.
- Se $l = [[1, 0, 0, 5], [2, 5, 0, 1, 10], [1, -8, 3, 0], [9, -12, 3]]$ e $i = 7$ la funzione restituisce: *False* poichè non soddisfa la condizione.

Esercizio 8

Difficoltà: ★★

Scrivere una funzione Python *listamaggiori(l)* che prende come parametro una lista di liste l e restituisce una lista composta da tutti gli elementi più grandi di ogni sottolista.

Esempio:

- Se $l = [[1, 0, 8, 0, 0, 2], [7, 1, 0, 0], [-1, 8, 2, 0, 1, 1], [5, 6, -1, 4, 9]]$ la funzione restituisce: $[8, 7, 8, 9]$
- Se $l = [[-1, 0, 0, -5], [2, 5, 0, 1, 10], [1, -8, 3, 0], [9, -12, 3]]$ la funzione restituisce: $[-1, 10, 3, 9]$

HomeWork

Esercizio 9

Homework

Difficoltà: ★★★

Scrivere una funzione Python *scambiacolonne*(*m*, *i1*, *i2*) che prende come parametri una matrice *m* e due interi *i1* ed *i2* che rappresentano gli indici di due colonne e restituisce una nuova **matrice** in cui le colonne *i1* e *i2* sono state scambiate tra loro. Se gli indici non sono validi, la funzione restituisce una **copia** della matrice iniziale.

Esempio:

- Se *m* rappresenta la matrice
$$\begin{bmatrix} 1 & 9 & 5 & 8 \\ 7 & 1 & 12 & 10 \\ -1 & 8 & 3 & 11 \\ 5 & 6 & -1 & 4 \end{bmatrix}$$
 e *i1* = 0 e *i2* = 2,
la funzione restituisce:
$$\begin{bmatrix} 5 & 9 & 1 & 8 \\ 12 & 1 & 7 & 10 \\ 3 & 8 & -1 & 11 \\ -1 & 6 & 5 & 4 \end{bmatrix}$$