# Laboratorio 8

Informatica@DSS 2019/2020 — (E-N)

Massimo Lauria <massimo.lauria@uniroma1.it>*

Lunedì, 25 Novembre 2019

Lo scopo del laboratorio è di esercitarsi e misurare la propria preparazione: gli esercizi non sono troppo difficili, se si sono seguite le lezioni. Non vi viene comunque messo alcun voto.

**Modalità di lavoro**: gli studenti devono lavorare in autonomia o in piccoli gruppi, senza disturbare i colleghi. Il lavoro di gruppo è fruttuoso solo se tutti partecipano e se ognuno scrive una propria soluzione per tutti gli esercizi.

Il docente cercherà per quanto possibile di non occupare il tempo del laboratorio per introdurre materiale nuovo, anche se a volte questo sarà necessario. Il docente è a disposizione per aiutare gli studenti, che possono iniziare a lavorare anche prima che il docente arrivi in aula, se lo desiderano

**Raccomandazioni**: leggete bene il testo degli esercizi prima di chiedere chiarimenti. In ogni caso sarò in aula con voi. Alla fine della lezione per favore rispondete al questionario, disponibile al link alla fine di questo documento.

**Uso dei file di test**: per aiutarvi a completare questa esercitazione avete a disposizione dei programmi di test per testare la vostra soluzione. Questi sono simili a quelli che avrete in sede di esame, pertanto vi consiglio di imparate ad usarli. Per portare a termine l'esercizio è necessario

- scrivere un file di soluzione **col nome specificato**;
- avere dentro la funzione **col nome specificato**;
- porre il file di test corrispondente **nella stessa cartella**;

*http://massimolauria.net/courses/informatica2019/

- eseguire in quella cartella il comando

```
$ python3 <fileditest>
```

dove `<fileditest>` va ovviamente sostituito con il nome del file di test appropriato per l'esercizio su cui state lavorando.

Per ogni esercizio ci sta un file di test indipendente, così da poter lavorare sugli esercizi uno alla volta con più agio. Controllate i parametri passati in input ed eventualmente sollevate le eccezioni `ValueError` o `TypeError`.

Il risultato di ogni test è una schermata (o più schermate) nella quale si mostra:

- se è stato trovato il file con il nome corretto,
- se il file contiene la funzione con il nome corretto,
- se chiamate alla funzione con diversi valori dei parametri terminano restituendo risultati corretti.

Per ogni funzione scritta vengono eseguite chiamate con diversi valori dei parametri. L'esito dei test viene riportato con il carattere

- `E` se la chiamata non può essere eseguita,
- `F` se la funzione non restituisce il risultato corretto,
- `.` se la funzione restituisce il risultato corretto.

Questi esercizi riguardano la rappresentazione di matrici di numeri in Python attraverso liste di liste di numeri. Ad esempio una matrice

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$$

viene rappresentata in Python come la lista

```
[[1,2,3], [4,5,6], [7,8,9], [10,11,12]]
```

# 1 Stampare una matrice

Scrivere una funzione `stampa_mat(m)` che:

- riceve come parametro una lista di liste `m`;
- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- stampa la lista di liste `m`, sotto forma di tabella, andando a capo alla fine di ciascuna riga. Ad esempio, se viene chiamata con parametro

```
[ [2, 4, 3, 6, 8], [3, 1, 4, 0, 0], [1, 5, 1, 7, 4] ]
```

  deve essere visualizzato

```
2  4  3  6  8
3  1  4  0  0
1  5  1  7  4
```

- la funzione non deve restituire nulla.

Per questo esercizio non viene fornito un file di test. Provare a chiamare la funzione con argomenti diversi e verificare l'aspetto della stampa.

# 2 Somma degli elementi di una matrice

Scrivere una funzione `somma_mat(m)` che:

- riceve come parametro una lista di liste `m`;
- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- solleva una eccezione `TypeError` se gli elementi della matrice non sono numeri;
- restituisce la somma di tutti gli elementi della matrice.

File della soluzione: `lab08somma.py`

File di test: `test_lab08somma.py`

# 3 Minimo degli elementi di una matrice

Scrivere una funzione `min_mat(m)` che:

- riceve come parametro una lista di liste `m`;

- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- solleva una eccezione `ValueError` se la lista è vuota e se la prima riga della matrice non contiene almeno un elemento;
- solleva una eccezione `TypeError` se gli elementi della matrice non sono numeri;
- restituisce il minimo elemento della matrice.

File della soluzione: `lab08min.py`

File di test: `test_lab08min.py`

# 4 Posizione del massimo elemento di una matrice

Scrivere una funzione `pos_max_mat(m)` che:

- riceve come parametro una lista di liste `m`;
- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- solleva una eccezione `ValueError` se la lista è vuota e se la prima riga della matrice non contiene almeno un elemento;
- solleva una eccezione `TypeError` se gli elementi della matrice non sono numeri;
- restituisce una coppia di interi (ad esempio con un `return i, j`) che sono rispettivamente l'indice di riga e di colonna in cui compare il massimo elemento della matrice. In caso di più elementi pari al massimo deve essere restituita la posizione del primo di essi (minima riga e, in caso di più massimi sulla riga, minima colonna).

File della soluzione: `lab08posmax.py`

File di test: `test_lab08posmax.py`

# 5 Somma degli elementi della diagonale principale di una matrice

Scrivere una funzione `somma_diagonale_mat(m)` che:

- riceve come parametro una lista di liste `m`;
- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- solleva una eccezione `TypeError` se gli elementi della matrice non sono numeri;
- solleva una eccezione `ValueError` se la matrice non è quadrata (ciascuna riga ha la stessa lunghezza, che è anche il numero di righe);
- restituisce la somma degli elementi sulla diagonale principale. Notare che per una matrice con `n` righe ed `n` colonne si devono sommare solo `n` elementi.

File della soluzione: `lab08diagonale.py`

File di test: `test_lab08diagonale.py`

# 6 Somme per riga di una matrice

Scrivere una funzione `somme_per_riga_mat(m)` che:

- riceve come parametro una lista di liste `m`;
- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- solleva una eccezione `TypeError` se gli elementi della matrice non sono numeri;
- restituisce una lista in cui l'elemento di posto `i` è la somma degli elementi della riga `i`-esima. La lunghezza di tale lista è uguale al numero di righe della matrice. Se viene passata una matrice vuota (quindi `[]`) deve essere restituita una lista vuota (quindi `[]`).

File della soluzione: `lab08righe.py`

File di test: `test_lab08righe.py`

# 7 Somme per colonna di una matrice

Scrivere una funzione `somme_per_colonna_mat(m)` che:

- riceve come parametro una lista di liste m;
- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- solleva una eccezione `ValueError` se le righe della matrice non sono tutte della stessa lunghezza;
- solleva una eccezione `TypeError` se gli elementi della matrice non sono numeri;
- restituisce una lista in cui l'elemento di posto `i` è la somma degli elementi della colonna `i`-esima. La lunghezza di tale lista è uguale al numero di colonne della matrice. Se viene passata una matrice vuota (quindi `[]`) deve essere restituita una lista vuota (quindi `[]`).

File della soluzione: `lab08colonne.py`

File di test: `test_lab08colonne.py`

# 8 Posizione della riga con somma massima

Scrivere una funzione `pos_riga_massima_mat(m)` che:

- riceve come parametro una lista di liste `m`;
- solleva una eccezione `TypeError` se il parametro ricevuto non è una lista, o se gli elementi della lista non sono liste;
- solleva una eccezione `TypeError` se gli elementi della matrice non sono numeri;
- restituisce la posizione della riga che ha massima somma. Ad esempio, se viene chiamata con parametro

  ```
  [ [2, 4, 3, 6, 8], [3, 1, 4, 0, 0], [9, 5, 9, 7, 4] ]
  ```

  deve restituire il valore 2, poiché la riga in posizione 2 ha somma 34, mentre le righe in posizione 0 e 1 hanno rispettivamente somma 23 e 8. Se la matrice ha 0 righe (matrice vuota []) deve essere restituito il valore -1.

File della soluzione: `lab08maxriga.py`

File di test: `test_lab08maxriga.py`

---

Questionario: rispondete **dopo** aver fatto gli esercizi.

# http://bit.ly/INFO2019-LAB08