

matrici

liste multidimensionali

- o nelle liste bidimensionali (*matrici*), i dati sono organizzati per *righe* e per *colonne*, come in una *tabella*

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$$

```
a = [['A', 'B', 'C', 'D'],
      ['E', 'F', 'G', 'H'],
      ['I', 'L', 'M', 'N']] # 2D
```

- ogni riga della matrice viene rappresentata da una lista
- la matrice è una lista i cui elementi sono le liste che rappresentano le righe
- per individuare un elemento si deve specificare la lista e la posizione dell'elemento all'interno della lista

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

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

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

matrix[0][0] is 1
matrix[4][4] is 3
```

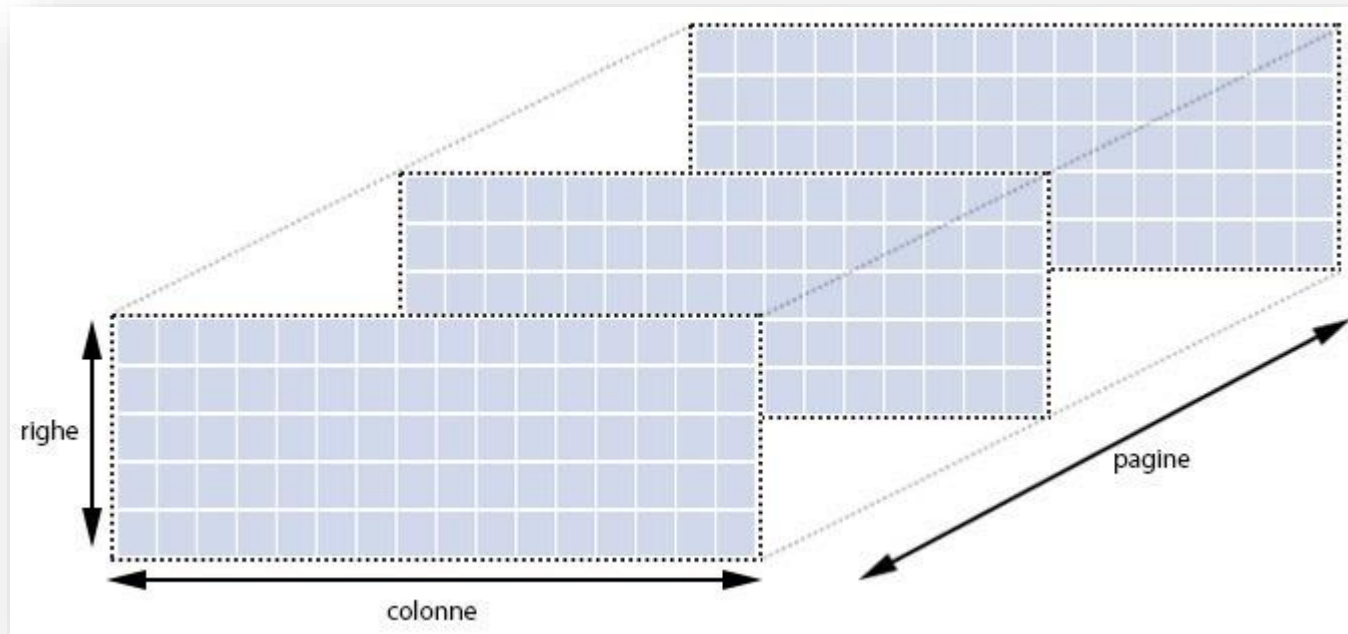
```
matrice = [[2, 4, 3, 8],
           [9, 3, 2, 7],
           [5, 6, 9, 1]]
righe = len(matrice)
colonne = len(matrice[0])

for r in range(righe):
    totale = 0
    for c in range(colonne):
        val = matrice[r][c]
        totale += val
    print("riga #", r, "totale: ", totale)
```

- *problema:*
 - *memorizzare i risultati ottenuti da n studenti in m prove, con n uguale a 4 e m uguale a 3*
 - *calcolare la **somma dei voti** che ogni studente ha ottenuto nelle differenti prove e la **media** dello studente*
 - *calcolare la **somma dei voti degli studenti per ogni prova** e quella relativa per ogni prova*
- *input: voti ottenuti dagli studenti per ogni prova*
- *output: visualizzare la tabella dei risultati*
 - *per ogni studente la somma dei voti delle differenti prove e la relativa media*
 - *per ogni prova la somma dei voti degli studenti e la relativa media*

```
colonne = 3          #dato noto
righe = 4           #dato noto
#inizializzazione di tutti gli elementi a ' '
matrice = [[' ' for c in range(colonne)] for r in range(righe)]
```

```
#metodo alternativo
matrice = []
for r in range(righe):
    nuova_riga = []
    for c in range(colonne):
        nuova_riga.append(' ')
    matrice.append(nuova_riga)
```



MATRICI
esercizi



- caricare una matrice di 4 righe e 5 colonne con valori casuali compresi fra 1 e 100
- visualizzare la matrice formattandola opportunamente
- ricercare il valore massimo presente nella matrice e visualizzare le coordinate (riga e colonna) di tale valore

- *incolonnamento dati*
- visualizzare due *tabelle* con i caratteri *ASCII*
 - 4 righe x 24 colonne, codici da 32 a 126
- tabella 1: mostrare in ordine i caratteri, colonna per *colonna*
- tabella 2: mostrare in ordine i caratteri, riga per *riga*

```
$ ( , 048 < @DHLPTX \ ` dhlptx |
! % ) - 159 = AEIMQUY ] aeimquy }
" & * . 26 : > BFJNRVZ ^ bfjnrvz ~
# ' + / 37 ; ? CGKOSW [ _ cgkosw {
```

```
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7
8 9 : ; < = > ? @ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g
h i j k l m n o p q r s t u v w x y z { | } ~
```

- la distanza fra città viene normalmente rappresentata mediante tabelle (vedi figura)
- analizzare le opportune strutture dati
- scrivere un'applicazione che richiede il nome di due città e visualizza la distanza fra queste

Distance Table (in miles)

	Chicago	Boston	New York	Atlanta	Miami	Dallas	Houston
Chicago	0	983	787	714	1,375	967	1,087
Boston	983	0	214	1,102	1,505	1,723	1,842
New York	787	214	0	888	1,549	1,548	1,627
Atlanta	714	1,102	888	0	661	781	810
Miami	1,375	1,505	1,549	661	0	1,426	1,187
Dallas	967	1,723	1,548	781	1,426	0	239
Houston	1,087	1,842	1,627	810	1,187	239	0