



UNIVERSITÀ  
DI TRENTO

Dipartimento di Ingegneria e  
Scienza dell'Informazione  
DISI - Trento

# Programmazione 1

## 14 - Esercitazione

Giovanni De Toni

[giovanni.detoni@unitn.it](mailto:giovanni.detoni@unitn.it)

### Attenzione

La presente esercitazione verrà trasmessa via Zoom. Essa verrà anche registrata e successivamente messa a disposizione degli studenti dell'Università degli Studi di Trento. Per gli utenti connessi attraverso Zoom, in caso non desideriate per qualunque motivo essere registrati, siete pregati di effettuare la disconnessione ora. La lezione sarà comunque visionabile in modo asincrono.

Anno Accademico 2021/2022

# 00 - Array Multidimensionali

## Definizione

Array: Sequenza finita di elementi consecutivi dello stesso tipo



questi elementi possono essere a loro volta altri array (array multidimensionali)

- sintassi :
- `id[exp1][exp2]...`;
  - `id[exp1][exp2]... = {list};`
  - `id[][exp2]... = {list};`

# 00 - Array Multidimensionali

## Matrici

Array bidimensionali sono chiamati *matrici*

```
int matrix [3][3];
```

m[0,0]	m[0,1]	m[0,2]
m[1,0]	m[1,1]	m[1,2]
m[2,0]	m[2,1]	m[2,2]

m[0,0]	m[0,1]	m[0,2]	m[1,0]	m[1,1]	m[1,2]	m[2,0]	m[2,1]	m[2,2]
--------	--------	--------	--------	--------	--------	--------	--------	--------

# 00 - Array Multidimensionali

## Passaggio a funzioni

```
int function(const int matrix[][dim]);
```

```
int function(const int matrix[][dim][dim2]);
```

```
int function(const int matrix[][dim][dim2][dim3]);
```

...

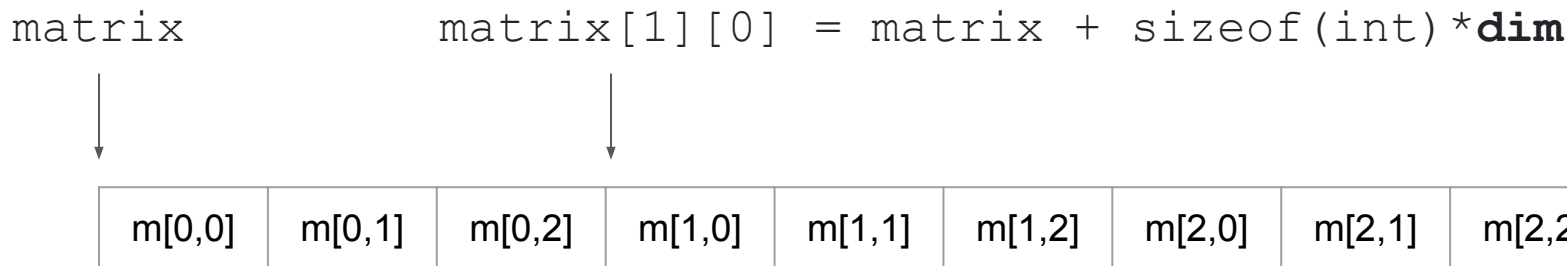
perché bisogna specificare `dim` ?

# 00 - Array Multidimensionali

## Passaggio a funzioni

```
int function(const int matrix[][dim]);
```

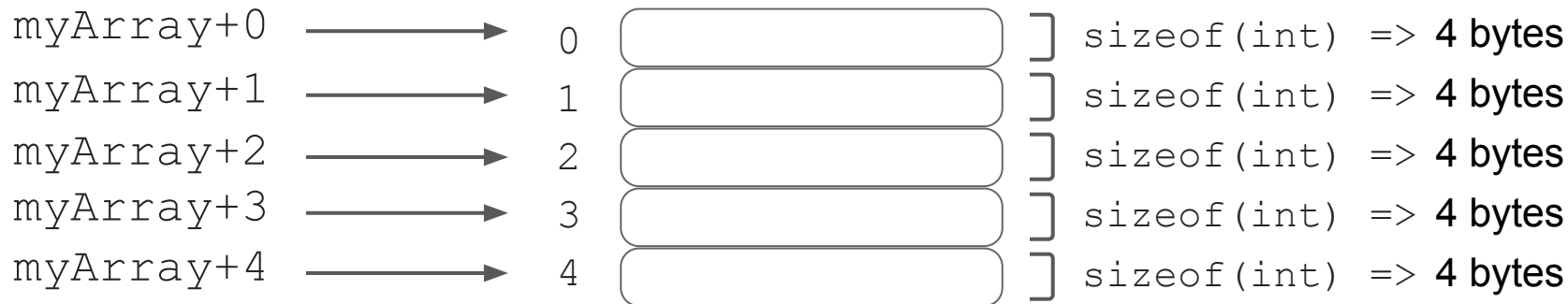
Perché `matrix` è un indirizzo di memoria, ma la funzione deve sapere dove accedere quando chiedo `matrix[1][0]`



# 00 - Array Multidimensionali

## Aritmetica dei puntatori

```
int myArray [5];
```



Sommando un valore  $x$  ad un indirizzo di memoria (e.g., puntatore  $p$ ) di tipo  $T$ , ottengo  $p + \text{sizeof}(T) * x$

# 00 - Array Multidimensionali

## Puntatori e array

```
int myarray [5];  
int* puntatore = myarray;
```

```
* (puntatore+1) == myarray[1]
```

```
* (puntatore+2) == myarray[2]
```

```
* (puntatore+3) == myarray[3]
```

```
* (puntatore+4) == myarray[4]
```

...

```
* (puntatore+i) == myarray[i]
```

# 01 - Stampa Puntatore

**Scrivere un programma che dichiari e poi inizializzi (con valori random) un array di 10 interi e poi stampi a video tutti gli elementi dell'array tramite una funzione, passando l'array come puntatore.**

**(i.e., potete usare i caratteri ' [ ' e ' ] ' solo quando dichiarate l'array)**



## 02 - Stampa Matrice

**Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di interi e poi stampi a video tutti gli elementi della matrice (ricordarsi di gestire anche le matrici non quadrate).**

## 03 - Matrice Trasposta

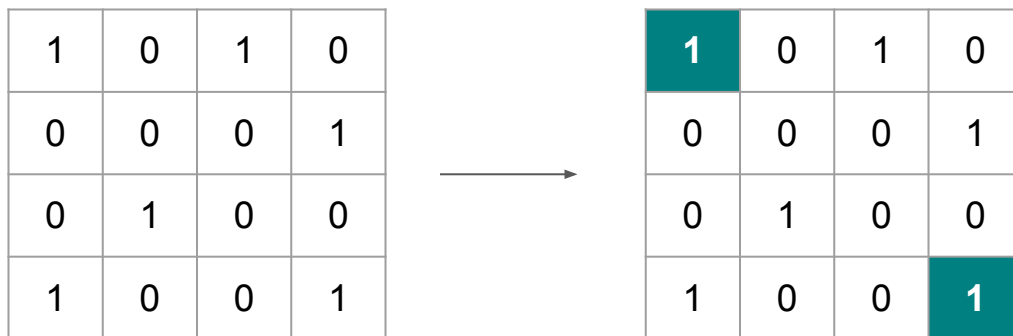
**Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di interi, la stampi a video, calcoli la matrice trasposta e la stampi a video (ricordarsi di gestire anche le matrici non quadrate).**

$$\begin{pmatrix} 1 & 2 & 8 \\ 3 & 4 & 3 \\ 5 & 6 & 1 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 8 & 3 & 1 \end{pmatrix}$$

## 04 - Isole

**Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di 10x10 interi con valore 0 oppure 1.**

**Scrivere poi una funzione che torni il numero di “isole”, definite come elementi della matrice con valore 1 circondati da soli zeri.**



## 05 - Sarrus

**Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di 3x3 interi. Scrivere poi una funzione che calcoli il prodotto dei valori di una determinata diagonale.**

**Usare infine questa funzione per calcolare il determinante della matrice tramite la regola di Sarrus.**

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - gec - hfa - idb$$

## 05 - Sarrus

**Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di 3x3 interi. Scrivere poi una funzione che calcoli il prodotto dei valori di una determinata diagonale.**

**Usare infine questa funzione per calcolare il determinante della matrice tramite la regola di Sarrus.**

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \boxed{aei} + \boxed{bfg} + \boxed{cdh} - gec - hfa - idb$$

## 05 - Sarrus

Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di 3x3 interi. Scrivere poi una funzione che calcoli il prodotto dei valori di una determinata diagonale.

Usare infine questa funzione per calcolare il determinante della matrice tramite la regola di Sarrus.

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - \boxed{gec} - \boxed{hfa} - \boxed{idb}$$