



**POLITECNICO**  
MILANO 1863

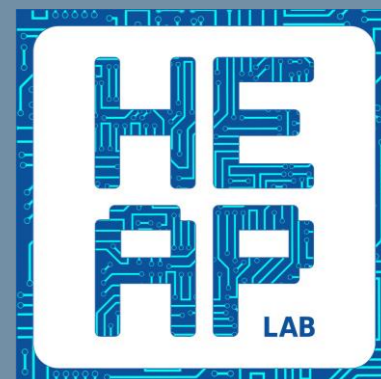
# INFORMATICA A

A.A. 2017-18

Laboratorio n°3

Ing. Gian Enrico Conti

Dott. Michele Zanella



# Info Logistiche (Last time)

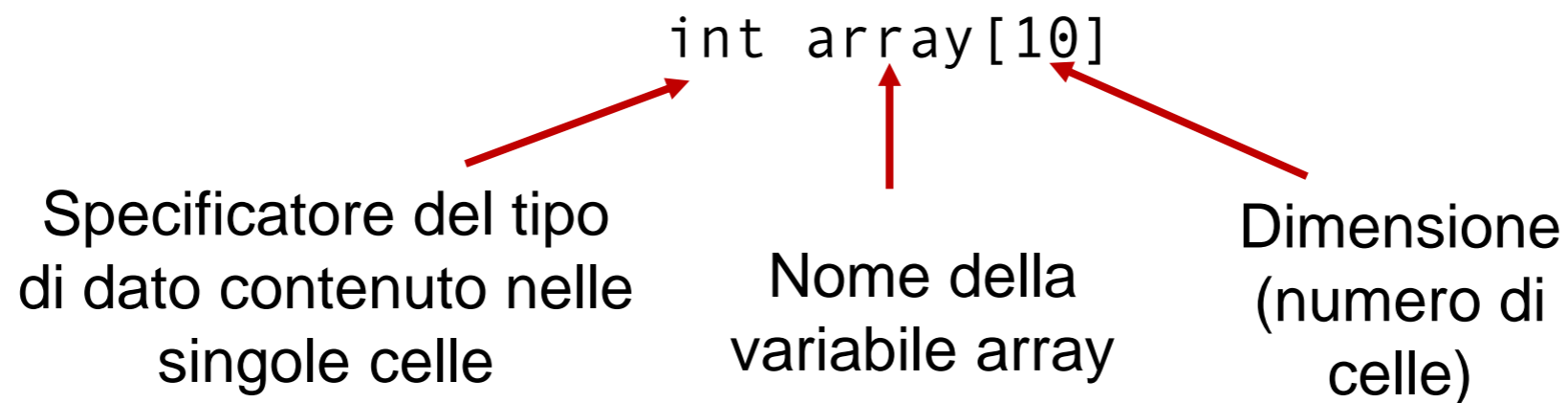
- Contatti:
  - [gianenrico.conti@mail.polimi.it](mailto:gianenrico.conti@mail.polimi.it)
  - [michele.zanella@polimi.it](mailto:michele.zanella@polimi.it)
    - HEAP Lab – Campus Leonardo, via Golgi 39, Edificio 21, Piano 1, Ufficio 4, +39 02 2399 **9613**  
(mandatemi una mail per ricordarci su giorno e ora)
- Tutor: Niccolò Izzo: [niccolò.izzo@mail.polimi.it](mailto:niccolò.izzo@mail.polimi.it)
- Sito web del corso:
  - <https://beep.metid.polimi.it>
- Repository con i sorgenti delle soluzioni:
  - <https://github.com/Fixfire/Info--A>
- **Nota per le mail:**  
Oggetto: *[INFO A] il vostro oggetto*

- **Calendario laboratori**

Data	Orario	Squadra	Aula	Resp.	Programma
23/10/2017	9:00 - 12:00	A	CS 0.2	Zanella	Lab 1
23/10/2017	9:00 - 12:00	B	CS 1.4	Conti	
26/10/2017	14:15-17:15	C	CS 0.11	Conti	
30/10/2017	9:00 - 12:00	A	CS 0.2	Zanella	Lab 2
30/10/2017	9:00 - 12:00	B	CS 1.4	Conti	
02/11/2017	14:15-17:15	C	CS 0.11	Conti	
06/11/2017	9:00 - 12:00	A	CS 0.2	Zanella	Lab 3
06/11/2017	9:00 - 12:00	B	CS 1.4	Conti	
09/11/2017	14:15-17:15	C	CS 0.11	Conti	
20/11/2017	9:00 - 12:00	A	CS 0.2	Zanella	Lab 4
20/11/2017	9:00 - 12:00	B	CS 1.4	Conti	
23/11/2017	14:15-17:15	C	CS 0.11	Conti	
27/11/2017	9:00 - 12:00	A	CS 0.2	Zanella	Lab 5
27/11/2017	9:00 - 12:00	B	CS 1.4	Conti	
30/11/2017	14:15-17:15	C	CS 0.11	Conti	
04/12/2017	9:00 - 12:00	A	CS 0.2	Zanella	Lab 6
04/12/2017	9:00 - 12:00	B	CS 1.4	Conti	
14/12/2017	14:15-17:15	C	CS 0.11	Conti	

# Array

- Sono dei contenitori formati da una serie di celle in numero fissato contenenti valori di un certo tipo (int, long, char, ...).



- Abbiamo già incontrato gli array con le stringhe: array di caratteri  
`char stringa[20];`
- Accesso agli elementi degli array, se la sua dimensione è 10:
  - Primo elemento: `array[0];`
  - *i*-esimo elemento: `array[i];`
  - Ultimo elemento: `array[9];` ← **ATTENZIONE!**

- Sono degli array bidimensionali (righe,colonne)

```
int matrix[10][10]
```

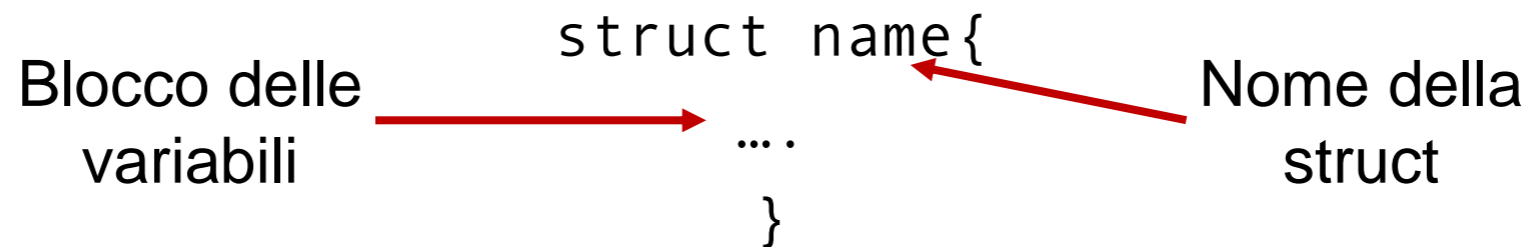
- Inizializzazione degli array e delle matrici:

```
char a[2] = {'a', 'b'};

int m[2][2] = {
    {0, 1}, // Riga 0
    {2, 3}  // Riga 1
};
```

# Struct

- Sono dei tipi di dato composti da una serie di variabili di vario tipo poste sotto lo stesso blocco di memoria, in questo modo esse sono accessibili da un singolo *puntatore* o tramite una variabile istanziata come il tipo della *struct*.



- Posso definire un nuovo tipo di dato che si riferisce alla struct in modo da poter istanziare delle variabili tramite l'operatore *typedef*:

```
typedef struct name { ...  
} type_name ← Nome del nuovo tipo
```

- Per dichiarare una variabile del tipo da noi definito:

```
type_name var;
```

- Esempio, dichiariamo una struct per definire un punto cartesiano:

```
typedef struct point{  
    int x; // Coordinata x  
    int y; // Coordinata y  
} point;  
  
point p;
```

- Per accedere e assegnare ai campi della variabile *p* utilizziamo la notazione puntata ossia l'operatore ".":

```
p.x = 10; // Coordinata x del punto p  
p.y = 20; // Coordinata y del punto p  
  
printf(" Le coordinate di p sono (%d,%d)\n", p.x, p.y);
```

# Esercizio 1: Palindromi

Si scriva un programma che verifichi se una parola è palindroma. Palindroma è una parola che letta da sinistra a destra e viceversa rimane invariata.

## Esempio:

anna

Osso

ereggere

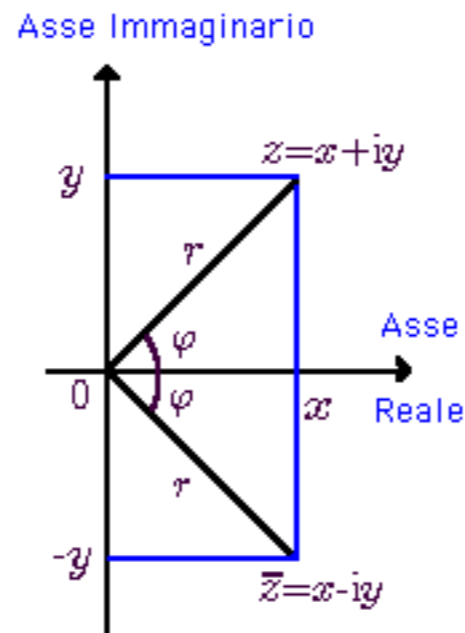
Sono parole palindrome

**Variante:** si possono verificare anche se le frasi sono palindrome, basta inserirle senza spazi: *"itopinonavevanonipoti"*. Oppure togliendo gli spazi alla frase inserita.



## Esercizio 2: Numeri complessi

1) si scriva una opportuna *struct* del C atta a rappresentare i numeri complessi.



Un numero complesso deve avere parte Reale e parte Immaginaria.

2) si dichiara una variabile  $N$  del tipo *struct* appena creato.

## Esercizio 2: Numeri complessi (cont'd)

Si scrivano un frammento di codice che:

- 3) legga le componenti  $R$  e  $I$  del numero  $N$
- 4) calcoli e stampi il Modulo di  $N$  acquisito
- 5) legga due numeri complessi  $N1$  e  $N2$  e ne calcoli la somma (la somma è a sua volta un numero complesso)
- 6) legga due numeri complessi  $N1$  e  $N2$  e dica se sono uguali in argomento (fase)
- 7) legga un array (detto "valori") di "MAX" numeri complessi e stampi il più grande in modulo