



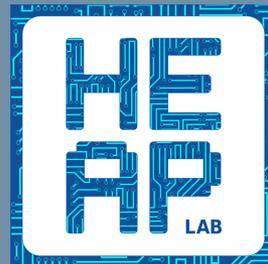
POLITECNICO
MILANO 1863

INFORMATICA (PER AEROSPAZIALI)

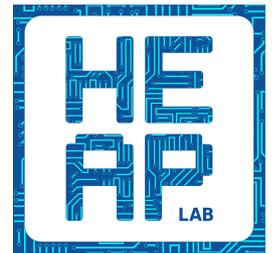
A.A. 2018-19

Laboratorio n°2

Dott. Michele Zanella

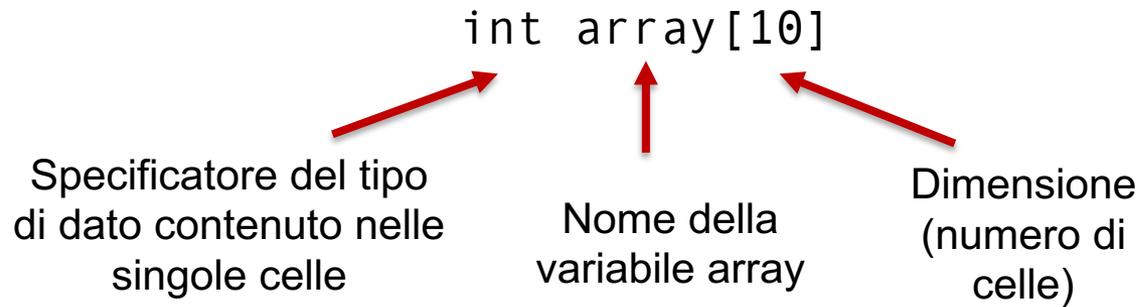


- Contatti:
 - michele.zanella@polimi.it
 - [HEAP Lab](#) – Campus Leonardo, via Golgi 39, Edificio 21, Piano 1, Ufficio 4, +39 02 2399 **9613**
(mandatemi una mail per accordarci su giorno e ora)
- Sito web del laboratorio:
 - [Pagina del corso BeeP \(Cartella Laboratori\)](#)
 - <http://zanella.faculty.polimi.it/teaching/informatica-aero>
- **Nota per le mail:**
Oggetto: *[INFO-AERO] il vostro oggetto*



Array

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



- 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
};
```

- 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*.

```
struct name{  
    ...  
}
```

Blocco delle variabili → ...

Nome 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_t;

point_t 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 2.1: Ricerca sequenziale

Scrivere un programma che, dato un vettore di n numeri, ricerchi un valore al suo interno, indicando l'indice del vettore che contiene il numero da ricercare

Esercizio 2.2: Crivello di Eratostene

Scrivere un programma che implementi il [crivello di Eratostene](#), antico algoritmo per il calcolo dei primi n numeri primi.

Esercizio 2.3: Ordinamento – Bubble sort

Scrivere un programma che, dato un vettore non ordinato di n numeri, implementi l'algoritmo Bubble sort per ordinare i valori in ordine crescente seguendo.

Bubble sort: Ogni coppia di elementi adiacenti viene comparata e invertita di posizione se sono nell'ordine sbagliato. L'algoritmo continua nuovamente a ri-eseguire questi passaggi per tutta la lista finché non vengono più eseguiti scambi, situazione che indica che la lista è ordinata.

Esercizio 2.4: Somma tra matrici

Scrivere un programma che date due matrici ne calcoli la matrice somma.

Le matrici devono avere stesse dimensioni tra di loro.

Esempio:

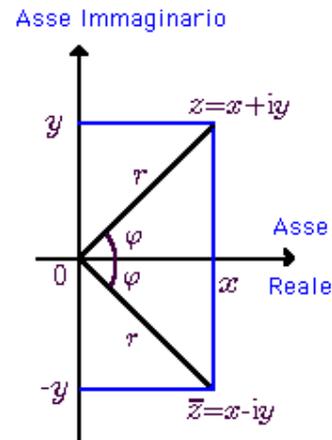
$$\begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 3 & 10 \\ 6 & 5 \end{bmatrix} = \begin{bmatrix} 7 & 12 \\ 6 & 6 \end{bmatrix}$$

Esercizio 2.5: Matrice - Riempimento concentrico

Scrivere un programma che riempia una matrice in modo concentrico, ricevendo gli interi da I/O, e che stampi la matrice formattata.

Esercizio 2.6: 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 dichiari una variabile N del tipo struct appena creato.

Esercizio 2.6: 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) dati due numeri complessi N1 e N2, ne calcoli la somma (la somma è a sua volta un numero complesso)
- 6) dati due numeri complessi N1 e N2, dica se sono uguali in modulo

Esercizio 2.7: Rubrica del telefono

Si scriva un programma che implementi una semplice rubrica del telefono.

Requisiti:

- Possibilità di inserire/cancellare contatti
- Possibilità di stampare elenco contatti con informazioni
- Possibilità di ricercare un contatto dato il cognome

Hints:

- Riutilizzare eventuali funzioni contenuti in librerie (es. `string.h`)
- Settare un numero massimo `CONT_MAX` di contatti

Esercizio 2.7: Rubrica del telefono (cont'd)

Contatto:

Definiamo un tipo di dato personalizzato *contatto*. Il tipo deve includere le informazioni relative a:

- Cognome
- Nome
- Età
- Numero di telefono

Esercizio a casa 2.1: Ordinamento – Cocktail (Shaker) sort

Scrivere un programma che, dato un vettore non ordinato di n numeri, implementi l'algoritmo Cocktail sort per ordinare i valori in ordine crescente seguendo.

Cocktail (shaker) sort: Versione bidirezionale del Bubble sort. Si differenzia da quest'ultimo poiché il ciclo più interno anziché scorrere dall'inizio alla fine del vettore, inverte la sua direzione ad ogni ciclo.

Provare ad implementare anche altri algoritmi di ordinamento: Selection sort, Insertion sort... https://en.wikipedia.org/wiki/Sorting_algorithm