



Scuola di Ingegneria Industriale  
Laurea in Ingegneria Energetica  
Laurea in Ingegneria Meccanica



**POLITECNICO  
DI MILANO**

Dipartimento di  
Elettronica, Informazione  
e Bioingegneria



# Informatica B

## Sezione D

*Franchi Alessio Mauro, PhD    [alessiomauro.franchi@polimi.it](mailto:alessiomauro.franchi@polimi.it)*



# Esercizio 24

Scrivere un programma che definisca un sequenza di numeri interi (=array) e la stampi a video, prima in ordine di inserimento, poi al contrario.



# Esercizio 24 - Soluzione

dichiarazione

```
int v[12];
```

nome array

v[0]

v[1]

v[2]

v[3]

v[4]

v[5]

v[6]

v[7]

v[8]

v[9]

v[10]

v[11]

-45

6

0

72

1543

-89

0

62

-3

1

6453

78

indice elemento

In c una sequenza di numeri, caratteri è un **array**

- Un array è un **gruppo di celle di memoria consecutive**;
- Hanno tutte **stesso nome e tipo**;
- Per riferirsi a un elemento, si specificano:
  - Il nome dell'array
  - La posizione dell'elemento (indice)
  - Formato: nomearray[posizione]
- Il primo elemento ha **indice 0**
- L'n° elemento dell'array v è **v[n-1]**



# Esercizio 24 - Soluzione

Esempio di array: **char parola[12];**

- È un array di 12 elementi di tipo char, vale a dire un vettore di 12 caratteri
- La lunghezza dell'array è comunque decisa durante la compilazione del programma
- Non si usano variabili per specificare, nella dichiarazione, la dimensione degli array

```
int i = 10;  
char parola[i] = {'a','b'};
```

**Non compila!**



# Esercizio 24 - Soluzione

Un array si può dichiarare in diversi modi: questi sono tutti **equivalenti!**

```
int main()
{
    int i; //variabile contatore per i cicli for

    /*questi sono tre modi identici di dichiarare un array*/

    int seq[5] = {1,4,2,6,10};

    int seq1[] = {7,3,1,23};

    int seq2[4];
    seq2[0] = 1;
    seq2[1] = 10;
    seq2[2] = 90;
    seq2[3] = 4;
```



# Esercizio 24 - Soluzione

Sugli array si lavora sempre **elemento per elemento**; ogni elemento è una normale variabile

Esempio: **stampa di un array**

```
int i, seq[5] = {1,2,3,4,5};  
for (i = 0; i < 5; i++ )  
{  
    printf("%d ", seq[i]);  
}
```

Esempio: **come inizializzo un array?**

```
int i;  
for (i = 0; i < 20; i++)  
    a[i] = 0;
```

Pongo tutti gli elementi del vettore a pari a 0;



# Esercizio 24 - Soluzione

Il problema chiede di stampare gli array anche in ordine inverso.  
Utilizzo un ciclo for che parte dalla fine dell'array e arrivi all'inizio!

L'indice *i* parte da 3 e non da 4 (l'ultimo elemento ha indice 3 perché parto a contare da 0)

Scorro l'array al contrario, quindi ciclo fino a quando l'indice è pari a 0 (0 compreso!)

```
for (i = 3; i >= 0; i--)
```

```
{
```

```
    printf("%d ", seq1[i]);
```

```
}
```

Decremento l'indice



# Esercizio 25

Scrivere un programma che definisca una sequenza di numeri di lunghezza massima MAXLEN definita a priori e ne calcoli il massimo, il minimo e il valore medio.





# Esercizio 25

Anche in questo caso ci serve un array. Definisco prima anche una define per la lunghezza massima dell'array.

```
#define MAXLEN 5
```

```
int main()  
{
```

```
    int arr[MAXLEN] = {1,4,2,6,10};
```

Questa volta non è richiesto l'input da tastiera; inizializzo il mio array nel codice

```
    int max,min,i;  
    float mean,sum;  
    sum = arr[0];  
    max = arr[0];  
    min = arr[0];
```

Ricordatevi di inizializzare la variabile somma!

Per calcolare il massimo e il minimo di un array inizializzo due variabili pari al primo elemento dell'array (che potrebbe essere sia il massimo che il minimo!)



# Esercizio 25

Ora devo calcolare minimo, massimo e valore medio. Devo scorrere tutto l'array (del quale conosco la lunghezza): uso un ciclo for!

Il primo elemento è già stato considerato, parto dal secondo, quindi  $i=1$

```
for(i=1; i<MAXLEN;i++)  
{
```

Per calcolare la media devo prima di tutto sommare tutti gli elementi. “+=” significa  $sum = sum + arr[i]$

```
    sum += arr[i];
```

```
    if (arr[i]>max)
```

```
        max = arr[i];
```

```
    if (arr[i]<min)
```

```
        min = arr[i];
```

Mantengo aggiornata la variabile max; se l'elemento attuale è maggiore del max attuale, aggiorno il max.

Stesso discorso per il minimo

```
}
```

```
mean = sum/MAXLEN;
```

Calcolo la media aritmetica

```
printf("La media del vettore è %f\n",mean);
```

```
printf("Il massimo del vettore è %d\n",max);
```

```
printf("Il minimo del vettore è %d",min);
```



# Esercizio 26, 27, 28

Scrivere un programma che legga una sequenza di numeri interi (-1 per terminare) e la stampa in ordine invertito. La lunghezza massima della sequenza deve essere minore od uguale a 50. (attenzione: cosa succede se l'utente inserisce 51 numeri?).

Data una sequenza di caratteri, scrivere un programma che chieda all'utente di inserire un carattere da tastiera e verifichi se è presente nella sequenza. In caso positivo stampare a video l'indice della prima occorrenza, -1 altrimenti.

Dato un array di caratteri non ordinato, ordinarlo in ordine crescente e stamparlo a video.



# Esercizio 29

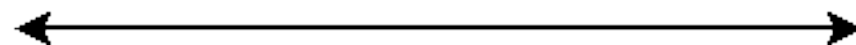
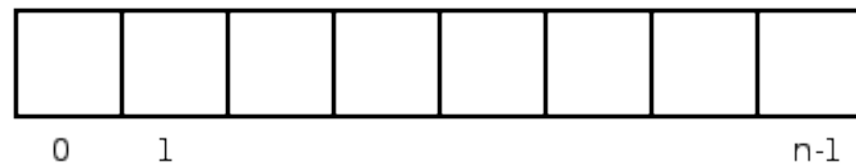
Scrivere un programma che definisca una matrice quadrata di interi di dimensione  $N$  e la stampi a video.



# Esercizio 29 - Soluzione

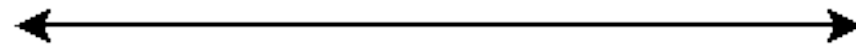
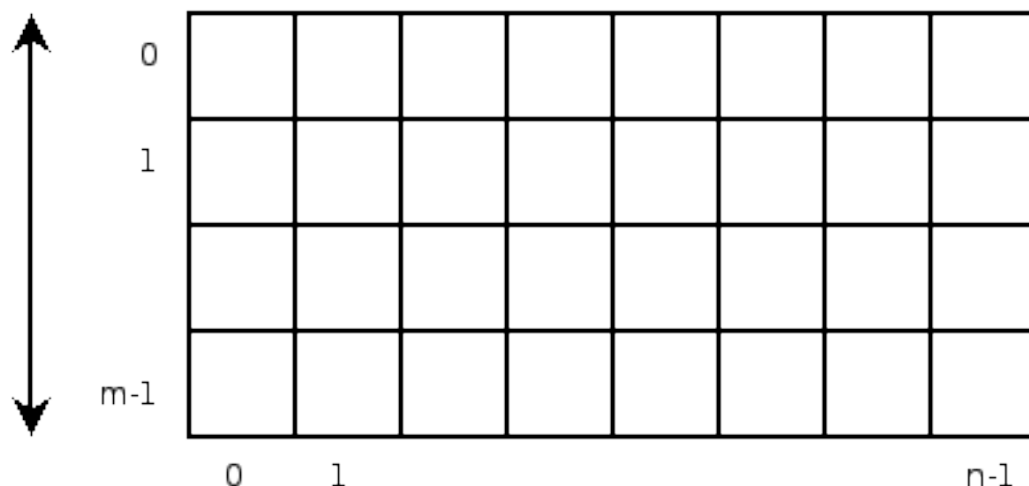
Ripasso: un **array** è una collezione di elementi; sono tutti dello stesso tipo e hanno tutti lo stesso nome.

Ha una sola dimensione (è tutto su una sola riga) quindi mi basta un solo indice per selezionare i vari elementi.



Una **matrice** è un array con due dimensioni, le righe e le colonne. Quindi mi servono due indici per selezionare un elemento, la riga e la colonna.

E' come la battaglia navale!





# Esercizio 29 - Soluzione

Definisco la mia matrice:

```
#include <stdio.h>
```

```
#define ROW 3
```

```
#define COL 3
```

Ho bisogno di due define, una per ciascuna dimensione

```
int main()
```

```
{
```

```
    int matrix[ROW][COL] = {{0,1,2},{3,4,5},{6,7,8}};
```

Ognuna di queste è una riga e va racchiusa tra graffe; le righe assieme vanno racchiuse tra graffe più esterne!

La variabile matrix ha due dimensioni: uso due “[ ]” “[ ]” per specificarne la lunghezza massima



# Esercizio 29 - Soluzione

Devo stampare a video la matrice; la devo scansionare tutta. Avendo due dimensioni mi servono **due cicli for, uno dentro l'altro**; in questo modo quello più esterno scansiona le righe, quello più interno le colonne

```
for(i=0;i<ROW;i++)  
{  
    for(j=0;j<COL;j++)  
        printf("%d\t",matrix[i][j]);  
    printf("\n");  
}  
  
return 0;
```

Il ciclo esterno fissa la riga attuale: l'indice i

Il ciclo esterno, avendo fissato la riga, la scansiona tutta, colonna per colonna: l'indice j

"\t" inserisci quattro spazi bianchi

Notate la sua posizione subito dopo il ciclo interno. Quando ho stampato a video tutta una riga, vado a capo e stampo la riga successiva!



# Esercizio 30

Scrivere un programma che definisca una matrice quadrata di dimensione  $N$  e stampi a video se la matrice è simmetrica o no.

(Prima pensate all'algoritmo, poi pensate se può essere ottimizzato in qualche modo)





# Esercizio 31

Scrivere un programma che definisca una stringa di lunghezza LEN e la stampi a video prima normale e poi al contrario.

(Ricordatevi che una stringa è un array di caratteri!)



# Esercizio 31

Una stringa è un array di char, terminata dal carattere speciale “\0”

Per definire una stringa ci sono due modi totalmente equivalenti:

```
char c[LEN] = {'H','e','l','l','o','\0'};  
char c1[LEN] = "Hello world";
```

← Come un vero e proprio array, e devo specificare io stesso il carattere di terminazione “\0”

↖ Con i doppi apici “stringa”, non serve in questo specificare il “\0”

Essendo a tutti gli effetti un array, si lavora come sugli array, quindi **elemento per elemento**.



# Esercizio 31


Ad esempio, stampare una stringa, due modi differenti ma con medesimo risultato:

```
for(i=0; i<LEN;i++)  
{  
    printf("%c",c[i]);  
}
```

```
i = 0;  
while(c[i]!='\0')  
{  
    printf("%c",c[i]);  
    i++;  
}
```

```
printf("La stringa è: %s\n",varStringa);
```

“%s” denota una stringa; questo stampa tutta la stringa, senza doverla scorrere elemento per elemento “a mano”





# Esercizio 32, 33, 34

Scrivere un programma che chieda all'utente di inserire una parola e la ristampi a video con tutti i caratteri maiuscoli.

Scrivere un programma che verifichi se due stringhe inserite dall'utente da tastiera sono uguali.

Creare un registro di cognomi.

L'utente da tastiera può inserire un nuovo cognome (la lista non deve contenere duplicati), leggere la lista di tutti i cognomi inseriti e verificare se un cognome esiste, restituendone a video l'indice.