



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 35

Dati due array A e B di lunghezza massima N, chiedere all'utente quanti numeri vuole inserire, leggere da tastiera A e B e costruire quindi un array C (di dimensione massima  $2N$ ) che contenga nelle posizioni di indice pari gli elementi di A e nelle posizioni di indice dispari gli elementi di B.



# Esercizio 35 - Soluzione

L'esercizio è esplicito; chiede di definire due array iniziali di lunghezza  $N$ , più un terzo di lunghezza  $2*N$

```
int a[MAX],b[MAX],c[2*MAX], d[2*MAX];  
int n;  
  
do{  
    printf("Inserisci il numero di elementi: ");  
    scanf("%d",&n);  
}while(n < 0 || n>MAX);
```

## A cosa serve il ciclo while?

Mi permette di controllare l'input da tastiera;  $n$  deve essere maggiore di 0 e minore di  $MAX$  (definito da un `DEFINE`)



# Esercizio 35 - Soluzione

Devo riempire gli array leggendo i valori da tastiera. Perchè usamo il for?

```
for(i=0;i<n;i++)  
{  
    printf("A[%d]:",i);  
    scanf("%d",&a[i]);  
}
```

Ora devo creare il terzo array! Nelle posizioni di indice pari metto gli elementi di a, in quelle dispari gli elementi di b

```
for(i=0;i<n;i++)  
{  
    c[j++]= a[i];  
    c[j++]= b[i];  
}
```



```
for(i=0;i<n;i++)  
{  
    d[++j]= a[i];  
    d[++j]= b[i];  
}
```

Fanno la stessa cosa? Danno lo stesso output? Provate!



# Esercizio 36

Scrivere un programma in linguaggio C che acquisisca da tastiera due valori interi "row" e "col" (i due valori devono essere pari, maggiori di 0 e minori di MAX=100). Acquisisca quindi da tastiera una matrice m di interi di dimensioni massime MAX x MAX, organizzandoli su un numero di righe uguale a row e un numero di colonne uguale a col. Stamparla a video ed estrarre quindi tutte le sottomatrici 2x2 e calcolarne il determinante.



# Esercizio 36 - Soluzione

Una matrice è un array a due dimensioni, le righe e le colonne;

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Si definisce come

tipo nome[DIM1][DIM2];

Esempio

**int matrice[10][10];**

Come per gli array, anche sulla matrici si lavora sempre elemento per elemento; ogni elemento in questo caso ha due indici:

Esempio:

**matrice[2][4] = 10;**

Valgono sempre tutte le raccomandazioni fatte per gli array:

Controllare gli indici che utilizzate;

Non lavorate sulle matrici intere;

Una matrice comporta dei cicli, attenzione ai cicli infiniti!



# Esercizio 36 - Soluzione

Acquisiamo da tastiera il numero di righe e di colonne; questi devono essere pari e compresi tra 0 e MAX = 100, usiamo quindi un while per il controllo dell'input!

```
int m[MAX][MAX], row, col;

do{
    printf("Inserisci il numero di righe: ");
    scanf("%d",&row);
}while(row < 0 || row>MAX || (row%2!=0));

do{
    printf("Inserisci il numero di colonne: ");
    scanf("%d",&col);
}while(col < 0 || col>MAX || col%2!=0);
```

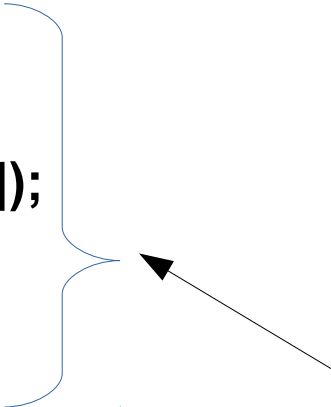


# Esercizio 36 - Soluzione

La matrice si riempie elemento per elemento, leggendo ogni intero da tastiera; che ciclo (e quanti) usiamo?

```
for(i = 0; i < row; i++)  
    for(j = 0; j < col; j++){  
        printf("Inserire l'elemento [%d, %d]: ", i, j);  
        scanf("%d", &m[i][j]);  
    }
```

```
for(i = 0; i < row; i++){  
    for(j = 0; j < col; j++)  
        printf("%d\t", m[i][j]);  
    printf("\n");  
}
```



Stampa a video della matrice.  
Il “\t” inserisce quattro spazi  
bianchi





# Esercizio 36 - Soluzione

Dobbiamo adesso estrarre le sottomatrici (stamparle a video) 2x2.

```
for(i = 0; i < row; i = i + 2)
    for(j = 0; j < col; j = j + 2)
    {
        // ...
    }
```

1	<table><tr><td>3</td><td>20</td></tr><tr><td>6</td><td>7</td></tr></table>	3	20	6	7	2	<table><tr><td>4</td><td>5</td></tr><tr><td>50</td><td>60</td></tr></table>	4	5	50	60	3	<table><tr><td>2</td><td>7</td></tr><tr><td>9</td><td>15</td></tr></table>	2	7	9	15
3	20																
6	7																
4	5																
50	60																
2	7																
9	15																
4	<table><tr><td>2</td><td>4</td></tr><tr><td>80</td><td>3</td></tr></table>	2	4	80	3	5	<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1	6	<table><tr><td>20</td><td>10</td></tr><tr><td>8</td><td>12</td></tr></table>	20	10	8	12
2	4																
80	3																
1	1																
1	1																
20	10																
8	12																

Calcolo determinante di una matrice quadrata 2x2

$$\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = (a_1 \cdot b_2) - (a_2 \cdot b_1)$$

```
det = (m[i][j] * m[i+1][j+1] - m[i+1][j] * m[i][j+1]);
printf("%d\t%d\n%d\t%d", m[i][j], m[i][j+1], m[i+1][j],
        m[i+1][j+1]);
printf("\n Il determinante della sottomatrice è
        %d\n", det);
```



# Esercizio 37

Scrivere un programma che legga da tastiera un array di dimensione massima MAXLEN contenente numeri interi negativi (zero per terminare l'inserimento); il programma deve copiare l'array in un secondo array in ordine inverso e calcolare la media mean dei valori dispari inseriti in posizioni multiple di due dell'array.



# Esercizio 38

Scrivere un programma che richieda all'utente di inserire i dati di studenti, al massimo  $N=10$ . Ogni studente è rappresentato da matricola, nome e cognome. Gli studenti devono essere inseriti in un vettore di struct, che deve essere mantenuto ordinato dopo ogni inserimento.



# Esercizio 38 - Soluzione

Il problema chiede di registrare dieci studenti; utilizzeremo un array di dimensione massima 10.

Di che tipo lo faccio questo array?

## typedef struct

Dato che ogni studente è caratterizzato da matricola, nome e cognome (campi), dobbiamo definire un nuovo tipo di dato con una struct.

In generale...

```
typedef struct
{
    tipo membro_1;
    tipo membro_2;
    ...
    tipo membro_n;
} nomeTipo;
```

In questo caso...

```
typedef struct
{
    int matricola;
    char nome[20];
    char cognome[20];
} studente;
```



# Esercizio 38 - Soluzione

**Ogni struct va definita prima e all'esterno della funzione main**

Quando ho definito una struct posso poi dichiarare una nuova variabile che ha come tipo la struct stessa. Esempio:

**studente registro[MAXLEN]; //MAXLEN è una define!**

La variabile registro (in questo caso un array) è una una normalissima variabile, che ha però dei campi. Per accedere ai campi si usa la notazione “.”; esempio:

**registro[0].matricola = 10345366;**



Accedo al campo matricola della variabile registro (che è tipo studente, struct!)



# Esercizio 38 - Soluzione

Richiedo l'inserimento degli studenti; continuo sino a quando l'utente risponde 'y' all'ultima domanda!

**do{**

**printf("Inserisci la matricola dello studente: ");**

**scanf("%d",&registro[len].matricola);**

**printf("Inserisci il nome dello studente: ");**

**scanf("%s",registro[len].nome);**

**getchar();** ◀

**printf("Inserisci il cognome dello studente: ");**

**scanf("%s",registro[len].cognome);**

**getchar();** ◀

**len++;**

**...**

**printf("Vuoi inserire un nuovo studente? ");**

**scanf("%c",&risp);**

**getchar();** ◀

**}while(risp!='n');**

La getchar() serve  
per eliminare il  
carattere speciale  
"invio".

Provate senza!

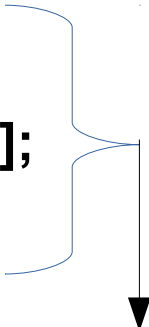


# Esercizio 38 - Soluzione

Devo però ordinare l'array di studenti rispetto al valore della matricola.  
Utilizzo un algoritmo di ordinamento di array: bubble sort!

<https://www.youtube.com/watch?v=8Kp-8OGwphY>

```
for(i=0;i<len-1;i++)  
    for(j=0;j<len-i-1;j++)  
        if(registro[j].matricola>registro[j+1].matricola)  
        {  
            temp=registro[j];  
            registro[j] = registro[j+1];  
            registro[j+1] = temp;  
        }
```



Classica procedura per lo scambio del valore di due variabili! Se avete:

```
int a=10, b=5;
```

E ne volete scambiare i due valori non basta scrivere:

```
a=b;
```

```
b=a;
```

Ma ci vuole una terza variabile dove salvare temporaneamente il valore di una delle due variabili!



# Esercizio 39

Si definisca un tipo strutturato `SquadraBasket`, che rappresenti i dati relativi a una squadra di basket. Una squadra di basket è rappresentata dalle seguenti informazioni:

- Nome della squadra: una stringa lunga al massimo 30 caratteri.
- Città della squadra: una stringa lunga al massimo 30 caratteri.
- Numero di giocatori che fanno parte della squadra: un intero, minore o uguale a 20.
- L'insieme di Giocatori che compongono la squadra.

Un giocatore è un dato strutturato costituito da:

- Nome del giocatore: una stringa lunga al massimo 30 caratteri.
- Numero di partite giocate dal giocatore nella stagione corrente: un intero.
- Numero di punti realizzati dal giocatore nella stagione corrente: un intero.

Scrivere un programma per la gestione di un campionato; si dichiara una variabile `campionato`, come vettore di 16 squadre; dichiarare poi un vettore di squadre contenente le squadre del campionato che hanno almeno 3 giocatori che hanno realizzato in media più di 10 punti a partita.





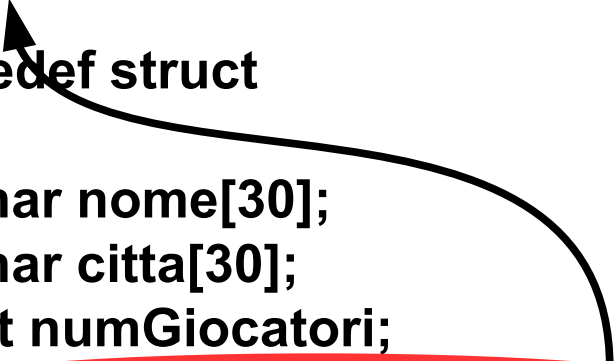
# Esercizio 39

E' un esercizio molto simile a quello precedente; dobbiamo gestire un "campionato", ci serve un array che contenga i dati delle squadre. Dato che una squadra è composta da più "campi" (nome, città...) ci serve una **struct**.

```
typedef struct
{
    char nome[30];
    int numeroPartire;
    int numeroPunt;
```

```
} giocatore;
```

```
typedef struct
{
    char nome[30];
    char citta[30];
    int numGiocatori;
    giocatore listaGiocatori[20];
} squadraBasket;
```



**Attenzione però:** ogni squadra ha una lista di giocatori; un giocatore è a sua volta composto da più campi, quindi è una struct!

La struct "giocatore" va definita prima di quella "squadraBasket" perchè utilizzata da quest'ultima!



# Esercizio 39

```
do{
```

```
    printf("\n\nInserimento nuova squadra: [%d]\n",i+1);
```

```
    printf("Inserisci il nome: ");
```

```
    scanf("%s", campionato[i].nome);
```

```
    printf("Inserisci città: ");
```

```
    scanf("%s", campionato[i].città);
```

```
    printf("Numero giocatori: ");
```

```
    scanf("%d", &campionato[i].numGiocatori);
```

```
    getchar();
```

```
    ...
```

```
    i++;
```

```
    if(i < 16)
```

```
    {
```

```
        printf("Vuoi inserire un'altra squadra? [y/n]");
```

```
        scanf("%c",&risp);
```

```
    }
```

```
    else
```

```
        printf("Campionato al completo!\n");
```

```
while(risp!='n' && i < 16);
```

Utilizzo della notazione "." per accedere ai campi della struttura

Chiedo all'utente se vuole inserire un'altra squadra ['y' o 'n']

Ciclo finchè l'utente scrive 'y' oppure ho finito gli spazi.

Quindi, **quale sarà la condizione di permanenza nel ciclo?**



# Esercizio 39

```
j=0;
printf("Vuoi inserire i giocatori? [y/n]");
scanf("%c",&risp);
while(risp!='n' && j < campionato[i].numGiocatori){
    printf("Nome %d giocatore: ",j+1);
    scanf("%s",campionato[i].listaGiocatori[j].nome);
    printf("Numero partite %d giocatore: ",j+1);
    scanf("%d",&campionato[i].listaGiocatori[j].numeroPartire);
    printf("Punti %d giocatore: ",j+1);
    scanf("%d",&campionato[i].listaGiocatori[j].numeroPunt);
    getchar();
    j++;
    if(j < campionato[i].numGiocatori){
        printf("Vuoi inserire un altro giocatore? [y/n]");
        scanf("%c",&risp);
        getchar();
    }
    else
        printf("Lista giocatori completa!\n");
}
```

Per accedere ai campi della struct "giocatore" devo "passare" dalla struct "squadraBasket". Mi servono quindi **due "punti"**



# Esercizio 39

Il problema chiede adesso di estrarre le squadre con almeno tre giocatori che abbiamo una media punti superiore a 10. Devo scansionare tutto l'array "campionato" e calcolare la media punti dei giocatori (scansioni l'array dei giocatori).

```
for(i=0;i<numeroSquadre;i++){  
    counter = 0;  
    for(j=0;j<campionato[i].numGiocatori;j++){  
        float mean =  
campionato[i].listaGiocatori[j].numeroPunt/campionato[i].  
listaGiocatori[j].numeroPartire;  
        if (mean >= 10)  
            counter++;  
    }  
    if (counter >= 3){  
        best[k] = campionato[i];  
        k++;  
    }  
}
```

Per ogni squadra scansiono tutto l'array dei giocatori e ne calcolo la media punti

Se la media è maggiore di 10 incremento un contatore

Se il contatore è maggiore di 3 copio la squadra nell'array best